Computer Algebra
Independent Integration Tests

Summer 2023 edition

1-Algebraic-functions/1.1-Binomial-products/1.1.1-Linear/16-1.1.1.5-P-x-a+b-x-^m-c+d-x-^n

Nasser M. Abbasi

September 5, 2023 Compiled on September 5, 2023 at 7:56pm

Contents

CHAPTER 1

INTRODUCTION

This report gives the result of running the computer algebra independent integration test.The download section in on the main webpage contains links to download the problems in plain text format used for all CAS systems. The number of integrals in this report is [34]. This is test number [16].

1.1 Listing of CAS systems tested

The following are the CAS systems tested:

1. Mathematica 13.3.1 (August 16, 2023) on windows 10.
2. Rubi 4.16.1 (Dec 19, 2018) on Mathematica 13.3 on windows 10
3. Maple 2023.1 (July, 12, 2023) on windows 10.
4. Maxima 5.47 (June 1, 2023) using Lisp SBCL 2.3.0 on Linux via sagemath 10.1 (Aug 20, 2023).
5. FriCAS 1.3.9 (July 8, 2023) based on sbcl 2.3.0 on Linux via sagemath 10.1 (Aug 20, 2023).
6. Giac/Xcas 1.9.0-57 (June 26, 2023) on Linux via sagemath 10.1 (Aug 20, 2023).
7. Sympy 1.12 (May 10, 2023) Using Python 3.11.3 on Linux.
8. Mupad using Matlab 2021a with Symbolic Math Toolbox Version 8.7 on windows 10.

Maxima and Fricas and Giac are called using Sagemath. This was done using Sagemath `integrate` command by changing the name of the algorithm to use the different CAS systems.

Sympy was run directly in Python not via sagemath.

1.2 Results

Important note: A number of problems in this test suite have no antiderivative in closed form. This means the antiderivative of these integrals can not be expressed in terms of elementary, special functions or `Hypergeometric2F1` functions. `RootSum` and `RootOf` are not allowed. If a CAS returns the above integral unevaluated within the time limit, then the result is counted as passed and assigned an A grade.

However, if CAS times out, then it is assigned an F grade even if the integral is not integrable, as this implies CAS could not determine that the integral is not integrable in the time limit.

If a CAS returns an antiderivative to such an integral, it is assigned an A grade automatically and this special result is listed in the introduction section of each individual test report to make it easy to identify as this can be important result to investigate.

The results given in in the table below reflects the above.

System	% solved	% Failed
Rubi	100.00 (34)	0.00 (0)
Mathematica	100.00 (34)	0.00 (0)
Maple	82.35 (28)	17.65 (6)
Fricas	82.35 (28)	17.65 (6)
Giac	82.35 (28)	17.65 (6)
Sympy	55.88 (19)	44.12 (15)
Maxima	47.06 (16)	52.94 (18)
Mupad	11.76 (4)	88.24 (30)

Table 1.1: Percentage solved for each CAS

The table below gives additional break down of the grading of quality of the antiderivatives generated by each CAS. The grading is given using the letters A,B,C and F with A being the best quality. The grading is accomplished by comparing the antiderivative generated with the optimal antiderivatives included in the test suite. The following table describes the meaning of these grades.

grade	description
A	Integral was solved and antiderivative is optimal in quality and leaf size.
B	Integral was solved and antiderivative is optimal in quality but leaf size is larger than twice the optimal antiderivatives leaf size.
C	Integral was solved and antiderivative is non-optimal in quality. This can be due to one or more of the following reasons 1. antiderivative contains a hypergeometric function and the optimal antiderivative does not. 2. antiderivative contains a special function and the optimal antiderivative does not. 3. antiderivative contains the imaginary unit and the optimal antiderivative does not.
F	Integral was not solved. Either the integral was returned unevaluated within the time limit, or it timed out, or CAS hanged or crashed or an exception was raised.

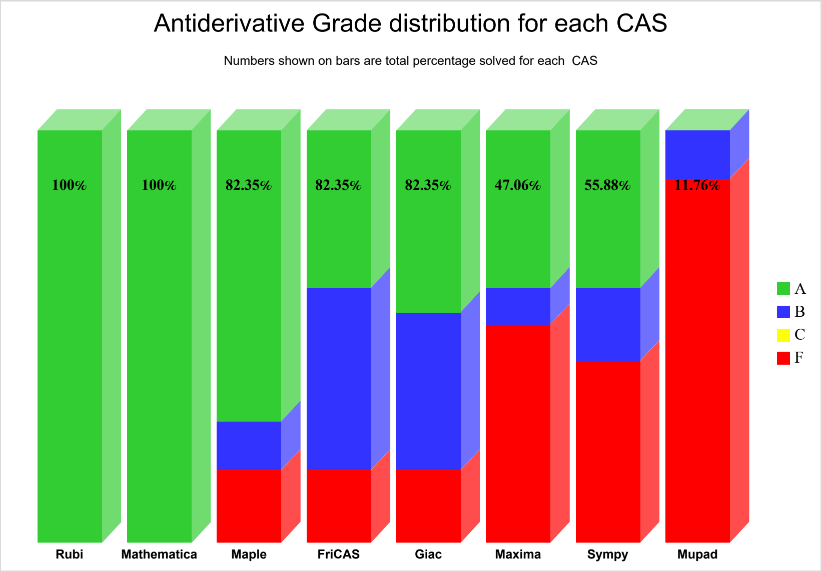
Table 1.2: Description of grading applied to integration result

Grading is implemented for all CAS systems. Based on the above, the following table summarizes the grading for this test suite.

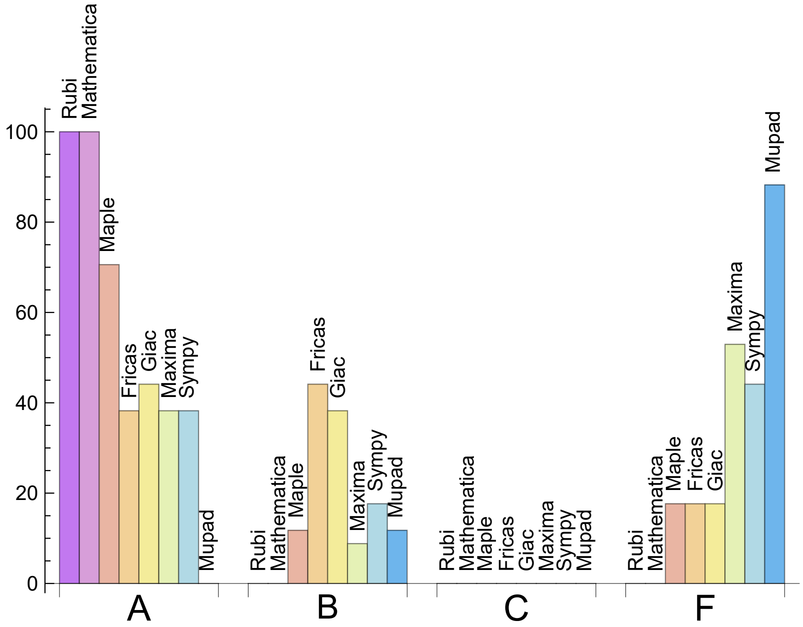
System	% A grade	% B grade	% C grade	% F grade
Rubi	100.000	0.000	0.000	0.000
Mathematica	100.000	0.000	0.000	0.000
Maple	70.588	11.765	0.000	17.647
Giac	44.118	38.235	0.000	17.647
Fricas	38.235	44.118	0.000	17.647
Maxima	38.235	8.824	0.000	52.941
Sympy	38.235	17.647	0.000	44.118
Mupad	0.000	11.765	0.000	88.235

Table 1.3: Antiderivative Grade distribution of each CAS

The following is a Bar chart illustration of the data in the above table.



The figure below compares the grades of the CAS systems.



The following table shows the distribution of the different types of failures for each CAS. There are 3 types failures. The first is when CAS returns the input within the time limit, which means it could not solve it. This is the typical failure and given as **F**.

The second failure is due to time out. CAS could not solve the integral within the 3 minutes time limit which is assigned. This is assigned **F(-1)**.

The third is due to an exception generated, indicated as **F(-2)**. This most likely indicates an interface problem between sagemath and the CAS (applicable only to FriCAS, Maxima and Giac) or it could be an indication of an internal error in the CAS itself. This type of error requires more investigation to determine the cause.

System	Number failed	Percentage normal failure	Percentage timeout failure	Percentage exception failure
Rubi	0	0.00	0.00	0.00
Mathematica	0	0.00	0.00	0.00
Fricas	6	100.00	0.00	0.00
Maple	6	100.00	0.00	0.00
Giac	6	100.00	0.00	0.00
Sympy	15	13.33	60.00	26.67
Maxima	18	33.33	0.00	66.67
Mupad	30	0.00	100.00	0.00

Table 1.4: Failure statistics for each CAS

1.3 Time and leaf size Performance

The table below summarizes the performance of each CAS system in terms of time used and leaf size of results.

Mean size is the average leaf size produced by the CAS (before any normalization). The Normalized mean is relative to the mean size of the optimal anti-derivative given in the input files.

For example, if CAS has **Normalized mean** of 3, then the mean size of its leaf size is 3 times as large as the mean size of the optimal leaf size.

Median size is value of leaf size where half the values are larger than this and half are smaller (before any normalization). i.e. The Middle value.

Similarly the **Normalized median** is relative to the median leaf size of the optimal.

For example, if a CAS has Normalized median of 1.2, then its median is 1.2 as large as the median leaf size of the optimal.

System	Mean time (sec)
Maxima	0.21
Rubi	0.29
Giac	0.31
Fricas	0.32
Mathematica	0.68
Maple	1.78
Mupad	3.99
Sympy	14.25

Table 1.5: Time performance for each CAS

System	Mean size	Normalized mean	Median size	Normalized median
Mathematica	282.68	0.94	242.50	0.95
Rubi	292.21	1.00	272.50	1.00
Mupad	426.50	1.50	432.50	1.61
Maxima	483.62	1.59	390.00	1.22
Maple	503.21	1.56	288.50	0.99
Giac	1084.11	3.27	587.50	1.76
Fricas	1349.57	4.07	777.50	3.75
Sympy	6431.84	18.78	476.00	1.66

Table 1.6: Leaf size performance for each CAS

1.4 Performance based on number of rules Rubi used

This section shows how each CAS performed based on the number of rules Rubi needed to solve the same integral. One diagram is given for each CAS.

On the y axis is the percentage solved which Rubi itself needed the number of rules given the x axis. These plots show that as more rules are needed then most CAS system percentage of solving decreases which indicates the integral is becoming more complicated to solve.

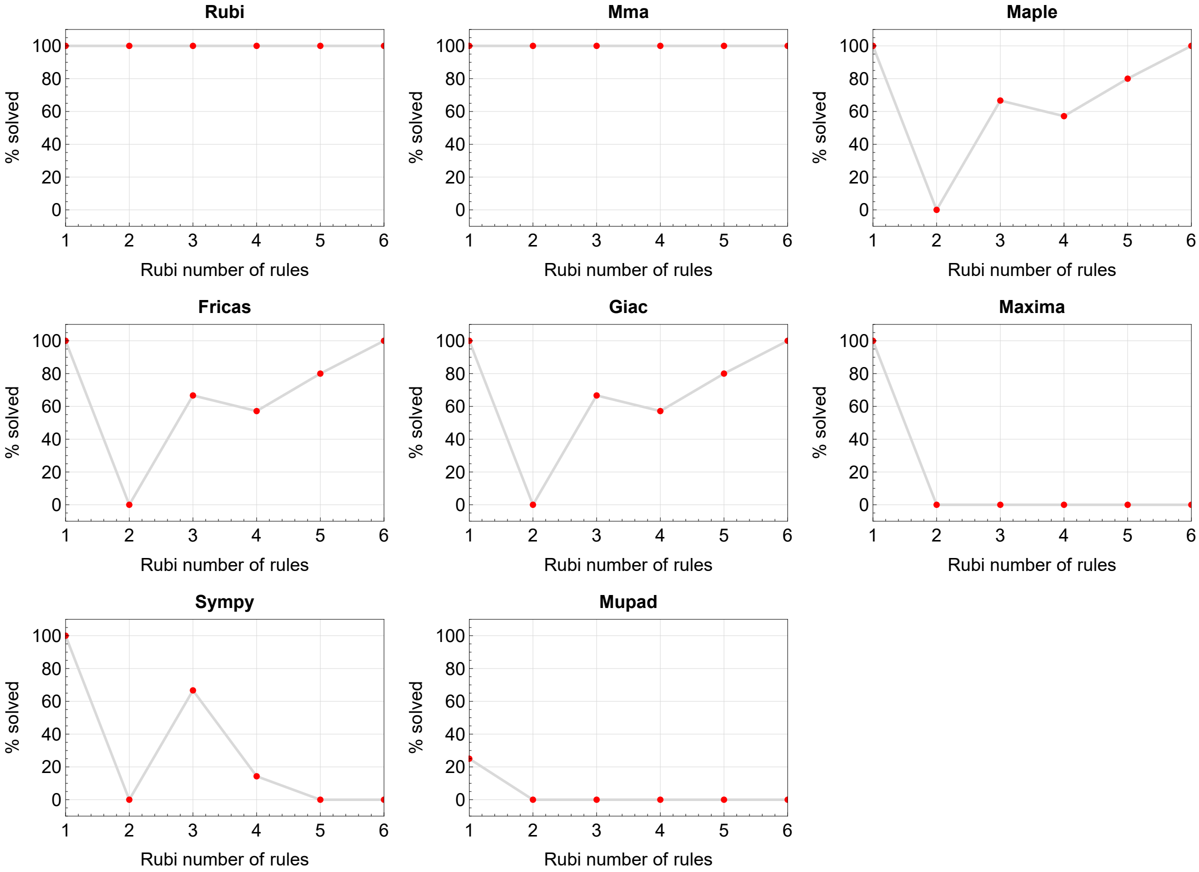


Figure 1.1: Solving statistics per number of Rubi rules used

1.5 Performance based on number of steps Rubi used

This section shows how each CAS performed based on the number of steps Rubi needed to solve the same integral. Note that the number of steps Rubi needed can be much higher than the number of rules, as the same rule could be used more than once.

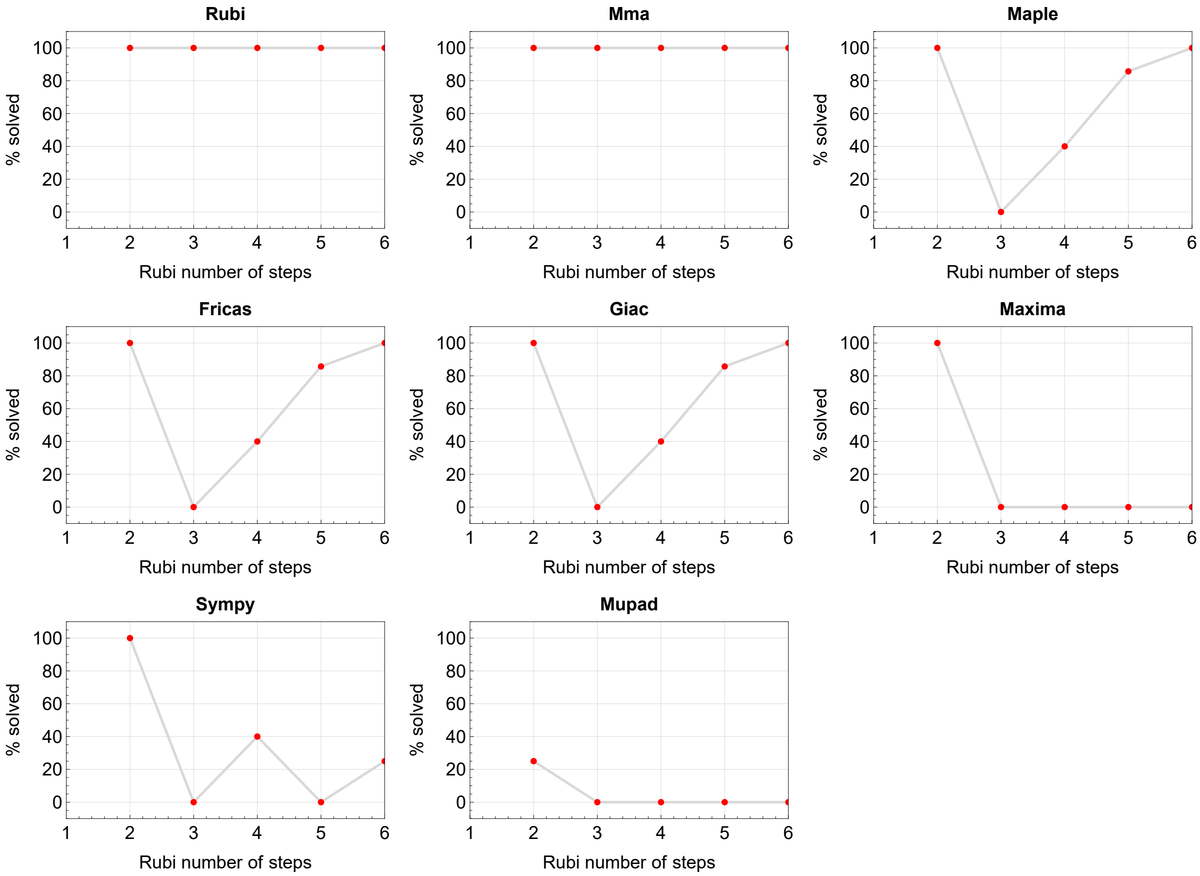
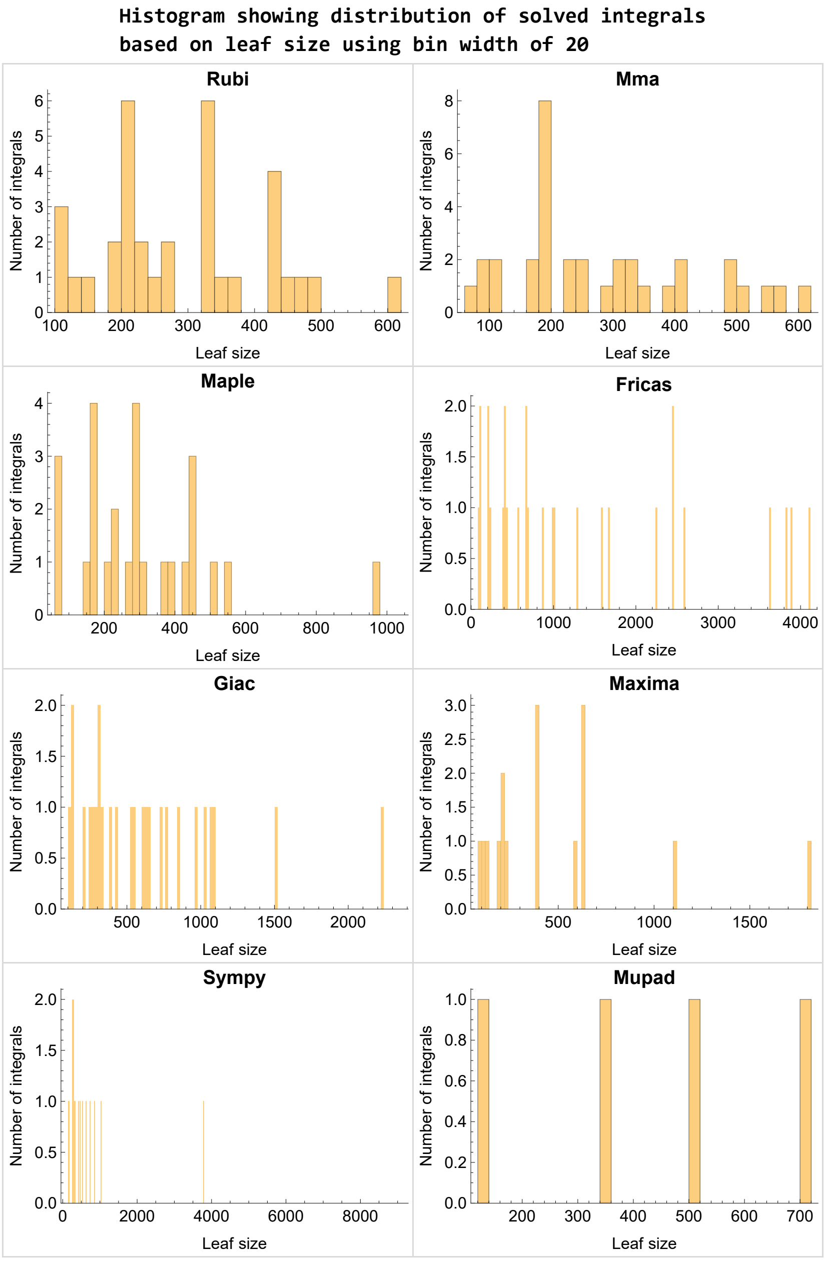


Figure 1.2: Solving statistics per number of Rubi steps used

The above diagram show that the precentage of solved intergals decreases for most CAS systems as the number of steps increases. As expected, for integrals that required less steps by Rubi, CAS systems had more success which indicates the integral was not as hard to solve. As Rubi needed more steps to solve the integral, the solved percentage decreased for most CAS systems which indicates the integral is becoming harder to solve.

1.6 Solved integrals histogram based on leaf size of result

The following shows the distribution of solved integrals for each CAS system based on leaf size of the antiderivatives produced by each CAS. It shows that most integrals solved produced leaf size less than about 100 to 150. The bin size used is 40.



Histogram showing distribution of solved integrals
based on leaf size using bin width of 20

Rubi

Mma

Maple

Fricas

Giac

Maxima

Sympy

Mupad

Figure 1.3: Solved integrals based on leaf size distribution

1.7 Solved integrals histogram based on CPU time used

The following shows the distribution of solved integrals for each CAS system based on CPU time used in seconds. The bin size used is 0.1 second.

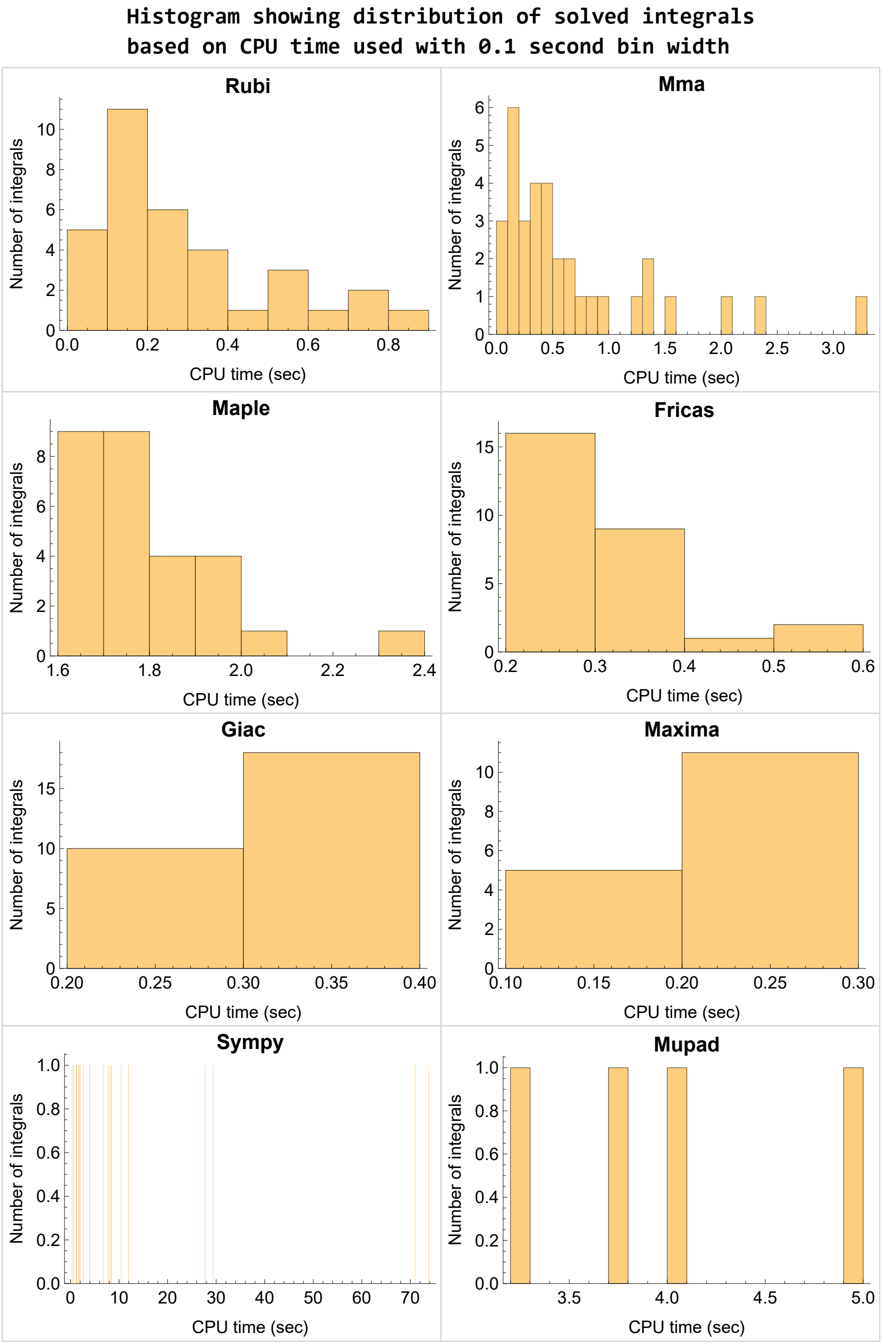


Figure 1.4: Solved integrals histogram based on CPU time used

1.8 Leaf size vs. CPU time used

The following shows the relation between the CPU time used to solve an integral and the leaf size of the antiderivative.

The result for Fricas, Maxima and Giac is shifted more to the right than the other CAS system due to the use of sagemath to call them, which causes an initial slight delay in the timing to start the integration due to overhead of starting a new process each time. This should also be taken into account when looking at the timing of these three CAS systems. Direct calls not using sagemath would result in faster timings, but current implementation uses sagemath as this makes testing much easier to do.

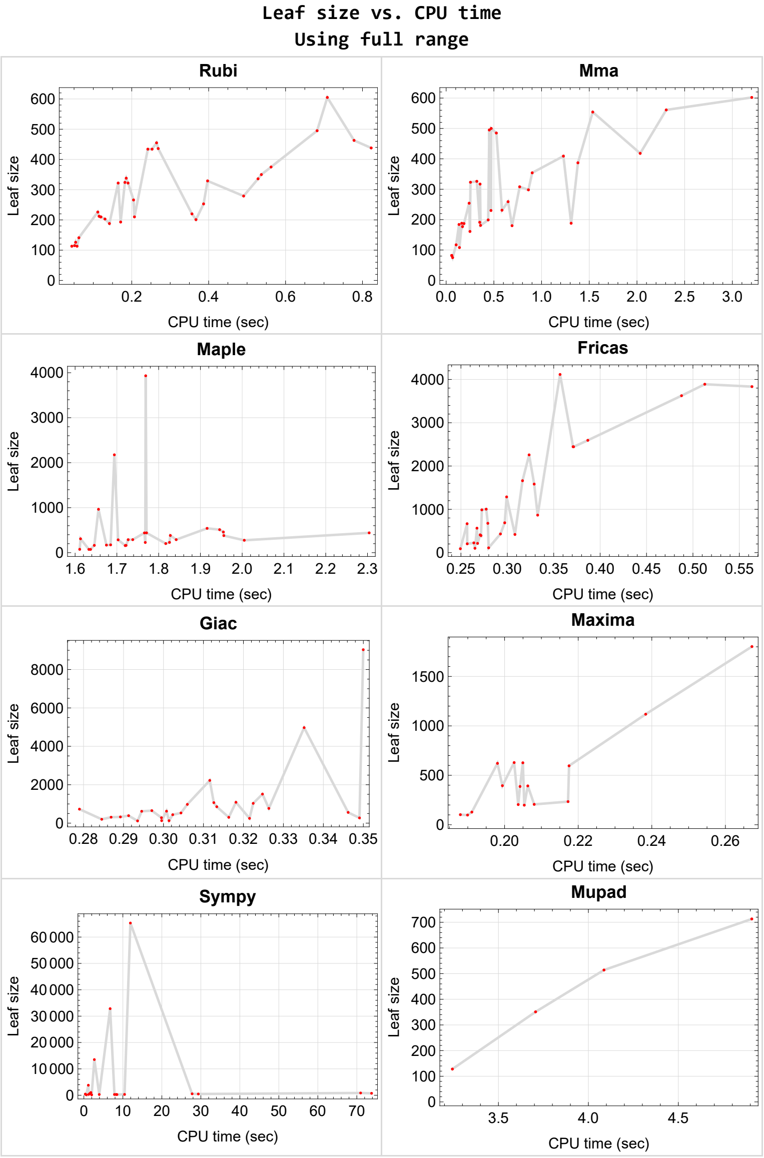


Figure 1.5: Leaf size vs. CPU time. Full range

1.9 list of integrals with no known antiderivative

{}

1.10 List of integrals solved by CAS but has no known antiderivative

Rubi {}

Mathematica {}

Maple {}

Maxima {}

Fricas {}

Sympy {}

Giac {}

Mupad {}

1.11 list of integrals solved by CAS but failed verification

The following are integrals solved by CAS but the verification phase failed to verify the anti-derivative produced is correct. This does not necessarily mean that the anti-derivative is wrong as additional methods of verification might be needed, or more time is needed (3 minutes time limit was used). These integrals are listed here to make it possible to do further investigation to determine why the result could not be verified.

Rubi {}

Mathematica {}

Maple {}

Maxima Verification phase not currently implemented.

Fricas Verification phase not currently implemented.

Sympy Verification phase not currently implemented.

Giac Verification phase not currently implemented.

Mupad Verification phase not currently implemented.

1.12 Timing

The command `AbsoluteTiming[]` was used in Mathematica to obtain the elapsed time for each integrate call. In Maple, the command `Usage` was used as in the following example

```
cpu_time := Usage(assign ('result_of_int',int(expr,x)),output='realtime'
```

For all other CAS systems, the elapsed time to complete each integral was found by taking the difference between the time after the call completed from the time before the call was made. This was done using Python's `time.time()` call.

All elapsed times shown are in seconds. A time limit of 3 CPU minutes was used for each integral. If the integrate command did not complete within this time limit, the integral was aborted and considered to have failed and assigned an F grade. The time used by failed integrals due to time out was not counted in the final statistics.

1.13 Verification

A verification phase was applied on the result of integration for `Rubi` and `Mathematica`.

Future version of this report will implement verification for the other CAS systems. For the integrals whose result was not run through a verification phase, it is assumed that the antiderivative was correct.

Verification phase also had 3 minutes time out. An integral whose result was not verified could still be correct, but further investigation is needed on those integrals. These integrals were marked in the summary table below and also in each integral separate section so they are easy to identify and locate.

1.14 Important notes about some of the results

Important note about Maxima results

Since tests were run in a batch mode, and using an automated script, then any integral where Maxima needed an interactive response from the user to answer a question during the evaluation of the integral will fail.

The exception raised is `ValueError`. Therefore Maxima results is lower than what would result if Maxima was run directly and each question was answered correctly.

The percentage of such failures were not counted for each test file, but for an example, for the `Timofeev` test file, there were about 14 such integrals out of total 705, or about 2 percent. This percentage can be higher or lower depending on the specific input test file.

Such integrals can be identified by looking at the output of the integration in each section for Maxima. The exception message will indicate the cause of error.

Maxima integrate was run using SageMath with the following settings set by default

```
'besselexpand : true'
'display2d : false'
'domain : complex'
'keepfloat : true'
'load(to_poly_solve)'
'load(simplify_sum)'
'load(abs_integrate)' 'load(diag)'
```

SageMath automatic loading of Maxima `abs_integrate` was found to cause some problems. So the following code was added to disable this effect.

```
from sage.interfaces.maxima_lib import maxima_lib
maxima_lib.set('extra_definite_integration_methods', '[]')
maxima_lib.set('extra_integration_methods', '[]')
```

See `https://ask.sagemath.org/question/43088/integrate-results-that-are-different-from-using-maxima/` for reference.

Important note about FriCAS result

There were few integrals which failed due to SageMath interface and not because FriCAS system could not do the integration.

These will fail With error `Exception raised: NotImplementedError`.

The number of such cases seems to be very small. About 1 or 2 percent of all integrals. These can be identified by looking at the exception message given in the result.

Important note about finding leaf size of antiderivative

For Mathematica, Rubi, and Maple, the builtin system function `LeafSize` was used to find the leaf size of each antiderivative.

The other CAS systems (SageMath and Sympy) do not have special builtin function for this purpose at this time. Therefore the leaf size for Fricas and Sympy antiderivative was determined using the following function, thanks to user `slelievre` at `https://ask.sagemath.org/question/57123/could-we-have-a-leaf_count-function-in-base-sagemath/`

```
def tree_size(expr):
    r"""
    Return the tree size of this expression.
    """
    if expr not in SR:
        # deal with lists, tuples, vectors
        return 1 + sum(tree_size(a) for a in expr)
    expr = SR(expr)
```

```
  x, aa = expr.operator(), expr.operands()
  if x is None:
     return 1
  else:
     return 1 + sum(tree_size(a) for a in aa)
```

For Sympy, which was called directly from Python, the following code was used to obtain the leafsize of its result

```
try:
 # 1.7 is a fudge factor since it is low side from actual leaf count
 leafCount = round(1.7*count_ops(anti))

 except Exception as ee:
       leafCount =1
```

Important note about Mupad results

Matlab's symbolic toolbox does not have a leaf count function to measure the size of the antiderivative. Maple was used to determine the leaf size of Mupad output by post processing Mupad result.

Currently no grading of the antiderivative for Mupad is implemented. If it can integrate the problem, it was assigned a B grade automatically as a placeholder. In the future, when grading function is implemented for Mupad, the tests will be rerun again.

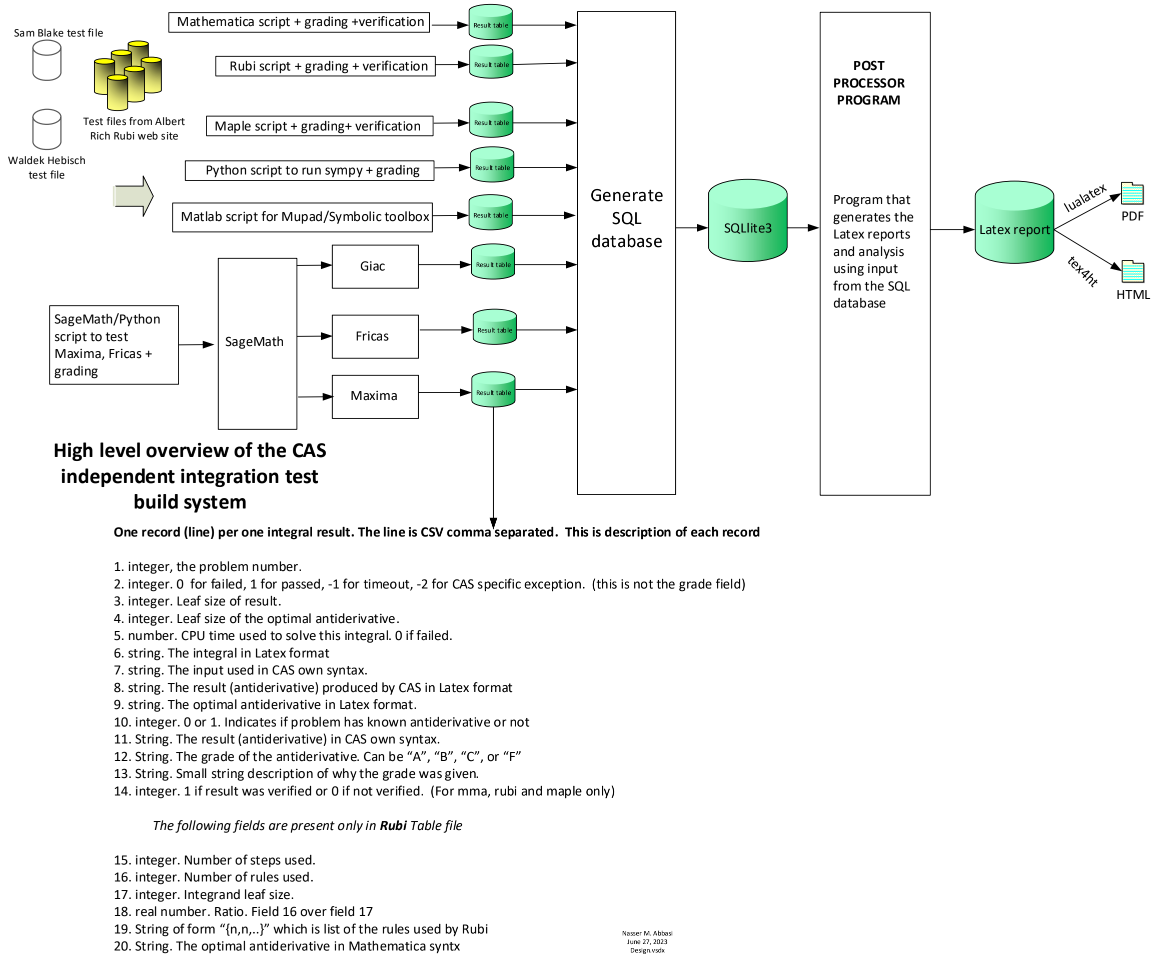
The following is an example of using Matlab's symbolic toolbox (Mupad) to solve an integral

```
integrand = evalin(symengine,'cos(x)*sin(x)')
the_variable = evalin(symengine,'x')
anti = int(integrand,the_variable)
```

Which gives `sin(x)^2/2`

1.15 Design of the test system

The following diagram gives a high level view of the current test build system.



High level overview of the CAS independent integration test build system

One record (line) per one integral result. The line is CSV comma separated. This is description of each record

1. integer, the problem number.
2. integer. 0 for failed, 1 for passed, -1 for timeout, -2 for CAS specific exception. (this is not the grade field)
3. integer. Leaf size of result.
4. integer. Leaf size of the optimal antiderivative.
5. number. CPU time used to solve this integral. 0 if failed.
6. string. The integral in Latex format
7. string. The input used in CAS own syntax.
8. string. The result (antiderivative) produced by CAS in Latex format
9. string. The optimal antiderivative in Latex format.
10. integer. 0 or 1. Indicates if problem has known antiderivative or not
11. String. The result (antiderivative) in CAS own syntax.
12. String. The grade of the antiderivative. Can be “A”, “B”, “C”, or “F”
13. String. Small string description of why the grade was given.
14. integer. 1 if result was verified or 0 if not verified. (For mma, rubi and maple only)

*The following fields are present only in **Rubi** Table file*

15. integer. Number of steps used.
16. integer. Number of rules used.
17. integer. Integrand leaf size.
18. real number. Ratio. Field 16 over field 17
19. String of form “{n,n,..}” which is list of the rules used by Rubi
20. String. The optimal antiderivative in Mathematica syntx

CHAPTER 2

DETAILED SUMMARY TABLES OF RESULTS

2.1 List of integrals sorted by grade for each CAS

Rubi

A grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34 }

B grade { }

C grade { }

F normal fail { }

F(-1) timedout fail { }

F(-2) exception fail { }

Mma

A grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34 }

B grade { }

C grade { }

F normal fail { }

F(-1) timedout fail { }

F(-2) exception fail { }

Maple

A grade $\{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 \}$

B grade $\{ 25, 26, 27, 28 \}$

C grade $\{ \}$

F normal fail $\{ 29, 30, 31, 32, 33, 34 \}$

F(-1) timedout fail $\{ \}$

F(-2) exception fail $\{ \}$

Fricas

A grade $\{ 1, 2, 3, 4, 5, 10, 11, 12, 13, 18, 19, 20, 21 \}$

B grade $\{ 6, 7, 8, 9, 14, 15, 16, 17, 22, 23, 24, 25, 26, 27, 28 \}$

C grade $\{ \}$

F normal fail $\{ 29, 30, 31, 32, 33, 34 \}$

F(-1) timedout fail $\{ \}$

F(-2) exception fail $\{ \}$

Maxima

A grade $\{ 1, 2, 3, 4, 10, 11, 12, 13, 18, 19, 20, 21, 28 \}$

B grade $\{ 25, 26, 27 \}$

C grade $\{ \}$

F normal fail $\{ 29, 30, 31, 32, 33, 34 \}$

F(-1) timedout fail $\{ \}$

F(-2) exception fail $\{ 5, 6, 7, 8, 9, 14, 15, 16, 17, 22, 23, 24 \}$

Giac

A grade $\{ 2, 3, 4, 5, 6, 12, 13, 14, 15, 16, 20, 21, 22, 23, 24 \}$

B grade $\{ 1, 7, 8, 9, 10, 11, 17, 18, 19, 25, 26, 27, 28 \}$

C grade $\{ \}$

F normal fail $\{ 29, 30, 31, 32, 33, 34 \}$

F(-1) timedout fail $\{ \}$

F(-2) exception fail $\{ \}$

Mupad

A grade $\{ \}$

B grade $\{ 1, 2, 3, 4 \}$

C grade $\{ \}$

F normal fail $\{ \}$

F(-1) timedout fail $\{ 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34 \}$

F(-2) exception fail $\{ \}$

Sympy

A grade $\{ 2, 3, 4, 5, 10, 11, 12, 13, 14, 18, 19, 20, 22 \}$

B grade $\{ 1, 21, 25, 26, 27, 28 \}$

C grade $\{ \}$

F normal fail $\{ 29, 31 \}$

F(-1) timedout fail $\{ 6, 7, 8, 9, 15, 16, 17, 23, 24 \}$

F(-2) exception fail $\{ 30, 32, 33, 34 \}$

2.2 Detailed conclusion table per each integral for all CAS systems

Detailed conclusion table per each integral is given by the table below. The elapsed time is in seconds. For failed result it is given as **F(-1)** if the failure was due to timeout. It is given as **F(-2)** if the failure was due to an exception being raised, which could indicate a bug in the system. If the failure was due to integral not being evaluated within the time limit, then it is given as **F**.

In this table, the column `N.S.` means **normalized size** and is defined as $\frac{\text{antiderivative leaf size}}{\text{optimal antiderivative leaf size}}$. To make the table fit the page, the name **`Mathematica`** was abbreviated to **`MMA`**.

Problem 1	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	436	436	495	440	621	668	1027	854	713
N.S.	1	1.00	1.14	1.01	1.42	1.53	2.36	1.96	1.64
time (sec)	N/A	0.269	0.452	2.304	0.198	0.257	1.795	0.313	4.908

Problem 2	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	324	324	323	285	387	410	639	558	514
N.S.	1	1.00	1.00	0.88	1.19	1.27	1.97	1.72	1.59
time (sec)	N/A	0.182	0.255	1.703	0.204	0.271	1.505	0.346	4.087

Problem 3	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	212	212	184	162	198	204	320	309	351
N.S.	1	1.00	0.87	0.76	0.93	0.96	1.51	1.46	1.66
time (sec)	N/A	0.115	0.135	1.646	0.205	0.257	1.248	0.287	3.706

Problem 4	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	115	115	82	72	128	90	163	128	128
N.S.	1	1.00	0.71	0.63	1.11	0.78	1.42	1.11	1.11
time (sec)	N/A	0.052	0.058	1.633	0.191	0.249	0.618	0.300	3.244

Problem 5	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	F(-2)	A	A	A	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	188	188	161	157	0	565	275	248	0
N.S.	1	1.00	0.86	0.84	0.00	3.01	1.46	1.32	0.00
time (sec)	N/A	0.142	0.250	1.720	0.000	0.267	3.942	0.322	0.000

Problem 6	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	F(-2)	B	F(-1)	A	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	201	201	231	226	0	1004	0	271	0
N.S.	1	1.00	1.15	1.12	0.00	5.00	0.00	1.35	0.00
time (sec)	N/A	0.367	0.585	1.768	0.000	0.277	0.000	0.349	0.000

Problem 7	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	F(-2)	B	F(-1)	B	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	279	279	298	289	0	1661	0	529	0
N.S.	1	1.00	1.07	1.04	0.00	5.95	0.00	1.90	0.00
time (sec)	N/A	0.491	0.862	1.842	0.000	0.316	0.000	0.304	0.000

Problem 8	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	F(-2)	B	F(-1)	B	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	375	375	409	382	0	2446	0	976	0
N.S.	1	1.00	1.09	1.02	0.00	6.52	0.00	2.60	0.00
time (sec)	N/A	0.562	1.229	1.828	0.000	0.371	0.000	0.306	0.000

Problem 9	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	F(-2)	B	F(-1)	B	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	495	495	602	540	0	3624	0	1512	0
N.S.	1	1.00	1.22	1.09	0.00	7.32	0.00	3.05	0.00
time (sec)	N/A	0.682	3.203	1.916	0.000	0.488	0.000	0.325	0.000

Problem 10	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	B	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	434	434	500	440	629	677	853	1067	0
N.S.	1	1.00	1.15	1.01	1.45	1.56	1.97	2.46	0.00
time (sec)	N/A	0.253	0.472	1.771	0.203	0.279	70.974	0.313	0.000

Problem 11	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	B	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	322	322	326	289	395	419	534	651	0
N.S.	1	1.00	1.01	0.90	1.23	1.30	1.66	2.02	0.00
time (sec)	N/A	0.190	0.321	1.738	0.199	0.308	27.760	0.297	0.000

Problem 12	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	210	210	188	173	206	213	280	323	0
N.S.	1	1.00	0.90	0.82	0.98	1.01	1.33	1.54	0.00
time (sec)	N/A	0.119	0.164	1.685	0.208	0.268	7.845	0.289	0.000

Problem 13	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	113	113	82	74	102	100	144	127	0
N.S.	1	1.00	0.73	0.65	0.90	0.88	1.27	1.12	0.00
time (sec)	N/A	0.058	0.059	1.611	0.188	0.265	1.995	0.301	0.000

Problem 14	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	F(-2)	B	A	A	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	193	193	191	161	0	866	262	200	0
N.S.	1	1.00	0.99	0.83	0.00	4.49	1.36	1.04	0.00
time (sec)	N/A	0.171	0.352	1.722	0.000	0.333	8.481	0.285	0.000

Problem 15	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	F(-2)	B	F(-1)	A	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	253	253	259	228	0	1583	0	388	0
N.S.	1	1.00	1.02	0.90	0.00	6.26	0.00	1.53	0.00
time (sec)	N/A	0.387	0.649	1.826	0.000	0.329	0.000	0.291	0.000

Problem 16	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	F(-2)	B	F(-1)	A	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	350	350	387	379	0	2594	0	617	0
N.S.	1	1.00	1.11	1.08	0.00	7.41	0.00	1.76	0.00
time (sec)	N/A	0.537	1.381	1.956	0.000	0.387	0.000	0.295	0.000

Problem 17	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	F(-2)	B	F(-1)	B	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	463	463	561	511	0	3834	0	1085	0
N.S.	1	1.00	1.21	1.10	0.00	8.28	0.00	2.34	0.00
time (sec)	N/A	0.778	2.307	1.946	0.000	0.564	0.000	0.318	0.000

Problem 18	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	B	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	434	434	485	439	627	689	729	1030	0
N.S.	1	1.00	1.12	1.01	1.44	1.59	1.68	2.37	0.00
time (sec)	N/A	0.242	0.526	1.766	0.205	0.297	73.782	0.322	0.000

Problem 19	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	B	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	322	322	317	288	393	431	476	622	0
N.S.	1	1.00	0.98	0.89	1.22	1.34	1.48	1.93	0.00
time (sec)	N/A	0.165	0.355	1.727	0.206	0.293	29.349	0.301	0.000

Problem 20	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	210	210	177	168	204	225	282	302	0
N.S.	1	1.00	0.84	0.80	0.97	1.07	1.34	1.44	0.00
time (sec)	N/A	0.120	0.171	1.675	0.204	0.264	8.266	0.316	0.000

Problem 21	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	B	A	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	113	113	75	72	98	110	425	115	0
N.S.	1	1.00	0.66	0.64	0.87	0.97	3.76	1.02	0.00
time (sec)	N/A	0.044	0.068	1.637	0.190	0.280	0.363	0.294	0.000

Problem 22	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	F(-2)	B	A	A	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	210	210	230	202	0	1287	306	281	0
N.S.	1	1.00	1.10	0.96	0.00	6.13	1.46	1.34	0.00
time (sec)	N/A	0.207	0.470	1.817	0.000	0.299	10.433	0.300	0.000

Problem 23	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	F(-2)	B	F(-1)	A	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	336	336	354	275	0	2444	0	439	0
N.S.	1	1.00	1.05	0.82	0.00	7.27	0.00	1.31	0.00
time (sec)	N/A	0.528	0.902	2.005	0.000	0.371	0.000	0.302	0.000

Problem 24	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	F(-2)	B	F(-1)	A	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	438	438	554	459	0	3889	0	767	0
N.S.	1	1.00	1.26	1.05	0.00	8.88	0.00	1.75	0.00
time (sec)	N/A	0.823	1.537	1.955	0.000	0.513	0.000	0.326	0.000

Problem 25	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	B	B	B	B	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	455	455	418	3932	1802	4115	65321	9032	0
N.S.	1	1.00	0.92	8.64	3.96	9.04	143.56	19.85	0.00
time (sec)	N/A	0.265	2.034	1.769	0.267	0.357	11.925	0.350	0.000

Problem 26	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	B	B	B	B	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	338	338	308	2175	1118	2258	32849	4972	0
N.S.	1	1.00	0.91	6.43	3.31	6.68	97.19	14.71	0.00
time (sec)	N/A	0.186	0.771	1.694	0.238	0.323	6.732	0.335	0.000

Problem 27	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	B	B	B	B	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	226	226	199	964	596	988	13522	2224	0
N.S.	1	1.00	0.88	4.27	2.64	4.37	59.83	9.84	0.00
time (sec)	N/A	0.112	0.441	1.656	0.218	0.273	2.682	0.312	0.000

Problem 28	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	A	B	B	B	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	126	126	108	308	234	394	3798	728	0
N.S.	1	1.00	0.86	2.44	1.86	3.13	30.14	5.78	0.00
time (sec)	N/A	0.054	0.139	1.613	0.217	0.272	1.146	0.279	0.000

Problem 29	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	203	203	181	0	0	0	0	0	0
N.S.	1	1.00	0.89	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.130	0.361	0.000	0.000	0.000	0.000	0.000	0.000

Problem 30	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F(-2)	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	220	220	180	0	0	0	0	0	0
N.S.	1	1.00	0.82	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.357	0.691	0.000	0.000	0.000	0.000	0.000	0.000

Problem 31	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	329	329	188	0	0	0	0	0	0
N.S.	1	1.00	0.57	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.397	1.310	0.000	0.000	0.000	0.000	0.000	0.000

Problem 32	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F(-2)	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	141	141	117	0	0	0	0	0	0
N.S.	1	1.00	0.83	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.063	0.105	0.000	0.000	0.000	0.000	0.000	0.000

Problem 33	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F(-2)	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	268	266	187	0	0	0	0	0	0
N.S.	1	0.99	0.70	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.205	0.187	0.000	0.000	0.000	0.000	0.000	0.000

Problem 34	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F(-2)	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	610	605	254	0	0	0	0	0	0
N.S.	1	0.99	0.42	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.709	0.241	0.000	0.000	0.000	0.000	0.000	0.000

2.3 Detailed conclusion table specific for Rubi results

The following table is specific to Rubi only. It gives additional statistics for each integral. the column **steps** is the number of steps used by Rubi to obtain the antiderivative. The **rules** column is the number of unique rules used. The **integrand size** column is the leaf size of the integrand. Finally the ratio $\frac{\text{number of rules}}{\text{integrand size}}$ is also given. The larger this ratio is, the harder the integral is to solve. In this test file, problem number [9] had the largest ratio of [.1875000000000000]

Table 2.1: Rubi specific breakdown of results for each integral

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
1	A	2	1	1.00	32	0.031
2	A	2	1	1.00	32	0.031
3	A	2	1	1.00	30	0.033
4	A	2	1	1.00	25	0.040
5	A	4	3	1.00	32	0.094
6	A	5	4	1.00	32	0.125
7	A	5	5	1.00	32	0.156
8	A	5	5	1.00	32	0.156
9	A	6	6	1.00	32	0.188
10	A	2	1	1.00	32	0.031
11	A	2	1	1.00	32	0.031
12	A	2	1	1.00	30	0.033
13	A	2	1	1.00	25	0.040
14	A	6	4	1.00	32	0.125
15	A	5	4	1.00	32	0.125
16	A	5	5	1.00	32	0.156
17	A	6	6	1.00	32	0.188
18	A	2	1	1.00	32	0.031
19	A	2	1	1.00	32	0.031
20	A	2	1	1.00	30	0.033
21	A	2	1	1.00	25	0.040
22	A	4	3	1.00	32	0.094
23	A	5	4	1.00	32	0.125
24	A	6	5	1.00	32	0.156

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
25	A	2	1	1.00	30	0.033
26	A	2	1	1.00	30	0.033
27	A	2	1	1.00	28	0.036
28	A	2	1	1.00	23	0.043
29	A	3	2	1.00	30	0.067
30	A	4	4	1.00	30	0.133
31	A	4	4	1.00	30	0.133
32	A	3	3	1.00	20	0.150
33	A	4	4	0.99	25	0.160
34	A	5	5	0.99	30	0.167

CHAPTER 3

LISTING OF INTEGRALS

3.1 $\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 436

$$
\begin{aligned}
&\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx \\
&= -\frac{2(bc-ad)^3\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)\sqrt{c+dx}}{d^7} \\
&\quad -\frac{2(bc-ad)^2\left(ad\left(2cCd-Bd^2-3c^2D\right)-b\left(5c^2Cd-4Bcd^2+3Ad^3-6c^3D\right)\right)(c+dx)^{3/2}}{3d^7} \\
&\quad -\frac{2(bc-ad)\left(a^2d^2(Cd-3cD)-abd\left(8cCd-3Bd^2-15c^2D\right)+b^2\left(10c^2Cd-6Bcd^2+3Ad^3-15c^3D\right)\right)(}{5d^7} \\
&\quad +\frac{2\left(a^3d^3D+3a^2bd^2(Cd-4cD)-3ab^2d\left(4cCd-Bd^2-10c^2D\right)+b^3\left(10c^2Cd-4Bcd^2+Ad^3-20c^3D\right)\right)}{7d^7} \\
&\quad +\frac{2b\left(3a^2d^2D+3abd(Cd-5cD)-b^2\left(5cCd-Bd^2-15c^2D\right)\right)(c+dx)^{9/2}}{9d^7} \\
&\quad +\frac{2b^2(bCd-6bcD+3adD)(c+dx)^{11/2}}{11d^7}+\frac{2b^3D(c+dx)^{13/2}}{13d^7}
\end{aligned}
$$

```
[Out] -2/3*(-a*d+b*c)^2*(a*d*(-B*d^2+2*C*c*d-3*D*c^2)-b*(3*A*d^3-4*B*c*d^2+5*C*c^
2*d-6*D*c^3))*(d*x+c)^(3/2)/d^7-2/5*(-a*d+b*c)*(a^2*d^2*(C*d-3*D*c)-a*b*d*(
-3*B*d^2+8*C*c*d-15*D*c^2)+b^2*(3*A*d^3-6*B*c*d^2+10*C*c^2*d-15*D*c^3))*(d*
x+c)^(5/2)/d^7+2/7*(a^3*d^3*D+3*a^2*b*d^2*(C*d-4*D*c)-3*a*b^2*d*(-B*d^2+4*C
*c*d-10*D*c^2)+b^3*(A*d^3-4*B*c*d^2+10*C*c^2*d-20*D*c^3))*(d*x+c)^(7/2)/d^7
+2/9*b*(3*a^2*d^2*D+3*a*b*d*(C*d-5*D*c)-b^2*(-B*d^2+5*C*c*d-15*D*c^2))*(d*x
+c)^(9/2)/d^7+2/11*b^2*(C*b*d+3*D*a*d-6*D*b*c)*(d*x+c)^(11/2)/d^7+2/13*b^3*
D*(d*x+c)^(13/2)/d^7-2*(-a*d+b*c)^3*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)*(d*x+c)^(
1/2)/d^7
```

Rubi [A] (verified)

Time = 0.27 (sec) , antiderivative size = 436, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.031, Rules used = {1634}

$$\int \frac{(a+bx)^3 (A+Bx+Cx^2+Dx^3)}{\sqrt{c+dx}}\,dx =$$

$$-\frac{2(c+dx)^{5/2}(bc-ad)\,(a^2d^2(Cd-3cD)-abd(-3Bd^2-15c^2D+8cCd)+b^2(3Ad^3-6Bcd^2-15c^3D+}{5d^7}$$

$$+\frac{2b(c+dx)^{9/2}\,(3a^2d^2D+3abd(Cd-5cD)-(b^2(-Bd^2-15c^2D+5cCd)))}{9d^7}$$

$$+\frac{2(c+dx)^{7/2}\,(a^3d^3D+3a^2bd^2(Cd-4cD)-3ab^2d(-Bd^2-10c^2D+4cCd)+b^3(Ad^3-4Bcd^2-20c^3D}{7d^7}$$

$$-\frac{2(c+dx)^{3/2}(bc-ad)^2\,(ad(-Bd^2-3c^2D+2cCd)-b(3Ad^3-4Bcd^2-6c^3D+5c^2Cd))}{3d^7}$$

$$-\frac{2\sqrt{c+dx}(bc-ad)^3\,(Ad^3-Bcd^2+c^3(-D)+c^2Cd)}{d^7}$$

$$+\frac{2b^2(c+dx)^{11/2}(3adD-6bcD+bCd)}{11d^7}+\frac{2b^3D(c+dx)^{13/2}}{13d^7}$$

```
[In] Int[((a + b*x)^3*(A + B*x + C*x^2 + D*x^3))/Sqrt[c + d*x],x]
```

```
[Out] (-2*(b*c - a*d)^3*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D)*Sqrt[c + d*x])/d^7 -
(2*(b*c - a*d)^2*(a*d*(2*c*C*d - B*d^2 - 3*c^2*D) - b*(5*c^2*C*d - 4*B*c*d^
2 + 3*A*d^3 - 6*c^3*D))*(c + d*x)^(3/2))/(3*d^7) - (2*(b*c - a*d)*(a^2*d^2*
(C*d - 3*c*D) - a*b*d*(8*c*C*d - 3*B*d^2 - 15*c^2*D) + b^2*(10*c^2*C*d - 6*
B*c*d^2 + 3*A*d^3 - 15*c^3*D))*(c + d*x)^(5/2))/(5*d^7) + (2*(a^3*d^3*D + 3
*a^2*b*d^2*(C*d - 4*c*D) - 3*a*b^2*d*(4*c*C*d - B*d^2 - 10*c^2*D) + b^3*(10
*c^2*C*d - 4*B*c*d^2 + A*d^3 - 20*c^3*D))*(c + d*x)^(7/2))/(7*d^7) + (2*b*(
3*a^2*d^2*D + 3*a*b*d*(C*d - 5*c*D) - b^2*(5*c*C*d - B*d^2 - 15*c^2*D))*(c
+ d*x)^(9/2))/(9*d^7) + (2*b^2*(b*C*d - 6*b*c*D + 3*a*d*D)*(c + d*x)^(11/2)
)/(11*d^7) + (2*b^3*D*(c + d*x)^(13/2))/(13*d^7)
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

$$\text{integral} = \int \left(\frac{(-bc+ad)^3\left(c^2Cd - Bcd^2 + Ad^3 - c^3D\right)}{d^6\sqrt{c+dx}} \right.$$

$$+ \frac{(bc-ad)^2\left(-ad(2cCd - Bd^2 - 3c^2D) + b(5c^2Cd - 4Bcd^2 + 3Ad^3 - 6c^3D)\right)\sqrt{c+dx}}{d^6}$$

$$+ \frac{(bc-ad)\left(-a^2d^2(Cd - 3cD) + abd(8cCd - 3Bd^2 - 15c^2D) - b^2(10c^2Cd - 6Bcd^2 + 3Ad^3 - 15c^3D)\right)(}{d^6}$$

$$+ \frac{\left(a^3d^3D + 3a^2bd^2(Cd - 4cD) - 3ab^2d(4cCd - Bd^2 - 10c^2D) + b^3(10c^2Cd - 4Bcd^2 + Ad^3 - 20c^3D)\right)(}{d^6}$$

$$+ \frac{b\left(3a^2d^2D + 3abd(Cd - 5cD) - b^2(5cCd - Bd^2 - 15c^2D)\right)(c+dx)^{7/2}}{d^6}$$

$$\left. + \frac{b^2(bCd - 6bcD + 3adD)(c+dx)^{9/2}}{d^6} + \frac{b^3D(c+dx)^{11/2}}{d^6} \right) dx$$

$$= -\frac{2(bc-ad)^3\left(c^2Cd - Bcd^2 + Ad^3 - c^3D\right)\sqrt{c+dx}}{d^7}$$

$$- \frac{2(bc-ad)^2\left(ad(2cCd - Bd^2 - 3c^2D) - b(5c^2Cd - 4Bcd^2 + 3Ad^3 - 6c^3D)\right)(c+dx)^{3/2}}{3d^7}$$

$$- \frac{2(bc-ad)\left(a^2d^2(Cd - 3cD) - abd(8cCd - 3Bd^2 - 15c^2D) + b^2(10c^2Cd - 6Bcd^2 + 3Ad^3 - 15\right.}{5d^7}$$

$$+ \frac{2\left(a^3d^3D + 3a^2bd^2(Cd - 4cD) - 3ab^2d(4cCd - Bd^2 - 10c^2D) + b^3(10c^2Cd - 4Bcd^2 + Ad^3 - 2\right.}{7d^7}$$

$$+ \frac{2b\left(3a^2d^2D + 3abd(Cd - 5cD) - b^2(5cCd - Bd^2 - 15c^2D)\right)(c+dx)^{9/2}}{9d^7}$$

$$+ \frac{2b^2(bCd - 6bcD + 3adD)(c+dx)^{11/2}}{11d^7} + \frac{2b^3D(c+dx)^{13/2}}{13d^7}$$

Mathematica [A] (verified)

Time = 0.45 (sec) , antiderivative size = 495, normalized size of antiderivative = 1.14

$$\int \frac{(a+bx)^3\left(A + Bx + Cx^2 + Dx^3\right)}{\sqrt{c+dx}}\,dx$$

$$= \frac{2\sqrt{c+dx}\left(429a^3d^3(-48c^3D + 8c^2d(7C + 3Dx) - 2cd^2(35B + x(14C + 9Dx)) + d^3(105A + x(35B + 3x(\right.}{}$$

```
[In] Integrate[((a + b*x)^3*(A + B*x + C*x^2 + D*x^3))/Sqrt[c + d*x],x]
```

```
[Out] (2*Sqrt[c + d*x]*(429*a^3*d^3*(-48*c^3*D + 8*c^2*d*(7*C + 3*D*x) - 2*c*d^2*
(35*B + x*(14*C + 9*D*x)) + d^3*(105*A + x*(35*B + 3*x*(7*C + 5*D*x)))) + 4
29*a^2*b*d^2*(128*c^4*D - 16*c^3*d*(9*C + 4*D*x) + 24*c^2*d^2*(7*B + x*(3*C
 + 2*D*x)) + d^4*x*(105*A + x*(63*B + 5*x*(9*C + 7*D*x))) - 2*c*d^3*(105*A
+ x*(42*B + x*(27*C + 20*D*x)))) + 39*a*b^2*d*(-1280*c^5*D + 128*c^4*d*(11*
```

```
C + 5*D*x) - 16*c^3*d^2*(99*B + 44*C*x + 30*D*x^2) + 8*c^2*d^3*(231*A + x*(
99*B + 66*C*x + 50*D*x^2)) + d^5*x^2*(693*A + 5*x*(99*B + 7*x*(11*C + 9*D*x
))) - 2*c*d^4*x*(462*A + x*(297*B + 5*x*(44*C + 35*D*x)))) + b^3*(15360*c^6
*D - 1280*c^5*d*(13*C + 6*D*x) - 16*c^3*d^3*(1287*A + 572*B*x + 390*C*x^2 +
 300*D*x^3) + 128*c^4*d^2*(143*B + 5*x*(13*C + 9*D*x)) + 5*d^6*x^3*(1287*A
+ 7*x*(143*B + 117*C*x + 99*D*x^2)) + 8*c^2*d^4*x*(1287*A + x*(858*B + 650*
C*x + 525*D*x^2)) - 2*c*d^5*x^2*(3861*A + 5*x*(572*B + 7*x*(65*C + 54*D*x))
))))/(45045*d^7)
```

Maple [A] (verified)

Time = 2.30 (sec) , antiderivative size = 440, normalized size of antiderivative = 1.01

method	result
derivativedivides	$\frac{\frac{2Db^3(dx+c)^{\frac{13}{2}}}{13}+\frac{2\left(3(ad-bc)b^2D+b^3(Cd-3Dc)\right)(dx+c)^{\frac{11}{2}}}{11}+\frac{2\left(3(ad-bc)^2bD+3(ad-bc)b^2(Cd-3Dc)+b^3\left(Bd^2-2Ccd+3Dc^2\right)\right)(dx+}{9}}{}$
default	$\frac{\frac{2Db^3(dx+c)^{\frac{13}{2}}}{13}+\frac{2\left(3(ad-bc)b^2D+b^3(Cd-3Dc)\right)(dx+c)^{\frac{11}{2}}}{11}+\frac{2\left(3(ad-bc)^2bD+3(ad-bc)b^2(Cd-3Dc)+b^3\left(Bd^2-2Ccd+3Dc^2\right)\right)(dx+}{9}}{}$
pseudoelliptic	$\frac{2\sqrt{dx+c}\Bigg(\bigg(\frac{x^3\left(\frac{7}{13}Dx^3+\frac{7}{11}Cx^2+\frac{7}{9}Bx+A\right)b^3}{7}+\frac{3ax^2\left(\frac{5}{11}Dx^3+\frac{5}{9}Cx^2+\frac{5}{7}Bx+A\right)b^2}{5}+a^2x(\frac{1}{3}Dx^3+\frac{3}{7}Cx^2+\frac{3}{5}Bx+A)b+a^3(A+}{}$
gosper	$\frac{2\sqrt{dx+c}\,(3465Db^3x^6d^6+4095Cb^3d^6x^5+12285Dab^2d^6x^5-3780Db^3cd^5x^5+5005Bb^3d^6x^4+15015Cab^2d^6x^4-4550Cb^3c}{}$
trager	$\frac{2\sqrt{dx+c}\,(3465Db^3x^6d^6+4095Cb^3d^6x^5+12285Dab^2d^6x^5-3780Db^3cd^5x^5+5005Bb^3d^6x^4+15015Cab^2d^6x^4-4550Cb^3c}{}$

```
[In] int((b*x+a)^3*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x,method=_RETURNVERBOSE)
```

```
[Out] 2/d^7*(1/13*D*b^3*(d*x+c)^(13/2)+1/11*(3*(a*d-b*c)*b^2*D+b^3*(C*d-3*D*c))*(
d*x+c)^(11/2)+1/9*(3*(a*d-b*c)^2*b*D+3*(a*d-b*c)*b^2*(C*d-3*D*c)+b^3*(B*d^2
-2*C*c*d+3*D*c^2))*(d*x+c)^(9/2)+1/7*((a*d-b*c)^3*D+3*(a*d-b*c)^2*b*(C*d-3*
D*c)+3*(a*d-b*c)*b^2*(B*d^2-2*C*c*d+3*D*c^2)+b^3*(A*d^3-B*c*d^2+C*c^2*d-D*c
^3))*(d*x+c)^(7/2)+1/5*((a*d-b*c)^3*(C*d-3*D*c)+3*(a*d-b*c)^2*b*(B*d^2-2*C*
c*d+3*D*c^2)+3*(a*d-b*c)*b^2*(A*d^3-B*c*d^2+C*c^2*d-D*c^3))*(d*x+c)^(5/2)+1
/3*((a*d-b*c)^3*(B*d^2-2*C*c*d+3*D*c^2)+3*(a*d-b*c)^2*b*(A*d^3-B*c*d^2+C*c^
2*d-D*c^3))*(d*x+c)^(3/2)+(a*d-b*c)^3*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)*(d*x+c)
^(1/2))
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 668, normalized size of antiderivative = 1.53

$$\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx$$
$$= \frac{2\,(3465\,Db^3d^6x^6+15360\,Db^3c^6+45045\,Aa^3d^6+24024\,(Ca^3+3\,Ba^2b+3\,Aab^2)c^2d^4-30030\,(Ba^3+3\,A}{}$$

```
[In] integrate((b*x+a)^3*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x, algorithm="fricas"
)
```

```
[Out] 2/45045*(3465*D*b^3*d^6*x^6 + 15360*D*b^3*c^6 + 45045*A*a^3*d^6 + 24024*(C*
a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^2*d^4 - 30030*(B*a^3 + 3*A*a^2*b)*c*d^5 - 31
5*(12*D*b^3*c*d^5 - 13*(3*D*a*b^2 + C*b^3)*d^6)*x^5 + 35*(120*D*b^3*c^2*d^4
 + 143*(3*D*a^2*b + 3*C*a*b^2 + B*b^3)*d^6 - 130*(3*D*a*b^2*c + C*b^3*c)*d^
5)*x^4 - 20592*(D*a^3*c^3 + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^3)*d^3 - 5*(9
60*D*b^3*c^3*d^3 - 1287*(D*a^3 + 3*C*a^2*b + 3*B*a*b^2 + A*b^3)*d^6 + 1144*
(3*D*a^2*b*c + (3*C*a*b^2 + B*b^3)*c)*d^5 - 1040*(3*D*a*b^2*c^2 + C*b^3*c^2
)*d^4)*x^3 + 18304*(3*D*a^2*b*c^4 + (3*C*a*b^2 + B*b^3)*c^4)*d^2 + 3*(1920*
D*b^3*c^4*d^2 + 3003*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*d^6 - 2574*(D*a^3*c +
(3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c)*d^5 + 2288*(3*D*a^2*b*c^2 + (3*C*a*b^2 +
 B*b^3)*c^2)*d^4 - 2080*(3*D*a*b^2*c^3 + C*b^3*c^3)*d^3)*x^2 - 16640*(3*D*a
*b^2*c^5 + C*b^3*c^5)*d - (7680*D*b^3*c^5*d + 12012*(C*a^3 + 3*B*a^2*b + 3*
A*a*b^2)*c*d^5 - 15015*(B*a^3 + 3*A*a^2*b)*d^6 - 10296*(D*a^3*c^2 + (3*C*a^
2*b + 3*B*a*b^2 + A*b^3)*c^2)*d^4 + 9152*(3*D*a^2*b*c^3 + (3*C*a*b^2 + B*b^
3)*c^3)*d^3 - 8320*(3*D*a*b^2*c^4 + C*b^3*c^4)*d^2)*x)*sqrt(d*x + c)/d^7
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1027 vs. 2(454) = 908.

Time = 1.80 (sec) , antiderivative size = 1027, normalized size of antiderivative = 2.36

$$\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx$$
$$= \begin{cases} \frac{2\left(\frac{Db^3(c+dx)^{\frac{13}{2}}}{13d^6}+\frac{(c+dx)^{\frac{11}{2}}\left(Cb^3d+3Dab^2d-6Db^3c\right)}{11d^6}+\frac{(c+dx)^{\frac{9}{2}}\left(Bb^3d^2+3Cab^2d^2-5Cb^3cd+3Da^2bd^2-15Dab^2cd+15Db^3c^2\right)}{9d^6}+\frac{(c+dx)^{\frac{7}{2}}(Ab^3d^3+3Bab^2}{}\right.}{} \\ \frac{Aa^3x+\frac{Db^3x^7}{7}+\frac{x^6\left(Cb^3+3Dab^2\right)}{6}+\frac{x^5\left(Bb^3+3Cab^2+3Da^2b\right)}{5}+\frac{x^4\left(Ab^3+3Bab^2+3Ca^2b+Da^3\right)}{4}+\frac{x^3\cdot\left(3Aab^2+3Ba^2b+Ca^3\right)}{3}+\frac{x^2\cdot\left(3Aa^2b+Ba^3\right)}{2}}{\sqrt{c}} \end{cases}$$

```
[In] integrate((b*x+a)**3*(D*x**3+C*x**2+B*x+A)/(d*x+c)**(1/2),x)
```

```
[Out] Piecewise((2*(D*b**3*(c + d*x)**(13/2)/(13*d**6) + (c + d*x)**(11/2)*(C*b**
3*d + 3*D*a*b**2*d - 6*D*b**3*c)/(11*d**6) + (c + d*x)**(9/2)*(B*b**3*d**2
+ 3*C*a*b**2*d**2 - 5*C*b**3*c*d + 3*D*a**2*b*d**2 - 15*D*a*b**2*c*d + 15*D
*b**3*c**2)/(9*d**6) + (c + d*x)**(7/2)*(A*b**3*d**3 + 3*B*a*b**2*d**3 - 4*
B*b**3*c*d**2 + 3*C*a**2*b*d**3 - 12*C*a*b**2*c*d**2 + 10*C*b**3*c**2*d + D
*a**3*d**3 - 12*D*a**2*b*c*d**2 + 30*D*a*b**2*c**2*d - 20*D*b**3*c**3)/(7*d
**6) + (c + d*x)**(5/2)*(3*A*a*b**2*d**4 - 3*A*b**3*c*d**3 + 3*B*a**2*b*d**
4 - 9*B*a*b**2*c*d**3 + 6*B*b**3*c**2*d**2 + C*a**3*d**4 - 9*C*a**2*b*c*d**
3 + 18*C*a*b**2*c**2*d**2 - 10*C*b**3*c**3*d - 3*D*a**3*c*d**3 + 18*D*a**2*
b*c**2*d**2 - 30*D*a*b**2*c**3*d + 15*D*b**3*c**4)/(5*d**6) + (c + d*x)**(3
/2)*(3*A*a**2*b*d**5 - 6*A*a*b**2*c*d**4 + 3*A*b**3*c**2*d**3 + B*a**3*d**5
 - 6*B*a**2*b*c*d**4 + 9*B*a*b**2*c**2*d**3 - 4*B*b**3*c**3*d**2 - 2*C*a**3
*c*d**4 + 9*C*a**2*b*c**2*d**3 - 12*C*a*b**2*c**3*d**2 + 5*C*b**3*c**4*d +
3*D*a**3*c**2*d**3 - 12*D*a**2*b*c**3*d**2 + 15*D*a*b**2*c**4*d - 6*D*b**3*
c**5)/(3*d**6) + sqrt(c + d*x)*(A*a**3*d**6 - 3*A*a**2*b*c*d**5 + 3*A*a*b**
2*c**2*d**4 - A*b**3*c**3*d**3 - B*a**3*c*d**5 + 3*B*a**2*b*c**2*d**4 - 3*B
*a*b**2*c**3*d**3 + B*b**3*c**4*d**2 + C*a**3*c**2*d**4 - 3*C*a**2*b*c**3*d
**3 + 3*C*a*b**2*c**4*d**2 - C*b**3*c**5*d - D*a**3*c**3*d**3 + 3*D*a**2*b*
c**4*d**2 - 3*D*a*b**2*c**5*d + D*b**3*c**6)/d**6)/d, Ne(d, 0)), ((A*a**3*x
 + D*b**3*x**7/7 + x**6*(C*b**3 + 3*D*a*b**2)/6 + x**5*(B*b**3 + 3*C*a*b**2
 + 3*D*a**2*b)/5 + x**4*(A*b**3 + 3*B*a*b**2 + 3*C*a**2*b + D*a**3)/4 + x**
3*(3*A*a*b**2 + 3*B*a**2*b + C*a**3)/3 + x**2*(3*A*a**2*b + B*a**3)/2)/sqrt
(c), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 621, normalized size of antiderivative = 1.42

$$\int \frac{(a+bx)^3 \left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx$$

$$= \frac{2\left(3465\,(dx+c)^{\frac{13}{2}}Db^3 - 4095\,(6\,Db^3c - (3\,Dab^2+Cb^3)d)(dx+c)^{\frac{11}{2}} + 5005\,(15\,Db^3c^2 - 5\,(3\,Dab^2+Cb^3)\right.}{}$$

```
[In] integrate((b*x+a)^3*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x, algorithm="maxima"
)
```

```
[Out] 2/45045*(3465*(d*x + c)^(13/2)*D*b^3 - 4095*(6*D*b^3*c - (3*D*a*b^2 + C*b^3
)*d)*(d*x + c)^(11/2) + 5005*(15*D*b^3*c^2 - 5*(3*D*a*b^2 + C*b^3)*c*d + (3
*D*a^2*b + 3*C*a*b^2 + B*b^3)*d^2)*(d*x + c)^(9/2) - 6435*(20*D*b^3*c^3 - 1
0*(3*D*a*b^2 + C*b^3)*c^2*d + 4*(3*D*a^2*b + 3*C*a*b^2 + B*b^3)*c*d^2 - (D*
a^3 + 3*C*a^2*b + 3*B*a*b^2 + A*b^3)*d^3)*(d*x + c)^(7/2) + 9009*(15*D*b^3*
c^4 - 10*(3*D*a*b^2 + C*b^3)*c^3*d + 6*(3*D*a^2*b + 3*C*a*b^2 + B*b^3)*c^2*
d^2 - 3*(D*a^3 + 3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c*d^3 + (C*a^3 + 3*B*a^2*b
```

```
+ 3*A*a*b^2)*d^4)*(d*x + c)^(5/2) - 15015*(6*D*b^3*c^5 - 5*(3*D*a*b^2 + C*b
^3)*c^4*d + 4*(3*D*a^2*b + 3*C*a*b^2 + B*b^3)*c^3*d^2 - 3*(D*a^3 + 3*C*a^2*
b + 3*B*a*b^2 + A*b^3)*c^2*d^3 + 2*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c*d^4 -
(B*a^3 + 3*A*a^2*b)*d^5)*(d*x + c)^(3/2) + 45045*(D*b^3*c^6 + A*a^3*d^6 - (
3*D*a*b^2 + C*b^3)*c^5*d + (3*D*a^2*b + 3*C*a*b^2 + B*b^3)*c^4*d^2 - (D*a^3
 + 3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^3*d^3 + (C*a^3 + 3*B*a^2*b + 3*A*a*b^2)
*c^2*d^4 - (B*a^3 + 3*A*a^2*b)*c*d^5)*sqrt(d*x + c))/d^7
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 854 vs. $2(412) = 824$.

Time = 0.31 (sec) , antiderivative size = 854, normalized size of antiderivative = 1.96

$$\int \frac{(a+bx)^3\,(A+Bx+Cx^2+Dx^3)}{\sqrt{c+dx}}\,dx = \text{Too large to display}$$

```
[In] integrate((b*x+a)^3*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x, algorithm="giac")
```

```
[Out] 2/45045*(45045*sqrt(d*x + c)*A*a^3 + 15015*((d*x + c)^(3/2) - 3*sqrt(d*x +
c)*c)*B*a^3/d + 45045*((d*x + c)^(3/2) - 3*sqrt(d*x + c)*c)*A*a^2*b/d + 300
3*(3*(d*x + c)^(5/2) - 10*(d*x + c)^(3/2)*c + 15*sqrt(d*x + c)*c^2)*C*a^3/d
^2 + 9009*(3*(d*x + c)^(5/2) - 10*(d*x + c)^(3/2)*c + 15*sqrt(d*x + c)*c^2)
*B*a^2*b/d^2 + 9009*(3*(d*x + c)^(5/2) - 10*(d*x + c)^(3/2)*c + 15*sqrt(d*x
 + c)*c^2)*A*a*b^2/d^2 + 1287*(5*(d*x + c)^(7/2) - 21*(d*x + c)^(5/2)*c + 3
5*(d*x + c)^(3/2)*c^2 - 35*sqrt(d*x + c)*c^3)*D*a^3/d^3 + 3861*(5*(d*x + c)
^(7/2) - 21*(d*x + c)^(5/2)*c + 35*(d*x + c)^(3/2)*c^2 - 35*sqrt(d*x + c)*c
^3)*C*a^2*b/d^3 + 3861*(5*(d*x + c)^(7/2) - 21*(d*x + c)^(5/2)*c + 35*(d*x
+ c)^(3/2)*c^2 - 35*sqrt(d*x + c)*c^3)*B*a*b^2/d^3 + 1287*(5*(d*x + c)^(7/2
) - 21*(d*x + c)^(5/2)*c + 35*(d*x + c)^(3/2)*c^2 - 35*sqrt(d*x + c)*c^3)*A
*b^3/d^3 + 429*(35*(d*x + c)^(9/2) - 180*(d*x + c)^(7/2)*c + 378*(d*x + c)^
(5/2)*c^2 - 420*(d*x + c)^(3/2)*c^3 + 315*sqrt(d*x + c)*c^4)*D*a^2*b/d^4 +
429*(35*(d*x + c)^(9/2) - 180*(d*x + c)^(7/2)*c + 378*(d*x + c)^(5/2)*c^2 -
 420*(d*x + c)^(3/2)*c^3 + 315*sqrt(d*x + c)*c^4)*C*a*b^2/d^4 + 143*(35*(d*
x + c)^(9/2) - 180*(d*x + c)^(7/2)*c + 378*(d*x + c)^(5/2)*c^2 - 420*(d*x +
 c)^(3/2)*c^3 + 315*sqrt(d*x + c)*c^4)*B*b^3/d^4 + 195*(63*(d*x + c)^(11/2)
 - 385*(d*x + c)^(9/2)*c + 990*(d*x + c)^(7/2)*c^2 - 1386*(d*x + c)^(5/2)*c
^3 + 1155*(d*x + c)^(3/2)*c^4 - 693*sqrt(d*x + c)*c^5)*D*a*b^2/d^5 + 65*(63
*(d*x + c)^(11/2) - 385*(d*x + c)^(9/2)*c + 990*(d*x + c)^(7/2)*c^2 - 1386*
(d*x + c)^(5/2)*c^3 + 1155*(d*x + c)^(3/2)*c^4 - 693*sqrt(d*x + c)*c^5)*C*b
^3/d^5 + 15*(231*(d*x + c)^(13/2) - 1638*(d*x + c)^(11/2)*c + 5005*(d*x + c
)^(9/2)*c^2 - 8580*(d*x + c)^(7/2)*c^3 + 9009*(d*x + c)^(5/2)*c^4 - 6006*(d
*x + c)^(3/2)*c^5 + 3003*sqrt(d*x + c)*c^6)*D*b^3/d^6)/d
```

Mupad [B] (verification not implemented)

Time = 4.91 (sec) , antiderivative size = 713, normalized size of antiderivative = 1.64

$$\int \frac{(a+bx)^3 (A+Bx+Cx^2+Dx^3)}{\sqrt{c+dx}}\,dx$$

$$= \frac{5544\,b^3\,c^6\,\sqrt{c+d\,x}\,D - 504\,b^3\,c\,(c+d\,x)^{11/2}\,D - 9240\,b^3\,c^5\,(c+d\,x)^{3/2}\,D + 11088\,b^3\,c^4\,(c+d\,x)^{5/2}\,D - 792}{3003\,d^7}$$

$$+ \frac{2\,C\,(c+d\,x)^{5/2}\,(a^3\,d^3 - 9\,a^2\,b\,c\,d^2 + 18\,a\,b^2\,c^2\,d - 10\,b^3\,c^3)}{5\,d^6} + \frac{2\,A\,b^3\,(c+d\,x)^{7/2}}{7\,d^4}$$

$$+ \frac{2\,B\,b^3\,(c+d\,x)^{9/2}}{9\,d^5} + \frac{2\,C\,b^3\,(c+d\,x)^{11/2}}{11\,d^6} + \frac{2\,A\,(a\,d-b\,c)^3\,\sqrt{c+d\,x}}{d^4}$$

$$+ \frac{2\,A\,b\,(a\,d-b\,c)^2\,(c+d\,x)^{3/2}}{d^4} + \frac{6\,A\,b^2\,(a\,d-b\,c)\,(c+d\,x)^{5/2}}{5\,d^4}$$

$$+ \frac{2\,B\,b^2\,(3\,a\,d-4\,b\,c)\,(c+d\,x)^{7/2}}{7\,d^5} - \frac{2\,B\,c\,(a\,d-b\,c)^3\,\sqrt{c+d\,x}}{d^5}$$

$$+ \frac{2\,C\,b^2\,(3\,a\,d-5\,b\,c)\,(c+d\,x)^{9/2}}{9\,d^6} + \frac{6\,B\,b\,(c+d\,x)^{5/2}\,(a^2\,d^2 - 3\,a\,b\,c\,d + 2\,b^2\,c^2)}{5\,d^5}$$

$$+ \frac{2\,C\,b\,(c+d\,x)^{7/2}\,(3\,a^2\,d^2 - 12\,a\,b\,c\,d + 10\,b^2\,c^2)}{7\,d^6}$$

$$+ \frac{2\,B\,(a\,d-b\,c)^2\,(a\,d-4\,b\,c)\,(c+d\,x)^{3/2}}{3\,d^5} + \frac{2\,C\,c^2\,(a\,d-b\,c)^3\,\sqrt{c+d\,x}}{d^6}$$

$$- \frac{2\,a^3\,\sqrt{c+d\,x}\,D\,\left(6\,c\,(c+d\,x)^2 - 20\,c^2\,(c+d\,x) + 30\,c^3 - 5\,d^3\,x^3\right)}{35\,d^4}$$

$$- \frac{2\,a\,b^2\,\sqrt{c+d\,x}\,D\,\left(70\,c\,(c+d\,x)^4 - 840\,c^4\,(c+d\,x) - 360\,c^2\,(c+d\,x)^3 + 756\,c^3\,(c+d\,x)^2 + 630\,c^5 - 63\,d\right.}{231\,d^6}$$

$$+ \frac{2\,a^2\,b\,\sqrt{c+d\,x}\,D\,\left(168\,c^2\,(c+d\,x)^2 - 280\,c^3\,(c+d\,x) - 40\,c\,(c+d\,x)^3 + 280\,c^4 + 35\,d^4\,x^4\right)}{105\,d^5}$$

$$- \frac{2\,C\,c\,(a\,d-b\,c)^2\,(2\,a\,d-5\,b\,c)\,(c+d\,x)^{3/2}}{3\,d^6}$$

```
[In] int(((a + b*x)^3*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(1/2),x)
```

```
[Out] (5544*b^3*c^6*(c + d*x)^(1/2)*D - 504*b^3*c*(c + d*x)^(11/2)*D - 9240*b^3*c
^5*(c + d*x)^(3/2)*D + 11088*b^3*c^4*(c + d*x)^(5/2)*D - 7920*b^3*c^3*(c +
d*x)^(7/2)*D + 3080*b^3*c^2*(c + d*x)^(9/2)*D + 462*b^3*d^6*x^6*(c + d*x)^(
1/2)*D)/(3003*d^7) + (2*C*(c + d*x)^(5/2)*(a^3*d^3 - 10*b^3*c^3 + 18*a*b^2*
c^2*d - 9*a^2*b*c*d^2))/(5*d^6) + (2*A*b^3*(c + d*x)^(7/2))/(7*d^4) + (2*B*
b^3*(c + d*x)^(9/2))/(9*d^5) + (2*C*b^3*(c + d*x)^(11/2))/(11*d^6) + (2*A*(
a*d - b*c)^3*(c + d*x)^(1/2))/d^4 + (2*A*b*(a*d - b*c)^2*(c + d*x)^(3/2))/d
^4 + (6*A*b^2*(a*d - b*c)*(c + d*x)^(5/2))/(5*d^4) + (2*B*b^2*(3*a*d - 4*b*
c)*(c + d*x)^(7/2))/(7*d^5) - (2*B*c*(a*d - b*c)^3*(c + d*x)^(1/2))/d^5 + (
2*C*b^2*(3*a*d - 5*b*c)*(c + d*x)^(9/2))/(9*d^6) + (6*B*b*(c + d*x)^(5/2)*(
```

```
a^2*d^2 + 2*b^2*c^2 - 3*a*b*c*d))/(5*d^5) + (2*C*b*(c + d*x)^(7/2)*(3*a^2*d
^2 + 10*b^2*c^2 - 12*a*b*c*d))/(7*d^6) + (2*B*(a*d - b*c)^2*(a*d - 4*b*c)*(
c + d*x)^(3/2))/(3*d^5) + (2*C*c^2*(a*d - b*c)^3*(c + d*x)^(1/2))/d^6 - (2*
a^3*(c + d*x)^(1/2)*D*(6*c*(c + d*x)^2 - 20*c^2*(c + d*x) + 30*c^3 - 5*d^3*
x^3))/(35*d^4) - (2*a*b^2*(c + d*x)^(1/2)*D*(70*c*(c + d*x)^4 - 840*c^4*(c
+ d*x) - 360*c^2*(c + d*x)^3 + 756*c^3*(c + d*x)^2 + 630*c^5 - 63*d^5*x^5))
/(231*d^6) + (2*a^2*b*(c + d*x)^(1/2)*D*(168*c^2*(c + d*x)^2 - 280*c^3*(c +
 d*x) - 40*c*(c + d*x)^3 + 280*c^4 + 35*d^4*x^4))/(105*d^5) - (2*C*c*(a*d -
 b*c)^2*(2*a*d - 5*b*c)*(c + d*x)^(3/2))/(3*d^6)
```

3.2 $\int \frac{(a+bx)^2(A+Bx+Cx^2+Dx^3)}{\sqrt{c+dx}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 324

$$\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx = \frac{2(bc-ad)^2\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)\sqrt{c+dx}}{d^6} + \frac{2(bc-ad)\left(ad(2cCd-Bd^2-3c^2D)-b(4c^2Cd-3Bcd^2+2Ad^3-5c^3D)\right)(c+dx)^{3/2}}{3d^6} + \frac{2\left(a^2d^2(Cd-3cD)-2abd(3cCd-Bd^2-6c^2D)+b^2(6c^2Cd-3Bcd^2+Ad^3-10c^3D)\right)(c+dx)^{5/2}}{5d^6} + \frac{2\left(a^2d^2D+2abd(Cd-4cD)-b^2(4cCd-Bd^2-10c^2D)\right)(c+dx)^{7/2}}{7d^6} + \frac{2b(bCd-5bcD+2adD)(c+dx)^{9/2}}{9d^6} + \frac{2b^2D(c+dx)^{11/2}}{11d^6}$$

```
[Out] 2/3*(-a*d+b*c)*(a*d*(-B*d^2+2*C*c*d-3*D*c^2)-b*(2*A*d^3-3*B*c*d^2+4*C*c^2*d
-5*D*c^3))*(d*x+c)^(3/2)/d^6+2/5*(a^2*d^2*(C*d-3*D*c)-2*a*b*d*(-B*d^2+3*C*c
*d-6*D*c^2)+b^2*(A*d^3-3*B*c*d^2+6*C*c^2*d-10*D*c^3))*(d*x+c)^(5/2)/d^6+2/7
*(a^2*d^2*D+2*a*b*d*(C*d-4*D*c)-b^2*(-B*d^2+4*C*c*d-10*D*c^2))*(d*x+c)^(7/2
)/d^6+2/9*b*(C*b*d+2*D*a*d-5*D*b*c)*(d*x+c)^(9/2)/d^6+2/11*b^2*D*(d*x+c)^(1
1/2)/d^6+2*(-a*d+b*c)^2*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)*(d*x+c)^(1/2)/d^6
```

Rubi [A] (verified)

Time = 0.18 (sec) , antiderivative size = 324, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.031, Rules used = {1634}

$$\begin{aligned}
&\int \frac{(a+bx)^2 (A+Bx+Cx^2+Dx^3)}{\sqrt{c+dx}} \, dx \\
&= \frac{2(c+dx)^{5/2} \left(a^2d^2(Cd-3cD)-2abd(-Bd^2-6c^2D+3cCd)+b^2(Ad^3-3Bcd^2-10c^3D+6c^2Cd)\right)}{5d^6} \\
&\quad + \frac{2(c+dx)^{7/2} \left(a^2d^2D+2abd(Cd-4cD)-(b^2(-Bd^2-10c^2D+4cCd))\right)}{7d^6} \\
&\quad + \frac{2(c+dx)^{3/2}(bc-ad) \left(ad(-Bd^2-3c^2D+2cCd)-b(2Ad^3-3Bcd^2-5c^3D+4c^2Cd)\right)}{3d^6} \\
&\quad + \frac{2\sqrt{c+dx}(bc-ad)^2 \left(Ad^3-Bcd^2+c^3(-D)+c^2Cd\right)}{d^6} \\
&\quad + \frac{2b(c+dx)^{9/2}(2adD-5bcD+bCd)}{9d^6}+\frac{2b^2D(c+dx)^{11/2}}{11d^6}
\end{aligned}$$

```
[In] Int[((a + b*x)^2*(A + B*x + C*x^2 + D*x^3))/Sqrt[c + d*x],x]
```

```
[Out] (2*(b*c - a*d)^2*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D)*Sqrt[c + d*x])/d^6 + (
2*(b*c - a*d)*(a*d*(2*c*C*d - B*d^2 - 3*c^2*D) - b*(4*c^2*C*d - 3*B*c*d^2 +
 2*A*d^3 - 5*c^3*D))*(c + d*x)^(3/2))/(3*d^6) + (2*(a^2*d^2*(C*d - 3*c*D) -
 2*a*b*d*(3*c*C*d - B*d^2 - 6*c^2*D) + b^2*(6*c^2*C*d - 3*B*c*d^2 + A*d^3 -
 10*c^3*D))*(c + d*x)^(5/2))/(5*d^6) + (2*(a^2*d^2*D + 2*a*b*d*(C*d - 4*c*D
) - b^2*(4*c*C*d - B*d^2 - 10*c^2*D))*(c + d*x)^(7/2))/(7*d^6) + (2*b*(b*C*
d - 5*b*c*D + 2*a*d*D)*(c + d*x)^(9/2))/(9*d^6) + (2*b^2*D*(c + d*x)^(11/2)
)/(11*d^6)
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

$$\begin{aligned}
\text{integral} = \int \Bigg(& \frac{(-bc+ad)^2 \left(c^2Cd - Bcd^2 + Ad^3 - c^3D\right)}{d^5\sqrt{c+dx}} \\
& + \frac{(bc-ad)\left(ad(2cCd - Bd^2 - 3c^2D) - b(4c^2Cd - 3Bcd^2 + 2Ad^3 - 5c^3D)\right)\sqrt{c+dx}}{d^5} \\
& + \frac{\left(a^2d^2(Cd - 3cD) - 2abd(3cCd - Bd^2 - 6c^2D) + b^2(6c^2Cd - 3Bcd^2 + Ad^3 - 10c^3D)\right)(c+dx)^{3/2}}{d^5} \\
& + \frac{\left(a^2d^2D + 2abd(Cd - 4cD) - b^2(4cCd - Bd^2 - 10c^2D)\right)(c+dx)^{5/2}}{d^5} \\
& + \frac{b(bCd - 5bcD + 2adD)(c+dx)^{7/2}}{d^5} + \frac{b^2D(c+dx)^{9/2}}{d^5}\Bigg)\,dx \\
= & \frac{2(bc-ad)^2 \left(c^2Cd - Bcd^2 + Ad^3 - c^3D\right)\sqrt{c+dx}}{d^6} \\
& + \frac{2(bc-ad)\left(ad(2cCd - Bd^2 - 3c^2D) - b(4c^2Cd - 3Bcd^2 + 2Ad^3 - 5c^3D)\right)(c+dx)^{3/2}}{3d^6} \\
& + \frac{2\left(a^2d^2(Cd - 3cD) - 2abd(3cCd - Bd^2 - 6c^2D) + b^2(6c^2Cd - 3Bcd^2 + Ad^3 - 10c^3D)\right)(c+dx)}{5d^6} \\
& + \frac{2\left(a^2d^2D + 2abd(Cd - 4cD) - b^2(4cCd - Bd^2 - 10c^2D)\right)(c+dx)^{7/2}}{7d^6} \\
& + \frac{2b(bCd - 5bcD + 2adD)(c+dx)^{9/2}}{9d^6} + \frac{2b^2D(c+dx)^{11/2}}{11d^6}
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.25 (sec) , antiderivative size = 323, normalized size of antiderivative = 1.00

$$\int \frac{(a+bx)^2 \left(A + Bx + Cx^2 + Dx^3\right)}{\sqrt{c+dx}}\,dx$$

$$= \frac{2\sqrt{c+dx}(33a^2d^2(-48c^3D + 8c^2d(7C + 3Dx) - 2cd^2(35B + x(14C + 9Dx)) + d^3(105A + x(35B + 3x(7C}{}$$

```
[In] Integrate[((a + b*x)^2*(A + B*x + C*x^2 + D*x^3))/Sqrt[c + d*x],x]
```

```
[Out] (2*Sqrt[c + d*x]*(33*a^2*d^2*(-48*c^3*D + 8*c^2*d*(7*C + 3*D*x) - 2*c*d^2*(
35*B + x*(14*C + 9*D*x)) + d^3*(105*A + x*(35*B + 3*x*(7*C + 5*D*x)))) + 22
*a*b*d*(128*c^4*D - 16*c^3*d*(9*C + 4*D*x) + 24*c^2*d^2*(7*B + x*(3*C + 2*D
*x)) + d^4*x*(105*A + x*(63*B + 5*x*(9*C + 7*D*x))) - 2*c*d^3*(105*A + x*(4
2*B + x*(27*C + 20*D*x)))) + b^2*(-1280*c^5*D + 128*c^4*d*(11*C + 5*D*x) -
16*c^3*d^2*(99*B + 44*C*x + 30*D*x^2) + 8*c^2*d^3*(231*A + x*(99*B + 66*C*x
 + 50*D*x^2)) + d^5*x^2*(693*A + 5*x*(99*B + 7*x*(11*C + 9*D*x))) - 2*c*d^4
*x*(462*A + x*(297*B + 5*x*(44*C + 35*D*x))))))/(3465*d^6)
```

Maple [A] (verified)

Time = 1.70 (sec) , antiderivative size = 285, normalized size of antiderivative = 0.88

method	result
pseudoelliptic	$2\sqrt{dx+c}\left(\left(\frac{x^2\left(\frac{5}{11}Dx^3+\frac{5}{9}C\,x^2+\frac{5}{7}Bx+A\right)b^2}{5}+\frac{2ax\left(\frac{1}{3}Dx^3+\frac{3}{7}C\,x^2+\frac{3}{5}Bx+A\right)b}{3}+a^2\left(A+\frac{1}{7}Dx^3+\frac{1}{5}C\,x^2+\frac{1}{3}Bx\right)\right)d^5-\right.$
derivativedivides	$\frac{2Db^2(dx+c)^{\frac{11}{2}}}{11}+\frac{2\left(2(ad-bc)bD+b^2(Cd-3Dc)\right)(dx+c)^{\frac{9}{2}}}{9}+\frac{2\left((ad-bc)^2D+2(ad-bc)b(Cd-3Dc)+b^2\left(B\,d^2-2Ccd+3Dc^2\right)\right)(dx+c)^{\frac{7}{2}}}{7}$
default	$\frac{2Db^2(dx+c)^{\frac{11}{2}}}{11}+\frac{2\left(2(ad-bc)bD+b^2(Cd-3Dc)\right)(dx+c)^{\frac{9}{2}}}{9}+\frac{2\left((ad-bc)^2D+2(ad-bc)b(Cd-3Dc)+b^2\left(B\,d^2-2Ccd+3Dc^2\right)\right)(dx+c)^{\frac{7}{2}}}{7}$
gosper	$2\sqrt{dx+c}\,(315Db^2x^5d^5+385C\,b^2d^5x^4+770Dab\,d^5x^4-350Db^2c\,d^4x^4+495B\,b^2d^5x^3+990Cab\,d^5x^3-440C\,b^2c\,d^4x^3+495$
trager	$2\sqrt{dx+c}\,(315Db^2x^5d^5+385C\,b^2d^5x^4+770Dab\,d^5x^4-350Db^2c\,d^4x^4+495B\,b^2d^5x^3+990Cab\,d^5x^3-440C\,b^2c\,d^4x^3+495$

```
[In] int((b*x+a)^2*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x,method=_RETURNVERBOSE)
```

```
[Out] 2*(d*x+c)^(1/2)*((1/5*x^2*(5/11*D*x^3+5/9*C*x^2+5/7*B*x+A)*b^2+2/3*a*x*(1/3
*D*x^3+3/7*C*x^2+3/5*B*x+A)*b+a^2*(A+1/7*D*x^3+1/5*C*x^2+1/3*B*x))*d^5-4/3*
(1/5*x*(25/66*D*x^3+10/21*C*x^2+9/14*B*x+A)*b^2+a*(4/21*D*x^3+9/35*C*x^2+2/
5*B*x+A)*b+1/2*a^2*(9/35*D*x^2+2/5*C*x+B))*c*d^4+8/15*((50/231*D*x^3+2/7*C*
x^2+3/7*B*x+A)*b^2+2*a*(2/7*D*x^2+3/7*C*x+B)*b+a^2*(3/7*D*x+C))*c^2*d^3-16/
35*c^3*((10/33*D*x^2+4/9*C*x+B)*b^2+2*a*(4/9*D*x+C)*b+D*a^2)*d^2+128/315*((
5/11*D*x+C)*b+2*D*a)*b*c^4*d-256/693*D*b^2*c^5)/d^6
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 410, normalized size of antiderivative = 1.27

$$\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx$$

$$= 2\,(315\,Db^2d^5x^5-1280\,Db^2c^5+3465\,Aa^2d^5+1848\,(Ca^2+2\,Bab+Ab^2)c^2d^3-2310\,(Ba^2+2\,Aab)cd^4-$$

```
[In] integrate((b*x+a)^2*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x, algorithm="fricas"
)
```

```
[Out] 2/3465*(315*D*b^2*d^5*x^5 - 1280*D*b^2*c^5 + 3465*A*a^2*d^5 + 1848*(C*a^2 +
 2*B*a*b + A*b^2)*c^2*d^3 - 2310*(B*a^2 + 2*A*a*b)*c*d^4 - 35*(10*D*b^2*c*d
^4 - 11*(2*D*a*b + C*b^2)*d^5)*x^4 + 5*(80*D*b^2*c^2*d^3 + 99*(D*a^2 + 2*C*
a*b + B*b^2)*d^5 - 88*(2*D*a*b*c + C*b^2*c)*d^4)*x^3 - 1584*(D*a^2*c^3 + (2
```

```
*C*a*b + B*b^2)*c^3)*d^2 - 3*(160*D*b^2*c^3*d^2 - 231*(C*a^2 + 2*B*a*b + A*
b^2)*d^5 + 198*(D*a^2*c + (2*C*a*b + B*b^2)*c)*d^4 - 176*(2*D*a*b*c^2 + C*b
^2*c^2)*d^3)*x^2 + 1408*(2*D*a*b*c^4 + C*b^2*c^4)*d + (640*D*b^2*c^4*d - 92
4*(C*a^2 + 2*B*a*b + A*b^2)*c*d^4 + 1155*(B*a^2 + 2*A*a*b)*d^5 + 792*(D*a^2
*c^2 + (2*C*a*b + B*b^2)*c^2)*d^3 - 704*(2*D*a*b*c^3 + C*b^2*c^3)*d^2)*x)*s
qrt(d*x + c)/d^6
```

Sympy [A] (verification not implemented)

Time = 1.50 (sec) , antiderivative size = 639, normalized size of antiderivative = 1.97

$$\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx$$

$$= \begin{cases} 2\left(\frac{Db^2(c+dx)^{\frac{11}{2}}}{11d^5}+\frac{(c+dx)^{\frac{9}{2}}\left(Cb^2d+2Dabd-5Db^2c\right)}{9d^5}+\frac{(c+dx)^{\frac{7}{2}}\left(Bb^2d^2+2Cabd^2-4Cb^2cd+Da^2d^2-8Dabcd+10Db^2c^2\right)}{7d^5}+\frac{(c+dx)^{\frac{5}{2}}\left(Ab^2d^3+2Babd^3-3Bb^2c\right.}{} \right. \\ \frac{Aa^2x+\frac{Db^2x^6}{6}+\frac{x^5\left(Cb^2+2Dab\right)}{5}+\frac{x^4\left(Bb^2+2Cab+Da^2\right)}{4}+\frac{x^3\left(Ab^2+2Bab+Ca^2\right)}{3}+\frac{x^2\cdot\left(2Aab+Ba^2\right)}{2}}{\sqrt{c}} \end{cases}$$

```
[In] integrate((b*x+a)**2*(D*x**3+C*x**2+B*x+A)/(d*x+c)**(1/2),x)
```

```
[Out] Piecewise((2*(D*b**2*(c + d*x)**(11/2)/(11*d**5) + (c + d*x)**(9/2)*(C*b**2
*d + 2*D*a*b*d - 5*D*b**2*c)/(9*d**5) + (c + d*x)**(7/2)*(B*b**2*d**2 + 2*C
*a*b*d**2 - 4*C*b**2*c*d + D*a**2*d**2 - 8*D*a*b*c*d + 10*D*b**2*c**2)/(7*d
**5) + (c + d*x)**(5/2)*(A*b**2*d**3 + 2*B*a*b*d**3 - 3*B*b**2*c*d**2 + C*a
**2*d**3 - 6*C*a*b*c*d**2 + 6*C*b**2*c**2*d - 3*D*a**2*c*d**2 + 12*D*a*b*c*
*2*d - 10*D*b**2*c**3)/(5*d**5) + (c + d*x)**(3/2)*(2*A*a*b*d**4 - 2*A*b**2
*c*d**3 + B*a**2*d**4 - 4*B*a*b*c*d**3 + 3*B*b**2*c**2*d**2 - 2*C*a**2*c*d*
*3 + 6*C*a*b*c**2*d**2 - 4*C*b**2*c**3*d + 3*D*a**2*c**2*d**2 - 8*D*a*b*c**
3*d + 5*D*b**2*c**4)/(3*d**5) + sqrt(c + d*x)*(A*a**2*d**5 - 2*A*a*b*c*d**4
 + A*b**2*c**2*d**3 - B*a**2*c*d**4 + 2*B*a*b*c**2*d**3 - B*b**2*c**3*d**2
+ C*a**2*c**2*d**3 - 2*C*a*b*c**3*d**2 + C*b**2*c**4*d - D*a**2*c**3*d**2 +
 2*D*a*b*c**4*d - D*b**2*c**5)/d**5)/d, Ne(d, 0)), ((A*a**2*x + D*b**2*x**6
/6 + x**5*(C*b**2 + 2*D*a*b)/5 + x**4*(B*b**2 + 2*C*a*b + D*a**2)/4 + x**3*
(A*b**2 + 2*B*a*b + C*a**2)/3 + x**2*(2*A*a*b + B*a**2)/2)/sqrt(c), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 387, normalized size of antiderivative = 1.19

$$\int \frac{(a+bx)^2\,(A+Bx+Cx^2+Dx^3)}{\sqrt{c+dx}}\,dx$$

$$=\frac{2\left(315\,(dx+c)^{\frac{11}{2}}Db^2-385\,(5\,Db^2c-(2\,Dab+Cb^2)d)(dx+c)^{\frac{9}{2}}+495\,(10\,Db^2c^2-4\,(2\,Dab+Cb^2)cd+\right.}{}$$

```
[In] integrate((b*x+a)^2*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x, algorithm="maxima"
)
```

```
[Out] 2/3465*(315*(d*x + c)^(11/2)*D*b^2 - 385*(5*D*b^2*c - (2*D*a*b + C*b^2)*d)*
(d*x + c)^(9/2) + 495*(10*D*b^2*c^2 - 4*(2*D*a*b + C*b^2)*c*d + (D*a^2 + 2*
C*a*b + B*b^2)*d^2)*(d*x + c)^(7/2) - 693*(10*D*b^2*c^3 - 6*(2*D*a*b + C*b^
2)*c^2*d + 3*(D*a^2 + 2*C*a*b + B*b^2)*c*d^2 - (C*a^2 + 2*B*a*b + A*b^2)*d^
3)*(d*x + c)^(5/2) + 1155*(5*D*b^2*c^4 - 4*(2*D*a*b + C*b^2)*c^3*d + 3*(D*a
^2 + 2*C*a*b + B*b^2)*c^2*d^2 - 2*(C*a^2 + 2*B*a*b + A*b^2)*c*d^3 + (B*a^2
+ 2*A*a*b)*d^4)*(d*x + c)^(3/2) - 3465*(D*b^2*c^5 - A*a^2*d^5 - (2*D*a*b +
C*b^2)*c^4*d + (D*a^2 + 2*C*a*b + B*b^2)*c^3*d^2 - (C*a^2 + 2*B*a*b + A*b^2
)*c^2*d^3 + (B*a^2 + 2*A*a*b)*c*d^4)*sqrt(d*x + c))/d^6
```

Giac [A] (verification not implemented)

none

Time = 0.35 (sec) , antiderivative size = 558, normalized size of antiderivative = 1.72

$$\int \frac{(a+bx)^2\,(A+Bx+Cx^2+Dx^3)}{\sqrt{c+dx}}\,dx$$

$$=\frac{2\left(3465\,\sqrt{dx+c}Aa^2+\frac{1155\,\left((dx+c)^{\frac{3}{2}}-3\,\sqrt{dx+c}c\right)Ba^2}{d}+\frac{2310\,\left((dx+c)^{\frac{3}{2}}-3\,\sqrt{dx+c}c\right)Aab}{d}+\frac{231\,\left(3\,(dx+c)^{\frac{5}{2}}-10\,(dx+c)^{\frac{3}{2}}c+15\right.}{d^2}\right.}{}$$

```
[In] integrate((b*x+a)^2*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x, algorithm="giac")
```

```
[Out] 2/3465*(3465*sqrt(d*x + c)*A*a^2 + 1155*((d*x + c)^(3/2) - 3*sqrt(d*x + c)*
c)*B*a^2/d + 2310*((d*x + c)^(3/2) - 3*sqrt(d*x + c)*c)*A*a*b/d + 231*(3*(d
*x + c)^(5/2) - 10*(d*x + c)^(3/2)*c + 15*sqrt(d*x + c)*c^2)*C*a^2/d^2 + 46
2*(3*(d*x + c)^(5/2) - 10*(d*x + c)^(3/2)*c + 15*sqrt(d*x + c)*c^2)*B*a*b/d
^2 + 231*(3*(d*x + c)^(5/2) - 10*(d*x + c)^(3/2)*c + 15*sqrt(d*x + c)*c^2)*
A*b^2/d^2 + 99*(5*(d*x + c)^(7/2) - 21*(d*x + c)^(5/2)*c + 35*(d*x + c)^(3/
2)*c^2 - 35*sqrt(d*x + c)*c^3)*D*a^2/d^3 + 198*(5*(d*x + c)^(7/2) - 21*(d*x
 + c)^(5/2)*c + 35*(d*x + c)^(3/2)*c^2 - 35*sqrt(d*x + c)*c^3)*C*a*b/d^3 +
```

```
99*(5*(d*x + c)^(7/2) - 21*(d*x + c)^(5/2)*c + 35*(d*x + c)^(3/2)*c^2 - 35*
sqrt(d*x + c)*c^3)*B*b^2/d^3 + 22*(35*(d*x + c)^(9/2) - 180*(d*x + c)^(7/2)
*c + 378*(d*x + c)^(5/2)*c^2 - 420*(d*x + c)^(3/2)*c^3 + 315*sqrt(d*x + c)*
c^4)*D*a*b/d^4 + 11*(35*(d*x + c)^(9/2) - 180*(d*x + c)^(7/2)*c + 378*(d*x
+ c)^(5/2)*c^2 - 420*(d*x + c)^(3/2)*c^3 + 315*sqrt(d*x + c)*c^4)*C*b^2/d^4
 + 5*(63*(d*x + c)^(11/2) - 385*(d*x + c)^(9/2)*c + 990*(d*x + c)^(7/2)*c^2
 - 1386*(d*x + c)^(5/2)*c^3 + 1155*(d*x + c)^(3/2)*c^4 - 693*sqrt(d*x + c)*
c^5)*D*b^2/d^5)/d
```

Mupad [B] (verification not implemented)

Time = 4.09 (sec) , antiderivative size = 514, normalized size of antiderivative = 1.59

$$
\begin{aligned}
&\int \frac{(a+bx)^2 (A+Bx+Cx^2+Dx^3)}{\sqrt{c+dx}}\,dx \\
&= \frac{2A\sqrt{c+dx}\left(3b^2(c+dx)^2+15a^2d^2+15b^2c^2-10b^2c(c+dx)+10abd(c+dx)-30abcd\right)}{15d^3} \\
&\quad+ \frac{2Bb^2(c+dx)^{7/2}}{7d^4} + \frac{2Cb^2(c+dx)^{9/2}}{9d^5} + \frac{2B(c+dx)^{3/2}(a^2d^2-4abcd+3b^2c^2)}{3d^4} \\
&\quad+ \frac{2C(c+dx)^{5/2}(a^2d^2-6abcd+6b^2c^2)}{5d^5} - \frac{2Bc(ad-bc)^2\sqrt{c+dx}}{d^4} \\
&\quad- \frac{4Cc(c+dx)^{3/2}(a^2d^2-3abcd+2b^2c^2)}{3d^5} \\
&\quad- \frac{10b^2cD\left(\frac{2(c+dx)^{9/2}}{9d^5}+\frac{2c^4\sqrt{c+dx}}{d^5}-\frac{8c^3(c+dx)^{3/2}}{3d^5}+\frac{12c^2(c+dx)^{5/2}}{5d^5}-\frac{8c(c+dx)^{7/2}}{7d^5}\right)}{11d} \\
&\quad+ \frac{2Cc^2(ad-bc)^2\sqrt{c+dx}}{d^5} \\
&\quad- \frac{2a^2\sqrt{c+dx}\,D\left(6c(c+dx)^2-20c^2(c+dx)+30c^3-5d^3x^3\right)}{35d^4} \\
&\quad+ \frac{2b^2x^5\sqrt{c+dx}\,D}{11d} + \frac{2Bb(2ad-3bc)(c+dx)^{5/2}}{5d^4} + \frac{4Cb(ad-2bc)(c+dx)^{7/2}}{7d^5} \\
&\quad+ \frac{4ab\sqrt{c+dx}\,D\left(168c^2(c+dx)^2-280c^3(c+dx)-40c(c+dx)^3+280c^4+35d^4x^4\right)}{315d^5}
\end{aligned}
$$

```
[In] int(((a + b*x)^2*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(1/2),x)
```

```
[Out] (2*A*(c + d*x)^(1/2)*(3*b^2*(c + d*x)^2 + 15*a^2*d^2 + 15*b^2*c^2 - 10*b^2*
c*(c + d*x) + 10*a*b*d*(c + d*x) - 30*a*b*c*d))/(15*d^3) + (2*B*b^2*(c + d*
x)^(7/2))/(7*d^4) + (2*C*b^2*(c + d*x)^(9/2))/(9*d^5) + (2*B*(c + d*x)^(3/2
)*(a^2*d^2 + 3*b^2*c^2 - 4*a*b*c*d))/(3*d^4) + (2*C*(c + d*x)^(5/2)*(a^2*d^
2 + 6*b^2*c^2 - 6*a*b*c*d))/(5*d^5) - (2*B*c*(a*d - b*c)^2*(c + d*x)^(1/2))
/d^4 - (4*C*c*(c + d*x)^(3/2)*(a^2*d^2 + 2*b^2*c^2 - 3*a*b*c*d))/(3*d^5) -
(10*b^2*c*D*((2*(c + d*x)^(9/2))/(9*d^5) + (2*c^4*(c + d*x)^(1/2))/d^5 - (8
```

```
*c^3*(c + d*x)^(3/2))/(3*d^5) + (12*c^2*(c + d*x)^(5/2))/(5*d^5) - (8*c*(c
+ d*x)^(7/2))/(7*d^5)))/(11*d) + (2*C*c^2*(a*d - b*c)^2*(c + d*x)^(1/2))/d^
5 - (2*a^2*(c + d*x)^(1/2)*D*(6*c*(c + d*x)^2 - 20*c^2*(c + d*x) + 30*c^3 -
 5*d^3*x^3))/(35*d^4) + (2*b^2*x^5*(c + d*x)^(1/2)*D)/(11*d) + (2*B*b*(2*a*
d - 3*b*c)*(c + d*x)^(5/2))/(5*d^4) + (4*C*b*(a*d - 2*b*c)*(c + d*x)^(7/2))
/(7*d^5) + (4*a*b*(c + d*x)^(1/2)*D*(168*c^2*(c + d*x)^2 - 280*c^3*(c + d*x
) - 40*c*(c + d*x)^3 + 280*c^4 + 35*d^4*x^4))/(315*d^5)
```

3.3 $\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx$

Optimal result

Integrand size = 30, antiderivative size = 212

$$\begin{aligned}
&\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx \\
&= -\frac{2(bc-ad)\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)\sqrt{c+dx}}{d^5} \\
&\quad -\frac{2\left(ad\left(2cCd-Bd^2-3c^2D\right)-b\left(3c^2Cd-2Bcd^2+Ad^3-4c^3D\right)\right)(c+dx)^{3/2}}{3d^5} \\
&\quad +\frac{2\left(ad(Cd-3cD)-b\left(3cCd-Bd^2-6c^2D\right)\right)(c+dx)^{5/2}}{5d^5} \\
&\quad +\frac{2(bCd-4bcD+adD)(c+dx)^{7/2}}{7d^5}+\frac{2bD(c+dx)^{9/2}}{9d^5}
\end{aligned}$$

```
[Out] -2/3*(a*d*(-B*d^2+2*C*c*d-3*D*c^2)-b*(A*d^3-2*B*c*d^2+3*C*c^2*d-4*D*c^3))*(
d*x+c)^(3/2)/d^5+2/5*(a*d*(C*d-3*D*c)-b*(-B*d^2+3*C*c*d-6*D*c^2))*(d*x+c)^(
5/2)/d^5+2/7*(C*b*d+D*a*d-4*D*b*c)*(d*x+c)^(7/2)/d^5+2/9*b*D*(d*x+c)^(9/2)/
d^5-2*(-a*d+b*c)*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)*(d*x+c)^(1/2)/d^5
```

Rubi [A] (verified)

Time = 0.12 (sec) , antiderivative size = 212, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.033, Rules used

$= \{1634\}$

$$\begin{aligned}
&\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx\\
&= -\frac{2(c+dx)^{3/2}\left(ad(-Bd^2-3c^2D+2cCd)-b(Ad^3-2Bcd^2-4c^3D+3c^2Cd)\right)}{3d^5}\\
&\quad -\frac{2\sqrt{c+dx}(bc-ad)\left(Ad^3-Bcd^2+c^3(-D)+c^2Cd\right)}{d^5}\\
&\quad +\frac{2(c+dx)^{5/2}\left(ad(Cd-3cD)-b(-Bd^2-6c^2D+3cCd)\right)}{5d^5}\\
&\quad +\frac{2(c+dx)^{7/2}(adD-4bcD+bCd)}{7d^5}+\frac{2bD(c+dx)^{9/2}}{9d^5}
\end{aligned}$$

```
[In] Int[((a + b*x)*(A + B*x + C*x^2 + D*x^3))/Sqrt[c + d*x],x]
```

```
[Out] (-2*(b*c - a*d)*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D)*Sqrt[c + d*x])/d^5 - (2
*(a*d*(2*c*C*d - B*d^2 - 3*c^2*D) - b*(3*c^2*C*d - 2*B*c*d^2 + A*d^3 - 4*c^
3*D))*(c + d*x)^(3/2))/(3*d^5) + (2*(a*d*(C*d - 3*c*D) - b*(3*c*C*d - B*d^2
 - 6*c^2*D))*(c + d*x)^(5/2))/(5*d^5) + (2*(b*C*d - 4*b*c*D + a*d*D)*(c + d
*x)^(7/2))/(7*d^5) + (2*b*D*(c + d*x)^(9/2))/(9*d^5)
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \left(\frac{(-bc+ad)\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{d^4\sqrt{c+dx}}\right.\\
&\quad +\frac{\left(-ad(2cCd-Bd^2-3c^2D)+b(3c^2Cd-2Bcd^2+Ad^3-4c^3D)\right)\sqrt{c+dx}}{d^4}\\
&\quad +\frac{\left(ad(Cd-3cD)-b(3cCd-Bd^2-6c^2D)\right)(c+dx)^{3/2}}{d^4}\\
&\quad \left.+\frac{(bCd-4bcD+adD)(c+dx)^{5/2}}{d^4}+\frac{bD(c+dx)^{7/2}}{d^4}\right)dx\\
&= -\frac{2(bc-ad)\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)\sqrt{c+dx}}{d^5}\\
&\quad -\frac{2\left(ad(2cCd-Bd^2-3c^2D)-b(3c^2Cd-2Bcd^2+Ad^3-4c^3D)\right)(c+dx)^{3/2}}{3d^5}\\
&\quad +\frac{2\left(ad(Cd-3cD)-b(3cCd-Bd^2-6c^2D)\right)(c+dx)^{5/2}}{5d^5}\\
&\quad +\frac{2(bCd-4bcD+adD)(c+dx)^{7/2}}{7d^5}+\frac{2bD(c+dx)^{9/2}}{9d^5}
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.13 (sec) , antiderivative size = 184, normalized size of antiderivative = 0.87

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx$$
$$= \frac{2\sqrt{c+dx}(3ad(-48c^3D+8c^2d(7C+3Dx)-2cd^2(35B+x(14C+9Dx))+d^3(105A+x(35B+3x(7C+}{}$$

```
[In] Integrate[((a + b*x)*(A + B*x + C*x^2 + D*x^3))/Sqrt[c + d*x],x]
```

```
[Out] (2*Sqrt[c + d*x]*(3*a*d*(-48*c^3*D + 8*c^2*d*(7*C + 3*D*x) - 2*c*d^2*(35*B
+ x*(14*C + 9*D*x)) + d^3*(105*A + x*(35*B + 3*x*(7*C + 5*D*x)))) + b*(128*
c^4*D - 16*c^3*d*(9*C + 4*D*x) + 24*c^2*d^2*(7*B + x*(3*C + 2*D*x)) + d^4*x
*(105*A + x*(63*B + 5*x*(9*C + 7*D*x))) - 2*c*d^3*(105*A + x*(42*B + x*(27*
C + 20*D*x))))))/(315*d^5)
```

Maple [A] (verified)

Time = 1.65 (sec) , antiderivative size = 162, normalized size of antiderivative = 0.76

method	result
pseudoelliptic	$\frac{2\sqrt{dx+c}\left(\left(\left(\frac{Db\,x^4}{9}+\frac{(Cb+Da)x^3}{7}+\frac{(Bb+Ca)x^2}{5}+\frac{(Ab+Ba)x}{3}+Aa\right)d^4-\frac{2\left(\frac{4Db\,x^3}{21}+\frac{9(Cb+Da)x^2}{35}+\frac{2(Bb+Ca)x}{5}+Ab+Ba\right)c\,d^3}{3}+\right.\right.}{d^5}$
derivativedivides	$\frac{\frac{2Db(dx+c)^{\frac{9}{2}}}{9}+\frac{2((ad-bc)D+b(Cd-3Dc))(dx+c)^{\frac{7}{2}}}{7}+\frac{2\left((ad-bc)(Cd-3Dc)+b\left(B\,d^2-2Ccd+3Dc^2\right)\right)(dx+c)^{\frac{5}{2}}}{5}+\frac{2\left((ad-bc)\left(B\,d^2-2Cc\right.\right.}{}}{d^5}$
default	$\frac{\frac{2Db(dx+c)^{\frac{9}{2}}}{9}+\frac{2((ad-bc)D+b(Cd-3Dc))(dx+c)^{\frac{7}{2}}}{7}+\frac{2\left((ad-bc)(Cd-3Dc)+b\left(B\,d^2-2Ccd+3Dc^2\right)\right)(dx+c)^{\frac{5}{2}}}{5}+\frac{2\left((ad-bc)\left(B\,d^2-2Cc\right.\right.}{}}{d^5}$
gosper	$\frac{2\sqrt{dx+c}\,\left(35Db\,x^4d^4+45Cb\,d^4x^3+45Da\,d^4x^3-40Dbc\,d^3x^3+63Bb\,d^4x^2+63Ca\,d^4x^2-54Cbc\,d^3x^2-54Dac\,d^3x^2+48Db\,c^2\right.}{}$
trager	$\frac{2\sqrt{dx+c}\,\left(35Db\,x^4d^4+45Cb\,d^4x^3+45Da\,d^4x^3-40Dbc\,d^3x^3+63Bb\,d^4x^2+63Ca\,d^4x^2-54Cbc\,d^3x^2-54Dac\,d^3x^2+48Db\,c^2\right.}{}$

```
[In] int((b*x+a)*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x,method=_RETURNVERBOSE)
```

```
[Out] 2*(d*x+c)^(1/2)*((1/9*D*b*x^4+1/7*(C*b+D*a)*x^3+1/5*(B*b+C*a)*x^2+1/3*(A*b+
B*a)*x+A*a)*d^4-2/3*(4/21*D*b*x^3+9/35*(C*b+D*a)*x^2+2/5*(B*b+C*a)*x+A*b+B*
a)*c*d^3+8/15*c^2*(2/7*D*b*x^2+3/7*(C*b+D*a)*x+B*b+C*a)*d^2-16/35*(4/9*D*b*
x+C*b+D*a)*c^3*d+128/315*D*b*c^4)/d^5
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 204, normalized size of antiderivative = 0.96

$$\int \frac{(a+bx)\,(A+Bx+Cx^2+Dx^3)}{\sqrt{c+dx}}\,dx$$
$$= \frac{2\,(35\,Dbd^4x^4 + 128\,Dbc^4 + 315\,Aad^4 + 168\,(Ca+Bb)c^2d^2 - 210\,(Ba+Ab)cd^3 - 5\,(8\,Dbcd^3 - 9\,(Da + C}{}$$

```
[In] integrate((b*x+a)*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x, algorithm="fricas")
```

```
[Out] 2/315*(35*D*b*d^4*x^4 + 128*D*b*c^4 + 315*A*a*d^4 + 168*(C*a + B*b)*c^2*d^2
 - 210*(B*a + A*b)*c*d^3 - 5*(8*D*b*c*d^3 - 9*(D*a + C*b)*d^4)*x^3 + 3*(16*
D*b*c^2*d^2 + 21*(C*a + B*b)*d^4 - 18*(D*a*c + C*b*c)*d^3)*x^2 - 144*(D*a*c
^3 + C*b*c^3)*d - (64*D*b*c^3*d + 84*(C*a + B*b)*c*d^3 - 105*(B*a + A*b)*d^
4 - 72*(D*a*c^2 + C*b*c^2)*d^2)*x)*sqrt(d*x + c)/d^5
```

Sympy [A] (verification not implemented)

Time = 1.25 (sec) , antiderivative size = 320, normalized size of antiderivative = 1.51

$$\int \frac{(a+bx)\,(A+Bx+Cx^2+Dx^3)}{\sqrt{c+dx}}\,dx$$
$$= \begin{cases} \frac{2\,(\frac{Db(c+dx)^{\frac{9}{2}}}{9d^4} + \frac{(c+dx)^{\frac{7}{2}}(Cbd+Dad-4Dbc)}{7d^4} + \frac{(c+dx)^{\frac{5}{2}}(Bbd^2+Cad^2-3Cbcd-3Dacd+6Dbc^2)}{5d^4} + \frac{(c+dx)^{\frac{3}{2}}(Abd^3+Bad^3-2Bbcd^2-2Cacd^2+3Cbc^2d+3Dac}{3d^4}}{d} \\ \frac{Aax+\frac{Dbx^5}{5}+\frac{x^4(Cb+Da)}{4}+\frac{x^3(Bb+Ca)}{3}+\frac{x^2(Ab+Ba)}{2}}{\sqrt{c}} \end{cases}$$

```
[In] integrate((b*x+a)*(D*x**3+C*x**2+B*x+A)/(d*x+c)**(1/2),x)
```

```
[Out] Piecewise((2*(D*b*(c + d*x)**(9/2)/(9*d**4) + (c + d*x)**(7/2)*(C*b*d + D*a
*d - 4*D*b*c)/(7*d**4) + (c + d*x)**(5/2)*(B*b*d**2 + C*a*d**2 - 3*C*b*c*d
- 3*D*a*c*d + 6*D*b*c**2)/(5*d**4) + (c + d*x)**(3/2)*(A*b*d**3 + B*a*d**3
- 2*B*b*c*d**2 - 2*C*a*c*d**2 + 3*C*b*c**2*d + 3*D*a*c**2*d - 4*D*b*c**3)/(
3*d**4) + sqrt(c + d*x)*(A*a*d**4 - A*b*c*d**3 - B*a*c*d**3 + B*b*c**2*d**2
 + C*a*c**2*d**2 - C*b*c**3*d - D*a*c**3*d + D*b*c**4)/d**4)/d, Ne(d, 0)),
((A*a*x + D*b*x**5/5 + x**4*(C*b + D*a)/4 + x**3*(B*b + C*a)/3 + x**2*(A*b
+ B*a)/2)/sqrt(c), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.21 (sec) , antiderivative size = 198, normalized size of antiderivative = 0.93

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx$$
$$= 2\left(35\,(dx+c)^{\frac{9}{2}}Db-45\,(4\,Dbc-(Da+Cb)d)(dx+c)^{\frac{7}{2}}+63\,(6\,Dbc^2-3\,(Da+Cb)cd+(Ca+Bb)d^2)(d$$

```
[In] integrate((b*x+a)*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x, algorithm="maxima")
```

```
[Out] 2/315*(35*(d*x + c)^(9/2)*D*b - 45*(4*D*b*c - (D*a + C*b)*d)*(d*x + c)^(7/2
) + 63*(6*D*b*c^2 - 3*(D*a + C*b)*c*d + (C*a + B*b)*d^2)*(d*x + c)^(5/2) -
105*(4*D*b*c^3 - 3*(D*a + C*b)*c^2*d + 2*(C*a + B*b)*c*d^2 - (B*a + A*b)*d^
3)*(d*x + c)^(3/2) + 315*(D*b*c^4 + A*a*d^4 - (D*a + C*b)*c^3*d + (C*a + B*
b)*c^2*d^2 - (B*a + A*b)*c*d^3)*sqrt(d*x + c))/d^5
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 309, normalized size of antiderivative = 1.46

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx$$
$$= 2\left(315\,\sqrt{dx+c}Aa+\frac{105\left((dx+c)^{\frac{3}{2}}-3\,\sqrt{dx+c}c\right)Ba}{d}+\frac{105\left((dx+c)^{\frac{3}{2}}-3\,\sqrt{dx+c}c\right)Ab}{d}+\frac{21\left(3\,(dx+c)^{\frac{5}{2}}-10\,(dx+c)^{\frac{3}{2}}c+15\,\sqrt{dx+c}c^2\right)C}{d^2}$$

```
[In] integrate((b*x+a)*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x, algorithm="giac")
```

```
[Out] 2/315*(315*sqrt(d*x + c)*A*a + 105*((d*x + c)^(3/2) - 3*sqrt(d*x + c)*c)*B*
a/d + 105*((d*x + c)^(3/2) - 3*sqrt(d*x + c)*c)*A*b/d + 21*(3*(d*x + c)^(5/
2) - 10*(d*x + c)^(3/2)*c + 15*sqrt(d*x + c)*c^2)*C*a/d^2 + 21*(3*(d*x + c)
^(5/2) - 10*(d*x + c)^(3/2)*c + 15*sqrt(d*x + c)*c^2)*B*b/d^2 + 9*(5*(d*x +
 c)^(7/2) - 21*(d*x + c)^(5/2)*c + 35*(d*x + c)^(3/2)*c^2 - 35*sqrt(d*x + c
)*c^3)*D*a/d^3 + 9*(5*(d*x + c)^(7/2) - 21*(d*x + c)^(5/2)*c + 35*(d*x + c)
^(3/2)*c^2 - 35*sqrt(d*x + c)*c^3)*C*b/d^3 + (35*(d*x + c)^(9/2) - 180*(d*x
 + c)^(7/2)*c + 378*(d*x + c)^(5/2)*c^2 - 420*(d*x + c)^(3/2)*c^3 + 315*sqr
t(d*x + c)*c^4)*D*b/d^4)/d
```

Mupad [B] (verification not implemented)

Time = 3.71 (sec) , antiderivative size = 351, normalized size of antiderivative = 1.66

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{\sqrt{c+dx}}\,dx$$
$$=\frac{2\,A\,b\,{\left(c+d\,x\right)}^{3/2}-6\,A\,b\,c\,\sqrt{c+d\,x}}{3\,d^2}+\frac{2\,B\,a\,{\left(c+d\,x\right)}^{3/2}-6\,B\,a\,c\,\sqrt{c+d\,x}}{3\,d^2}$$
$$+\frac{6\,B\,b\,{\left(c+d\,x\right)}^{5/2}+30\,B\,b\,c^2\,\sqrt{c+d\,x}-20\,B\,b\,c\,{\left(c+d\,x\right)}^{3/2}}{15\,d^3}$$
$$+\frac{6\,C\,a\,{\left(c+d\,x\right)}^{5/2}+30\,C\,a\,c^2\,\sqrt{c+d\,x}-20\,C\,a\,c\,{\left(c+d\,x\right)}^{3/2}}{15\,d^3}+\frac{2\,A\,a\,\sqrt{c+d\,x}}{d}$$
$$+\frac{2\,C\,b\,{\left(c+d\,x\right)}^{7/2}}{7\,d^4}-\frac{2\,a\,\sqrt{c+d\,x}\,D\,\left(6\,c\,{\left(c+d\,x\right)}^2-20\,c^2\,\left(c+d\,x\right)+30\,c^3-5\,d^3\,x^3\right)}{35\,d^4}$$
$$+\frac{2\,b\,x^4\,\sqrt{c+d\,x}\,D}{9\,d}-\frac{6\,C\,b\,c\,{\left(c+d\,x\right)}^{5/2}}{5\,d^4}$$
$$-\frac{8\,b\,c\,D\,\left(\frac{2\,{\left(c+d\,x\right)}^{7/2}}{7\,d^4}-\frac{2\,c^3\,\sqrt{c+d\,x}}{d^4}+\frac{2\,c^2\,{\left(c+d\,x\right)}^{3/2}}{d^4}-\frac{6\,c\,{\left(c+d\,x\right)}^{5/2}}{5\,d^4}\right)}{9\,d}$$
$$-\frac{2\,C\,b\,c^3\,\sqrt{c+d\,x}}{d^4}+\frac{2\,C\,b\,c^2\,{\left(c+d\,x\right)}^{3/2}}{d^4}$$

```
[In] int(((a + b*x)*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(1/2),x)
```

```
[Out] (2*A*b*(c + d*x)^(3/2) - 6*A*b*c*(c + d*x)^(1/2))/(3*d^2) + (2*B*a*(c + d*x
)^(3/2) - 6*B*a*c*(c + d*x)^(1/2))/(3*d^2) + (6*B*b*(c + d*x)^(5/2) + 30*B*
b*c^2*(c + d*x)^(1/2) - 20*B*b*c*(c + d*x)^(3/2))/(15*d^3) + (6*C*a*(c + d*
x)^(5/2) + 30*C*a*c^2*(c + d*x)^(1/2) - 20*C*a*c*(c + d*x)^(3/2))/(15*d^3)
+ (2*A*a*(c + d*x)^(1/2))/d + (2*C*b*(c + d*x)^(7/2))/(7*d^4) - (2*a*(c + d
*x)^(1/2)*D*(6*c*(c + d*x)^2 - 20*c^2*(c + d*x) + 30*c^3 - 5*d^3*x^3))/(35*
d^4) + (2*b*x^4*(c + d*x)^(1/2)*D)/(9*d) - (6*C*b*c*(c + d*x)^(5/2))/(5*d^4
) - (8*b*c*D*((2*(c + d*x)^(7/2))/(7*d^4) - (2*c^3*(c + d*x)^(1/2))/d^4 + (
2*c^2*(c + d*x)^(3/2))/d^4 - (6*c*(c + d*x)^(5/2))/(5*d^4)))/(9*d) - (2*C*b
*c^3*(c + d*x)^(1/2))/d^4 + (2*C*b*c^2*(c + d*x)^(3/2))/d^4
```

3.4 $\int \frac{A+Bx+Cx^2+Dx^3}{\sqrt{c+dx}}\,dx$

Optimal result

Integrand size = 25, antiderivative size = 115

$$\int \frac{A+Bx+Cx^2+Dx^3}{\sqrt{c+dx}}\,dx = \frac{2(c^2Cd-Bcd^2+Ad^3-c^3D)\sqrt{c+dx}}{d^4} - \frac{2(2cCd-Bd^2-3c^2D)(c+dx)^{3/2}}{3d^4} + \frac{2(Cd-3cD)(c+dx)^{5/2}}{5d^4} + \frac{2D(c+dx)^{7/2}}{7d^4}$$

```
[Out] -2/3*(-B*d^2+2*C*c*d-3*D*c^2)*(d*x+c)^(3/2)/d^4+2/5*(C*d-3*D*c)*(d*x+c)^(5/
2)/d^4+2/7*D*(d*x+c)^(7/2)/d^4+2*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)*(d*x+c)^(1/2
)/d^4
```

Rubi [A] (verified)

Time = 0.05 (sec) , antiderivative size = 115, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.040, Rules used = {1864}

$$\int \frac{A+Bx+Cx^2+Dx^3}{\sqrt{c+dx}}\,dx = \frac{2\sqrt{c+dx}(Ad^3-Bcd^2+c^3(-D)+c^2Cd)}{d^4} - \frac{2(c+dx)^{3/2}(-Bd^2-3c^2D+2cCd)}{3d^4} + \frac{2(c+dx)^{5/2}(Cd-3cD)}{5d^4} + \frac{2D(c+dx)^{7/2}}{7d^4}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/Sqrt[c + d*x],x]
```

```
[Out] (2*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D)*Sqrt[c + d*x])/d^4 - (2*(2*c*C*d - B
*d^2 - 3*c^2*D)*(c + d*x)^(3/2))/(3*d^4) + (2*(C*d - 3*c*D)*(c + d*x)^(5/2)
)/(5*d^4) + (2*D*(c + d*x)^(7/2))/(7*d^4)
```

Rule 1864

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_.))^(p_.), x_Symbol] :> Int[ExpandIntegrand
[Pq*(a + b*x^n)^p, x], x] /; FreeQ[{a, b, n}, x] && PolyQ[Pq, x] && (IGtQ[p
, 0] || EqQ[n, 1])
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \left(\frac{c^2Cd - Bcd^2 + Ad^3 - c^3D}{d^3\sqrt{c+dx}} + \frac{(-2cCd + Bd^2 + 3c^2D)\sqrt{c+dx}}{d^3} + \frac{(Cd-3cD)(c+dx)^{3/2}}{d^3} + \frac{D(c+dx)^{5/2}}{d^3}\right) dx \\ &= \frac{2(c^2Cd - Bcd^2 + Ad^3 - c^3D)\sqrt{c+dx}}{d^4} - \frac{2(2cCd - Bd^2 - 3c^2D)(c+dx)^{3/2}}{3d^4} + \frac{2(Cd-3cD)(c+dx)^{5/2}}{5d^4} + \frac{2D(c+dx)^{7/2}}{7d^4}\end{aligned}$$

Mathematica [A] (verified)

Time = 0.06 (sec) , antiderivative size = 82, normalized size of antiderivative = 0.71

$$\int \frac{A + Bx + Cx^2 + Dx^3}{\sqrt{c+dx}}\,dx = \frac{2\sqrt{c+dx}(-48c^3D + 8c^2d(7C+3Dx) - 2cd^2(35B + x(14C+9Dx)) + d^3(105A + x(35B + 3x(7C + 5}{105d^4}$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/Sqrt[c + d*x],x]
```

```
[Out] (2*Sqrt[c + d*x]*(-48*c^3*D + 8*c^2*d*(7*C + 3*D*x) - 2*c*d^2*(35*B + x*(14
*C + 9*D*x)) + d^3*(105*A + x*(35*B + 3*x*(7*C + 5*D*x)))))/(105*d^4)
```

Maple [A] (verified)

Time = 1.63 (sec) , antiderivative size = 72, normalized size of antiderivative = 0.63

method	result
pseudoelliptic	$\frac{2\left(\left(A+\frac{1}{7}Dx^{3}+\frac{1}{5}C\,x^{2}+\frac{1}{3}Bx\right)d^{3}-\frac{2c\left(\frac{9}{35}Dx^{2}+\frac{2}{5}Cx+B\right)d^{2}}{3}+\frac{8c^{2}\left(\frac{3Dx}{7}+C\right)d}{15}-\frac{16Dc^{3}}{35}\right)\sqrt{dx+c}}{d^{4}}$
gosper	$\frac{2\sqrt{dx+c}\,\left(15Dx^{3}d^{3}+21C\,d^{3}x^{2}-18Dc\,d^{2}x^{2}+35B\,d^{3}x-28Cc\,d^{2}x+24Dc^{2}dx+105A\,d^{3}-70Bc\,d^{2}+56C\,c^{2}d-48Dc^{3}\right)}{105d^{4}}$
trager	$\frac{2\sqrt{dx+c}\,\left(15Dx^{3}d^{3}+21C\,d^{3}x^{2}-18Dc\,d^{2}x^{2}+35B\,d^{3}x-28Cc\,d^{2}x+24Dc^{2}dx+105A\,d^{3}-70Bc\,d^{2}+56C\,c^{2}d-48Dc^{3}\right)}{105d^{4}}$
derivativedivides	$\frac{\frac{2D(dx+c)^{\frac{7}{2}}}{7}+\frac{2Cd(dx+c)^{\frac{5}{2}}}{5}-\frac{6Dc(dx+c)^{\frac{5}{2}}}{5}+\frac{2B\,d^{2}(dx+c)^{\frac{3}{2}}}{3}-\frac{4Ccd(dx+c)^{\frac{3}{2}}}{3}+2Dc^{2}(dx+c)^{\frac{3}{2}}+2A\,d^{3}\sqrt{dx+c}-2Bc\,d^{2}\sqrt{dx+c}+2}{d^{4}}$
default	$\frac{\frac{2D(dx+c)^{\frac{7}{2}}}{7}+\frac{2Cd(dx+c)^{\frac{5}{2}}}{5}-\frac{6Dc(dx+c)^{\frac{5}{2}}}{5}+\frac{2B\,d^{2}(dx+c)^{\frac{3}{2}}}{3}-\frac{4Ccd(dx+c)^{\frac{3}{2}}}{3}+2Dc^{2}(dx+c)^{\frac{3}{2}}+2A\,d^{3}\sqrt{dx+c}-2Bc\,d^{2}\sqrt{dx+c}+2}{d^{4}}$

```
[In] int((D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x,method=_RETURNVERBOSE)
```

```
[Out] 2*((A+1/7*D*x^3+1/5*C*x^2+1/3*B*x)*d^3-2/3*c*(9/35*D*x^2+2/5*C*x+B)*d^2+8/1
5*c^2*(3/7*D*x+C)*d-16/35*D*c^3)*(d*x+c)^(1/2)/d^4
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 90, normalized size of antiderivative = 0.78

$$\int \frac{A+Bx+Cx^2+Dx^3}{\sqrt{c+dx}}\,dx = \frac{2\,(15\,Dd^3x^3-48\,Dc^3+56\,Cc^2d-70\,Bcd^2+105\,Ad^3-3\,(6\,Dcd^2-7\,Cd^3)x^2+(24\,Dc^2d-28\,Ccd^2+35}{105\,d^4}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x, algorithm="fricas")
```

```
[Out] 2/105*(15*D*d^3*x^3 - 48*D*c^3 + 56*C*c^2*d - 70*B*c*d^2 + 105*A*d^3 - 3*(6
*D*c*d^2 - 7*C*d^3)*x^2 + (24*D*c^2*d - 28*C*c*d^2 + 35*B*d^3)*x)*sqrt(d*x
+ c)/d^4
```

Sympy [A] (verification not implemented)

Time = 0.62 (sec) , antiderivative size = 163, normalized size of antiderivative = 1.42

$$\int \frac{A+Bx+Cx^2+Dx^3}{\sqrt{c+dx}}\,dx = \begin{cases} \frac{2A\sqrt{c+dx}+\frac{2B\left(-c\sqrt{c+dx}+\frac{(c+dx)^{\frac{3}{2}}}{3}\right)}{d}+\frac{2C\left(c^{2}\sqrt{c+dx}-\frac{2c(c+dx)^{\frac{3}{2}}}{3}+\frac{(c+dx)^{\frac{5}{2}}}{5}\right)}{d^{2}}+\frac{2D\left(-c^{3}\sqrt{c+dx}+c^{2}(c+dx)^{\frac{3}{2}}-\frac{3c(c+dx)^{\frac{5}{2}}}{5}+\frac{(c+dx)^{\frac{7}{2}}}{7}\right)}{d^{3}}}{d} & \text{for } d \neq \\ \frac{Ax+\frac{Bx^{2}}{2}+\frac{Cx^{3}}{3}+\frac{Dx^{4}}{4}}{\sqrt{c}} & \text{otherw} \end{cases}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(d*x+c)**(1/2),x)
```

```
[Out] Piecewise(((2*A*sqrt(c + d*x) + 2*B*(-c*sqrt(c + d*x) + (c + d*x)**(3/2)/3)
/d + 2*C*(c**2*sqrt(c + d*x) - 2*c*(c + d*x)**(3/2)/3 + (c + d*x)**(5/2)/5)
/d**2 + 2*D*(-c**3*sqrt(c + d*x) + c**2*(c + d*x)**(3/2) - 3*c*(c + d*x)**(
5/2)/5 + (c + d*x)**(7/2)/7)/d**3)/d, Ne(d, 0)), ((A*x + B*x**2/2 + C*x**3/
3 + D*x**4/4)/sqrt(c), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 128, normalized size of antiderivative = 1.11

$$\int \frac{A+Bx+Cx^2+Dx^3}{\sqrt{c+dx}}\,dx = \frac{2\Big(105\sqrt{dx+c}A + \frac{35\left((dx+c)^{\frac{3}{2}}-3\sqrt{dx+c}c\right)B}{d} + \frac{7\left(3\,(dx+c)^{\frac{5}{2}}-10\,(dx+c)^{\frac{3}{2}}c+15\sqrt{dx+c}c^2\right)C}{d^2} + \frac{3\,(5\,(dx+c)^{\frac{7}{2}}-21\,(dx+c)^{\frac{5}{2}}c+35}{d^3}}{105\,d}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x, algorithm="maxima")
```

```
[Out] 2/105*(105*sqrt(d*x + c)*A + 35*((d*x + c)^(3/2) - 3*sqrt(d*x + c)*c)*B/d +
 7*(3*(d*x + c)^(5/2) - 10*(d*x + c)^(3/2)*c + 15*sqrt(d*x + c)*c^2)*C/d^2
+ 3*(5*(d*x + c)^(7/2) - 21*(d*x + c)^(5/2)*c + 35*(d*x + c)^(3/2)*c^2 - 35
*sqrt(d*x + c)*c^3)*D/d^3)/d
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 128, normalized size of antiderivative = 1.11

$$\int \frac{A+Bx+Cx^2+Dx^3}{\sqrt{c+dx}}\,dx = \frac{2\Big(105\sqrt{dx+c}A + \frac{35\left((dx+c)^{\frac{3}{2}}-3\sqrt{dx+c}c\right)B}{d} + \frac{7\left(3\,(dx+c)^{\frac{5}{2}}-10\,(dx+c)^{\frac{3}{2}}c+15\sqrt{dx+c}c^2\right)C}{d^2} + \frac{3\,(5\,(dx+c)^{\frac{7}{2}}-21\,(dx+c)^{\frac{5}{2}}c+35}{d^3}}{105\,d}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(d*x+c)^(1/2),x, algorithm="giac")
```

```
[Out] 2/105*(105*sqrt(d*x + c)*A + 35*((d*x + c)^(3/2) - 3*sqrt(d*x + c)*c)*B/d +
 7*(3*(d*x + c)^(5/2) - 10*(d*x + c)^(3/2)*c + 15*sqrt(d*x + c)*c^2)*C/d^2
+ 3*(5*(d*x + c)^(7/2) - 21*(d*x + c)^(5/2)*c + 35*(d*x + c)^(3/2)*c^2 - 35
*sqrt(d*x + c)*c^3)*D/d^3)/d
```

Mupad [B] (verification not implemented)

Time = 3.24 (sec) , antiderivative size = 128, normalized size of antiderivative = 1.11

$$
\begin{aligned}
&\int \frac{A+Bx+Cx^2+Dx^3}{\sqrt{c+dx}}\,dx \\
&= \frac{6\,C\,{\left(c+d\,x\right)}^{5/2}-20\,C\,c\,{\left(c+d\,x\right)}^{3/2}+30\,C\,c^2\,\sqrt{c+d\,x}}{15\,d^3} \\
&\quad+\frac{2\,B\,{\left(c+d\,x\right)}^{3/2}-6\,B\,c\,\sqrt{c+d\,x}}{3\,d^2}+\frac{2\,A\,\sqrt{c+d\,x}}{d} \\
&\quad-\frac{2\,\sqrt{c+d\,x}\,D\,\left(6\,c\,{\left(c+d\,x\right)}^2-20\,c^2\,\left(c+d\,x\right)+30\,c^3-5\,d^3\,x^3\right)}{35\,d^4}
\end{aligned}
$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/(c + d*x)^(1/2),x)
```

```
[Out] (6*C*(c + d*x)^(5/2) - 20*C*c*(c + d*x)^(3/2) + 30*C*c^2*(c + d*x)^(1/2))/(
15*d^3) + (2*B*(c + d*x)^(3/2) - 6*B*c*(c + d*x)^(1/2))/(3*d^2) + (2*A*(c +
 d*x)^(1/2))/d - (2*(c + d*x)^(1/2)*D*(6*c*(c + d*x)^2 - 20*c^2*(c + d*x) +
 30*c^3 - 5*d^3*x^3))/(35*d^4)
```

3.5 $\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)\sqrt{c+dx}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 188

$$\begin{aligned}\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)\sqrt{c+dx}}\,dx &= \frac{2(a^2d^2D-abd(Cd-cD)-b^2(cCd-Bd^2-c^2D))\sqrt{c+dx}}{b^3d^3} \\ &\quad+\frac{2(bCd-2bcD-adD)(c+dx)^{3/2}}{3b^2d^3}+\frac{2D(c+dx)^{5/2}}{5bd^3} \\ &\quad-\frac{2(Ab^3-a(b^2B-abC+a^2D))\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{7/2}\sqrt{bc-ad}}\end{aligned}$$

```
[Out] 2/3*(C*b*d-D*a*d-2*D*b*c)*(d*x+c)^(3/2)/b^2/d^3+2/5*D*(d*x+c)^(5/2)/b/d^3-2
*(A*b^3-a*(B*b^2-C*a*b+D*a^2))*arctanh(b^(1/2)*(d*x+c)^(1/2)/(-a*d+b*c)^(1/
2))/b^(7/2)/(-a*d+b*c)^(1/2)+2*(a^2*d^2*D-a*b*d*(C*d-D*c)-b^2*(-B*d^2+C*c*d
-D*c^2))*(d*x+c)^(1/2)/b^3/d^3
```

Rubi [A] (verified)

Time = 0.14 (sec) , antiderivative size = 188, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.094, Rules used = {1634, 65, 214}

$$\begin{aligned}&\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)\sqrt{c+dx}}\,dx \\ &= -\frac{2(Ab^3-a(a^2D-abC+b^2B))\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{7/2}\sqrt{bc-ad}} \\ &\quad+\frac{2\sqrt{c+dx}(a^2d^2D-abd(Cd-cD)-(b^2(-Bd^2+c^2(-D)+cCd)))}{b^3d^3} \\ &\quad+\frac{2(c+dx)^{3/2}(-adD-2bcD+bCd)}{3b^2d^3}+\frac{2D(c+dx)^{5/2}}{5bd^3}\end{aligned}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/((a + b*x)*Sqrt[c + d*x]),x]
```

```
[Out] (2*(a^2*d^2*D - a*b*d*(C*d - c*D) - b^2*(c*C*d - B*d^2 - c^2*D))*Sqrt[c + d
*x])/(b^3*d^3) + (2*(b*C*d - 2*b*c*D - a*d*D)*(c + d*x)^(3/2))/(3*b^2*d^3)
+ (2*D*(c + d*x)^(5/2))/(5*b*d^3) - (2*(A*b^3 - a*(b^2*B - a*b*C + a^2*D))*
ArcTanh[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[b*c - a*d]])/(b^(7/2)*Sqrt[b*c - a*d])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \left(\frac{a^2d^2D-abd(Cd-cD)-b^2(cCd-Bd^2-c^2D)}{b^3d^2\sqrt{c+dx}} + \frac{Ab^3-a(b^2B-abC+a^2D)}{b^3(a+bx)\sqrt{c+dx}} + \frac{(bCd-2bcD-adD)\sqrt{c+dx}}{b^2d^2} + \frac{D(c+dx)^{3/2}}{bd^2}\right)dx \\
&= \frac{2(a^2d^2D-abd(Cd-cD)-b^2(cCd-Bd^2-c^2D))\sqrt{c+dx}}{b^3d^3} + \frac{2(bCd-2bcD-adD)(c+dx)^{3/2}}{3b^2d^3} + \frac{2D(c+dx)^{5/2}}{5bd^3} + \left(A-\frac{a(b^2B-abC+a^2D)}{b^3}\right)\int \frac{1}{(a+bx)\sqrt{c+dx}}\,dx
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{2(a^2d^2D - abd(Cd - cD) - b^2(cCd - Bd^2 - c^2D))\sqrt{c+dx}}{b^3d^3} \\
&\quad + \frac{2(bCd - 2bcD - adD)(c+dx)^{3/2}}{3b^2d^3} + \frac{2D(c+dx)^{5/2}}{5bd^3} \\
&\quad + \frac{\left(2\left(A - \frac{a(b^2B - abC + a^2D)}{b^3}\right)\right)\text{Subst}\left(\int \frac{1}{a - \frac{bc}{d} + \frac{bx^2}{d}}\,dx, x, \sqrt{c+dx}\right)}{d} \\
&= \frac{2(a^2d^2D - abd(Cd - cD) - b^2(cCd - Bd^2 - c^2D))\sqrt{c+dx}}{b^3d^3} \\
&\quad + \frac{2(bCd - 2bcD - adD)(c+dx)^{3/2}}{3b^2d^3} + \frac{2D(c+dx)^{5/2}}{5bd^3} \\
&\quad - \frac{2(Ab^3 - a(b^2B - abC + a^2D))\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{7/2}\sqrt{bc-ad}}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.25 (sec) , antiderivative size = 161, normalized size of antiderivative = 0.86

$$
\begin{aligned}
&\int \frac{A + Bx + Cx^2 + Dx^3}{(a+bx)\sqrt{c+dx}}\,dx \\
&= \frac{2\sqrt{c+dx}(15a^2d^2D - 5abd(3Cd - 2cD + dDx) + b^2(8c^2D - 2cd(5C + 2Dx) + d^2(15B + 5Cx + 3Dx^2))}{15b^3d^3} \\
&\quad + \frac{2(Ab^3 - a(b^2B - abC + a^2D))\arctan\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{-bc+ad}}\right)}{b^{7/2}\sqrt{-bc+ad}}
\end{aligned}
$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/((a + b*x)*Sqrt[c + d*x]),x]
```

```
[Out] (2*Sqrt[c + d*x]*(15*a^2*d^2*D - 5*a*b*d*(3*C*d - 2*c*D + d*D*x) + b^2*(8*c
^2*D - 2*c*d*(5*C + 2*D*x) + d^2*(15*B + 5*C*x + 3*D*x^2))))/(15*b^3*d^3) +
 (2*(A*b^3 - a*(b^2*B - a*b*C + a^2*D))*ArcTan[(Sqrt[b]*Sqrt[c + d*x])/Sqrt
[-(b*c) + a*d]])/(b^(7/2)*Sqrt[-(b*c) + a*d])
```

Maple [A] (verified)

Time = 1.72 (sec) , antiderivative size = 157, normalized size of antiderivative = 0.84

method	result
pseudoelliptic	$\frac{2d^3\left(b^3A-a\,b^2B+C\,a^2b-Da^3\right)\arctan\left(\frac{b\sqrt{dx+c}}{\sqrt{(ad-bc)b}}\right)+2\sqrt{(ad-bc)b}\,\sqrt{dx+c}\left(\left(\left(\frac{1}{5}Dx^2+\frac{1}{3}Cx+B\right)b^2-a\left(\frac{Dx}{3}+C\right)b+Da^2\right)}{d^3b^3\sqrt{(ad-bc)b}}$
derivativedivides	$\frac{\frac{2\left(\frac{D(dx+c)^{\frac{5}{2}}b^2}{5}+\frac{C\,b^2d(dx+c)^{\frac{3}{2}}}{3}-\frac{Dabd(dx+c)^{\frac{3}{2}}}{3}-\frac{2Db^2c(dx+c)^{\frac{3}{2}}}{3}+B\,b^2d^2\sqrt{dx+c}-Cab\,d^2\sqrt{dx+c}-C\,b^2cd\sqrt{dx+c}+Da^2d^2\sqrt{dx+c}+\right.}{b^3}}{d^3}$
default	$\frac{\frac{2\left(\frac{D(dx+c)^{\frac{5}{2}}b^2}{5}+\frac{C\,b^2d(dx+c)^{\frac{3}{2}}}{3}-\frac{Dabd(dx+c)^{\frac{3}{2}}}{3}-\frac{2Db^2c(dx+c)^{\frac{3}{2}}}{3}+B\,b^2d^2\sqrt{dx+c}-Cab\,d^2\sqrt{dx+c}-C\,b^2cd\sqrt{dx+c}+Da^2d^2\sqrt{dx+c}+\right.}{b^3}}{d^3}$

```
[In] int((D*x^3+C*x^2+B*x+A)/(b*x+a)/(d*x+c)^(1/2),x,method=_RETURNVERBOSE)
```

```
[Out] 2/((a*d-b*c)*b)^(1/2)*(d^3*(A*b^3-B*a*b^2+C*a^2*b-D*a^3)*arctan(b*(d*x+c)^(
1/2)/((a*d-b*c)*b)^(1/2))+((a*d-b*c)*b)^(1/2)*(d*x+c)^(1/2)*(((1/5*D*x^2+1/
3*C*x+B)*b^2-a*(1/3*D*x+C)*b+D*a^2)*d^2-2/3*((2/5*D*x+C)*b-D*a)*b*c*d+8/15*
D*b^2*c^2))/d^3/b^3
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 565, normalized size of antiderivative = 3.01

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)\sqrt{c+dx}}\,dx$$

$$= \left[\frac{15\left(Da^3-Ca^2b+Bab^2-Ab^3\right)\sqrt{b^2c-abd}\,d^3\log\left(\frac{bdx+2\,bc-ad+2\,\sqrt{b^2c-abd}\sqrt{dx+c}}{bx+a}\right)+2\left(8\,Db^4c^3-15\left(Da^3b-\right.\right.}{}\right.$$

$$-\frac{2\left(15\left(Da^3-Ca^2b+Bab^2-Ab^3\right)\sqrt{-b^2c+abd}\,d^3\arctan\left(\frac{\sqrt{-b^2c+abd}\sqrt{dx+c}}{bdx+bc}\right)-\left(8\,Db^4c^3-15\left(Da^3b-C\right.\right.\right.}{}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)/(d*x+c)^(1/2),x, algorithm="fricas")
```

```
[Out] [1/15*(15*(D*a^3 - C*a^2*b + B*a*b^2 - A*b^3)*sqrt(b^2*c - a*b*d)*d^3*log((
b*d*x + 2*b*c - a*d + 2*sqrt(b^2*c - a*b*d)*sqrt(d*x + c))/(b*x + a)) + 2*(
8*D*b^4*c^3 - 15*(D*a^3*b - C*a^2*b^2 + B*a*b^3)*d^3 + 5*(D*a^2*b^2*c - (C*
a*b^3 - 3*B*b^4)*c)*d^2 + 3*(D*b^4*c*d^2 - D*a*b^3*d^3)*x^2 + 2*(D*a*b^3*c^
2 - 5*C*b^4*c^2)*d - (4*D*b^4*c^2*d - 5*(D*a^2*b^2 - C*a*b^3)*d^3 + (D*a*b^
3*c - 5*C*b^4*c)*d^2)*x)*sqrt(d*x + c))/(b^5*c*d^3 - a*b^4*d^4), -2/15*(15*
(D*a^3 - C*a^2*b + B*a*b^2 - A*b^3)*sqrt(-b^2*c + a*b*d)*d^3*arctan(sqrt(-b
^2*c + a*b*d)*sqrt(d*x + c)/(b*d*x + b*c)) - (8*D*b^4*c^3 - 15*(D*a^3*b - C
*a^2*b^2 + B*a*b^3)*d^3 + 5*(D*a^2*b^2*c - (C*a*b^3 - 3*B*b^4)*c)*d^2 + 3*(
D*b^4*c*d^2 - D*a*b^3*d^3)*x^2 + 2*(D*a*b^3*c^2 - 5*C*b^4*c^2)*d - (4*D*b^4
```

```
*c^2*d - 5*(D*a^2*b^2 - C*a*b^3)*d^3 + (D*a*b^3*c - 5*C*b^4*c)*d^2)*x)*sqrt
(d*x + c))/(b^5*c*d^3 - a*b^4*d^4)]
```

Sympy [A] (verification not implemented)

Time = 3.94 (sec) , antiderivative size = 275, normalized size of antiderivative = 1.46

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)\sqrt{c+dx}}\,dx$$
$$= \begin{cases} \frac{2\left(\frac{D(c+dx)^{\frac{5}{2}}}{5bd^2}+\frac{(c+dx)^{\frac{3}{2}}(Cbd-Dad-2Dbc)}{3b^2d^2}+\frac{\sqrt{c+dx}\left(Bb^2d^2-Cabd^2-Cb^2cd+Da^2d^2+Dabcd+Db^2c^2\right)}{b^3d^2}-\frac{d\left(-Ab^3+Bab^2-Ca^2b+Da^3\right)\operatorname{atan}\left(\frac{\sqrt{c+dx}}{\sqrt{\frac{ad-bc}{b}}}\right)}{b^4\sqrt{\frac{ad-bc}{b}}}\right.}{d} & \\ \frac{\frac{Dx^3}{3b}+\frac{x^2(Cb-Da)}{2b^2}+\frac{x\left(Bb^2-Cab+Da^2\right)}{b^3}-\frac{\left(-Ab^3+Bab^2-Ca^2b+Da^3\right)\left(\begin{cases}\frac{x}{a} & \text{for}\ b=0\\ \frac{\log\left(a+bx\right)}{b} & \text{otherwise}\end{cases}\right)}{b^3}}{\sqrt{c}} & \end{cases}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(b*x+a)/(d*x+c)**(1/2),x)
```

```
[Out] Piecewise((2*(D*(c + d*x)**(5/2)/(5*b*d**2) + (c + d*x)**(3/2)*(C*b*d - D*a
*d - 2*D*b*c)/(3*b**2*d**2) + sqrt(c + d*x)*(B*b**2*d**2 - C*a*b*d**2 - C*b
**2*c*d + D*a**2*d**2 + D*a*b*c*d + D*b**2*c**2)/(b**3*d**2) - d*(-A*b**3 +
 B*a*b**2 - C*a**2*b + D*a**3)*atan(sqrt(c + d*x)/sqrt((a*d - b*c)/b))/(b**
4*sqrt((a*d - b*c)/b)))/d, Ne(d, 0)), ((D*x**3/(3*b) + x**2*(C*b - D*a)/(2*
b**2) + x*(B*b**2 - C*a*b + D*a**2)/b**3 - (-A*b**3 + B*a*b**2 - C*a**2*b +
 D*a**3)*Piecewise((x/a, Eq(b, 0)), (log(a + b*x)/b, True))/b**3)/sqrt(c),
True))
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)\sqrt{c+dx}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)/(d*x+c)^(1/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 248, normalized size of antiderivative = 1.32

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)\sqrt{c+dx}}\,dx = -\frac{2\,(Da^3-Ca^2b+Bab^2-Ab^3)\arctan\left(\frac{\sqrt{dx+c}b}{\sqrt{-b^2c+abd}}\right)}{\sqrt{-b^2c+abd}b^3} + \frac{2\left(3\,(dx+c)^{\frac{5}{2}}Db^4d^{12}-10\,(dx+c)^{\frac{3}{2}}Db^4cd^{12}+15\,\sqrt{dx+c}Db^4c^2d^{12}-5\,(dx+c)^{\frac{3}{2}}Dab^3d^{13}+5\,(dx+c)^{\frac{3}{2}}\right.}{}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)/(d*x+c)^(1/2),x, algorithm="giac")
```

```
[Out] -2*(D*a^3 - C*a^2*b + B*a*b^2 - A*b^3)*arctan(sqrt(d*x + c)*b/sqrt(-b^2*c +
 a*b*d))/(sqrt(-b^2*c + a*b*d)*b^3) + 2/15*(3*(d*x + c)^(5/2)*D*b^4*d^12 -
10*(d*x + c)^(3/2)*D*b^4*c*d^12 + 15*sqrt(d*x + c)*D*b^4*c^2*d^12 - 5*(d*x
+ c)^(3/2)*D*a*b^3*d^13 + 5*(d*x + c)^(3/2)*C*b^4*d^13 + 15*sqrt(d*x + c)*D
*a*b^3*c*d^13 - 15*sqrt(d*x + c)*C*b^4*c*d^13 + 15*sqrt(d*x + c)*D*a^2*b^2*
d^14 - 15*sqrt(d*x + c)*C*a*b^3*d^14 + 15*sqrt(d*x + c)*B*b^4*d^14)/(b^5*d^
15)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)\sqrt{c+dx}}\,dx = \int \frac{A+B\,x+C\,x^2+x^3\,D}{(a+b\,x)\,\sqrt{c+d\,x}}\,dx$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)*(c + d*x)^(1/2)),x)
```

```
[Out] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)*(c + d*x)^(1/2)), x)
```

3.6 $\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2\sqrt{c+dx}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 201

$$\begin{aligned}\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2\sqrt{c+dx}}\,dx &= \frac{2(bCd-bcD-2adD)\sqrt{c+dx}}{b^3d^2}-\frac{\left(A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}\right)\sqrt{c+dx}}{(bc-ad)(a+bx)}+\frac{2D(c+dx)^{3/2}}{3b^2d^2}\\ &\quad -\frac{\left(b^3(2Bc-Ad)-ab^2(4cC+Bd)-5a^3dD+3a^2b(Cd+2cD)\right)\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{7/2}(bc-ad)^{3/2}}\end{aligned}$$

```
[Out] 2/3*D*(d*x+c)^(3/2)/b^2/d^2-(b^3*(-A*d+2*B*c)-a*b^2*(B*d+4*C*c)-5*a^3*d*D+3
*a^2*b*(C*d+2*D*c))*arctanh(b^(1/2)*(d*x+c)^(1/2)/(-a*d+b*c)^(1/2))/b^(7/2)
/(-a*d+b*c)^(3/2)+2*(C*b*d-2*D*a*d-D*b*c)*(d*x+c)^(1/2)/b^3/d^2-(A-a*(B*b^2
-C*a*b+D*a^2)/b^3)*(d*x+c)^(1/2)/(-a*d+b*c)/(b*x+a)
```

Rubi [A] (verified)

Time = 0.37 (sec) , antiderivative size = 201, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {1635, 911, 1167, 214}

$$\begin{aligned}&\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2\sqrt{c+dx}}\,dx\\&=-\frac{\sqrt{c+dx}\left(A-\frac{a(a^2D-abC+b^2B)}{b^3}\right)}{(a+bx)(bc-ad)}\\&\quad-\frac{\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)(-5a^3dD+3a^2b(2cD+Cd)-ab^2(Bd+4cC)+b^3(2Bc-Ad))}{b^{7/2}(bc-ad)^{3/2}}\\&\quad+\frac{2\sqrt{c+dx}(-2adD-bcD+bCd)}{b^3d^2}+\frac{2D(c+dx)^{3/2}}{3b^2d^2}\end{aligned}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^2*Sqrt[c + d*x]),x]
```

```
[Out] (2*(b*C*d - b*c*D - 2*a*d*D)*Sqrt[c + d*x])/(b^3*d^2) - ((A - (a*(b^2*B - a
*b*C + a^2*D))/b^3)*Sqrt[c + d*x])/((b*c - a*d)*(a + b*x)) + (2*D*(c + d*x)
^(3/2))/(3*b^2*d^2) - ((b^3*(2*B*c - A*d) - a*b^2*(4*c*C + B*d) - 5*a^3*d*D
 + 3*a^2*b*(C*d + 2*c*D))*ArcTanh[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[b*c - a*d]])
/(b^(7/2)*(b*c - a*d)^(3/2))
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 911

```
Int[((d_.) + (e_.)*(x_))^(m_)*((f_.) + (g_.)*(x_))^(n_)*((a_.) + (b_.)*(x_)
 + (c_.)*(x_)^2)^(p_.), x_Symbol] :> With[{q = Denominator[m]}, Dist[q/e, S
ubst[Int[x^(q*(m + 1) - 1)*((e*f - d*g)/e + g*(x^q/e))^n*((c*d^2 - b*d*e +
a*e^2)/e^2 - (2*c*d - b*e)*(x^q/e^2) + c*(x^(2*q)/e^2))^p, x], x, (d + e*x)
^(1/q)], x]] /; FreeQ[{a, b, c, d, e, f, g}, x] && NeQ[e*f - d*g, 0] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IntegersQ[n, p] && Fra
ctionQ[m]
```

Rule 1167

```
Int[((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.),
 x_Symbol] :> Int[ExpandIntegrand[(d + e*x^2)^q*(a + b*x^2 + c*x^4)^p, x],
x] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e
 + a*e^2, 0] && IGtQ[p, 0] && IGtQ[q, -2]
```

Rule 1635

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol] :
> With[{Qx = PolynomialQuotient[Px, a + b*x, x], R = PolynomialRemainder[Px
, a + b*x, x]}, Simp[R*(a + b*x)^(m + 1)*((c + d*x)^(n + 1)/((m + 1)*(b*c -
 a*d))), x] + Dist[1/((m + 1)*(b*c - a*d)), Int[(a + b*x)^(m + 1)*(c + d*x)
^n*ExpandToSum[(m + 1)*(b*c - a*d)*Qx - d*R*(m + n + 2), x], x], x]] /; Fre
eQ[{a, b, c, d, n}, x] && PolyQ[Px, x] && ILtQ[m, -1] && GtQ[Expon[Px, x],
2]
```

Rubi steps

$$\text{integral} = -\frac{\left(A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}\right)\sqrt{c+dx}}{(bc-ad)(a+bx)} + \frac{\int \frac{-\frac{b^3(2Bc-Ad)-ab^2(2cC+Bd)-a^3dD+a^2b(Cd+2cD)}{2b^3}-\frac{(bc-ad)(bC-aD)x}{b^2}-\left(c-\frac{ad}{b}\right)Dx^2}{(a+bx)\sqrt{c+dx}}\,dx}{-bc+ad}$$

$$= -\frac{\left(A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}\right)\sqrt{c+dx}}{(bc-ad)(a+bx)} - \frac{2\text{Subst}\Bigg(\int \frac{\frac{-c^2\left(c-\frac{ad}{b}\right)D+\frac{cd(bc-ad)(bC-aD)}{b^2}-\frac{d^2\left(b^3(2Bc-Ad)-ab^2(2cC+Bd)-a^3dD+a^2b(Cd+2cD)\right)}{2b^3}}{d^2}-\frac{\left(-2c\left(c-\frac{ad}{b}\right)D+\frac{d(bc-ad)(b}{b^2}}{d^2}}{\frac{-bc+ad}{d}+\frac{bx^2}{d}}}{d(bc-ad)}$$

$$= -\frac{\left(A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}\right)\sqrt{c+dx}}{(bc-ad)(a+bx)} - \frac{2\text{Subst}\left(\int \left(-\frac{(bc-ad)(bCd-bcD-2adD)}{b^3d}-\frac{(bc-ad)Dx^2}{b^2d}+\frac{-2b^3Bc+4ab^2cC+Ab^3d+ab^2Bd-3a^2bCd-6a^2bcD+5a^3dD}{2b^3\left(a-\frac{bc}{d}+\frac{bx^2}{d}\right)}\right.\right.}{d(bc-ad)}$$

$$= \frac{2(bCd-bcD-2adD)\sqrt{c+dx}}{b^3d^2} - \frac{\left(A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}\right)\sqrt{c+dx}}{(bc-ad)(a+bx)} + \frac{2D(c+dx)^{3/2}}{3b^2d^2} + \frac{\left(b^3(2Bc-Ad)-ab^2(4cC+Bd)-5a^3dD+3a^2b(Cd+2cD)\right)\text{Subst}\left(\int \frac{1}{a-\frac{bc}{d}+\frac{bx^2}{d}}\,dx,x,\sqrt{c+d}\right.}{b^3d(bc-ad)}$$

$$= \frac{2(bCd-bcD-2adD)\sqrt{c+dx}}{b^3d^2} - \frac{\left(A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}\right)\sqrt{c+dx}}{(bc-ad)(a+bx)} + \frac{2D(c+dx)^{3/2}}{3b^2d^2} - \frac{\left(b^3(2Bc-Ad)-ab^2(4cC+Bd)-5a^3dD+3a^2b(Cd+2cD)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{7/2}(bc-ad)^{3/2}}$$

Mathematica [A] (verified)

Time = 0.59 (sec) , antiderivative size = 231, normalized size of antiderivative = 1.15

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2\sqrt{c+dx}}\,dx = -\frac{\sqrt{c+dx}(-15a^3d^2D+a^2bd(9Cd+8cD-10dDx)+b^3(3Ad^2-2cx(3Cd-2cD+dDx))+ab^2(4c^2D-}{3b^3d^2(bc-ad)(a+bx)} - \frac{(b^3(2Bc-Ad)-ab^2(4cC+Bd)-5a^3dD+3a^2b(Cd+2cD))\arctan\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{-bc+ad}}\right)}{b^{7/2}(-bc+ad)^{3/2}}$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^2*Sqrt[c + d*x]),x]
```

```
[Out] -1/3*(Sqrt[c + d*x]*(-15*a^3*d^2*D + a^2*b*d*(9*C*d + 8*c*D - 10*d*D*x) + b
^3*(3*A*d^2 - 2*c*x*(3*C*d - 2*c*D + d*D*x)) + a*b^2*(4*c^2*D - 6*c*d*(C -
D*x) + d^2*(-3*B + 6*C*x + 2*D*x^2))))/(b^3*d^2*(b*c - a*d)*(a + b*x)) - ((
b^3*(2*B*c - A*d) - a*b^2*(4*c*C + B*d) - 5*a^3*d*D + 3*a^2*b*(C*d + 2*c*D)
)*ArcTan[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[-(b*c) + a*d]])/(b^(7/2)*(-(b*c) + a*
d)^(3/2))
```

Maple [A] (verified)

Time = 1.77 (sec) , antiderivative size = 226, normalized size of antiderivative = 1.12

method	result
derivativedivides	$\frac{\frac{2\left(\frac{D(dx+c)^{\frac{3}{2}}b}{3}+dbC\sqrt{dx+c}-2Dad\sqrt{dx+c}-Dcb\sqrt{dx+c}\right)}{b^3}+\frac{2d^2\left(\frac{d\left(b^3A-a\,b^2B+C\,a^2b-Da^3\right)\sqrt{dx+c}}{2(ad-bc)((dx+c)b+ad-bc)}+\frac{\left(A\,b^3d+Ba\,b^2d-2B\,b^3c-3C\right.}{}\right.}{b^3}}{d^2}$
default	$\frac{\frac{2\left(\frac{D(dx+c)^{\frac{3}{2}}b}{3}+dbC\sqrt{dx+c}-2Dad\sqrt{dx+c}-Dcb\sqrt{dx+c}\right)}{b^3}+\frac{2d^2\left(\frac{d\left(b^3A-a\,b^2B+C\,a^2b-Da^3\right)\sqrt{dx+c}}{2(ad-bc)((dx+c)b+ad-bc)}+\frac{\left(A\,b^3d+Ba\,b^2d-2B\,b^3c-3C\right.}{}\right.}{b^3}}{d^2}$
pseudoelliptic	$\frac{\left((Ad-2Bc)b^3+a\,b^2(Bd+4Cc)-3a^2b(Cd+2Dc)+5a^3dD\right)(bx+a)d^2\arctan\left(\frac{b\sqrt{dx+c}}{\sqrt{(ad-bc)b}}\right)+\sqrt{dx+c}\,\sqrt{(ad-bc)b}\left(\left(A\,d^2-2x\right.\right.}{\sqrt{(ad-bc)b}\,d^2b^3}$

```
[In] int((D*x^3+C*x^2+B*x+A)/(b*x+a)^2/(d*x+c)^(1/2),x,method=_RETURNVERBOSE)
```

```
[Out] 2/d^2*(1/b^3*(1/3*D*(d*x+c)^(3/2)*b+d*b*C*(d*x+c)^(1/2)-2*D*a*d*(d*x+c)^(1/
2)-D*c*b*(d*x+c)^(1/2))+d^2/b^3*(1/2*d*(A*b^3-B*a*b^2+C*a^2*b-D*a^3)/(a*d-b
*c)*(d*x+c)^(1/2)/((d*x+c)*b+a*d-b*c)+1/2*(A*b^3*d+B*a*b^2*d-2*B*b^3*c-3*C*
a^2*b*d+4*C*a*b^2*c+5*D*a^3*d-6*D*a^2*b*c)/(a*d-b*c)/((a*d-b*c)*b)^(1/2)*ar
ctan(b*(d*x+c)^(1/2)/((a*d-b*c)*b)^(1/2))))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 495 vs. $2(182) = 364$.

Time $= 0.28$ (sec) , antiderivative size $= 1004$, normalized size of antiderivative $= 5.00$

$$\int \frac{A + Bx + Cx^2 + Dx^3}{(a + bx)^2\sqrt{c + dx}}\, dx$$

$$= \Bigg[-\frac{3\,((5\,Da^4 - 3\,Ca^3b + Ba^2b^2 + Aab^3)d^3 - 2\,(3\,Da^3bc - (2\,Ca^2b^2 - Bab^3)c)d^2 + ((5\,Da^3b - 3\,Ca^2b^2 +}{}$$

$$-\frac{3\,((5\,Da^4 - 3\,Ca^3b + Ba^2b^2 + Aab^3)d^3 - 2\,(3\,Da^3bc - (2\,Ca^2b^2 - Bab^3)c)d^2 + ((5\,Da^3b - 3\,Ca^2b^2 +}{}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^2/(d*x+c)^(1/2),x, algorithm="fricas"
)

[Out] [-1/6*(3*((5*D*a^4 - 3*C*a^3*b + B*a^2*b^2 + A*a*b^3)*d^3 - 2*(3*D*a^3*b*c
- (2*C*a^2*b^2 - B*a*b^3)*c)*d^2 + ((5*D*a^3*b - 3*C*a^2*b^2 + B*a*b^3 + A*
b^4)*d^3 - 2*(3*D*a^2*b^2*c - (2*C*a*b^3 - B*b^4)*c)*d^2)*x)*sqrt(b^2*c - a
*b*d)*log((b*d*x + 2*b*c - a*d - 2*sqrt(b^2*c - a*b*d)*sqrt(d*x + c))/(b*x
+ a)) + 2*(4*D*a*b^4*c^3 + 3*(5*D*a^4*b - 3*C*a^3*b^2 + B*a^2*b^3 - A*a*b^4
)*d^3 - (23*D*a^3*b^2*c - 3*(5*C*a^2*b^3 - B*a*b^4 + A*b^5)*c)*d^2 - 2*(D*b
^5*c^2*d - 2*D*a*b^4*c*d^2 + D*a^2*b^3*d^3)*x^2 + 2*(2*D*a^2*b^3*c^2 - 3*C*
a*b^4*c^2)*d + 2*(2*D*b^5*c^3 + (5*D*a^3*b^2 - 3*C*a^2*b^3)*d^3 - 2*(4*D*a^
2*b^3*c - 3*C*a*b^4*c)*d^2 + (D*a*b^4*c^2 - 3*C*b^5*c^2)*d)*x)*sqrt(d*x + c
))/(a*b^6*c^2*d^2 - 2*a^2*b^5*c*d^3 + a^3*b^4*d^4 + (b^7*c^2*d^2 - 2*a*b^6*
c*d^3 + a^2*b^5*d^4)*x), -1/3*(3*((5*D*a^4 - 3*C*a^3*b + B*a^2*b^2 + A*a*b^
3)*d^3 - 2*(3*D*a^3*b*c - (2*C*a^2*b^2 - B*a*b^3)*c)*d^2 + ((5*D*a^3*b - 3*
C*a^2*b^2 + B*a*b^3 + A*b^4)*d^3 - 2*(3*D*a^2*b^2*c - (2*C*a*b^3 - B*b^4)*c
)*d^2)*x)*sqrt(-b^2*c + a*b*d)*arctan(sqrt(-b^2*c + a*b*d)*sqrt(d*x + c)/(b
*d*x + b*c)) + (4*D*a*b^4*c^3 + 3*(5*D*a^4*b - 3*C*a^3*b^2 + B*a^2*b^3 - A*
a*b^4)*d^3 - (23*D*a^3*b^2*c - 3*(5*C*a^2*b^3 - B*a*b^4 + A*b^5)*c)*d^2 - 2
*(D*b^5*c^2*d - 2*D*a*b^4*c*d^2 + D*a^2*b^3*d^3)*x^2 + 2*(2*D*a^2*b^3*c^2 -
 3*C*a*b^4*c^2)*d + 2*(2*D*b^5*c^3 + (5*D*a^3*b^2 - 3*C*a^2*b^3)*d^3 - 2*(4
*D*a^2*b^3*c - 3*C*a*b^4*c)*d^2 + (D*a*b^4*c^2 - 3*C*b^5*c^2)*d)*x)*sqrt(d*
x + c))/(a*b^6*c^2*d^2 - 2*a^2*b^5*c*d^3 + a^3*b^4*d^4 + (b^7*c^2*d^2 - 2*a
*b^6*c*d^3 + a^2*b^5*d^4)*x)]
```

Sympy [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2\sqrt{c+dx}}\,dx = \text{Timed out}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(b*x+a)**2/(d*x+c)**(1/2),x)
```

```
[Out] Timed out
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2\sqrt{c+dx}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^2/(d*x+c)^(1/2),x, algorithm="maxima"
)
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [A] (verification not implemented)

none

Time = 0.35 (sec) , antiderivative size = 271, normalized size of antiderivative = 1.35

$$\begin{aligned}
&\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2\sqrt{c+dx}}\,dx \\
&= \frac{(6\,Da^2bc-4\,Cab^2c+2\,Bb^3c-5\,Da^3d+3\,Ca^2bd-Bab^2d-Ab^3d)\arctan\left(\frac{\sqrt{dx+c}b}{\sqrt{-b^2c+abd}}\right)}{(b^4c-ab^3d)\sqrt{-b^2c+abd}} \\
&\quad+\frac{\sqrt{dx+c}Da^3d-\sqrt{dx+c}Ca^2bd+\sqrt{dx+c}Bab^2d-\sqrt{dx+c}Ab^3d}{(b^4c-ab^3d)((dx+c)b-bc+ad)} \\
&\quad+\frac{2\left((dx+c)^{\frac{3}{2}}Db^4d^4-3\sqrt{dx+c}Db^4cd^4-6\sqrt{dx+c}Dab^3d^5+3\sqrt{dx+c}Cb^4d^5\right)}{3\,b^6d^6}
\end{aligned}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^2/(d*x+c)^(1/2),x, algorithm="giac")
```

```
[Out] (6*D*a^2*b*c - 4*C*a*b^2*c + 2*B*b^3*c - 5*D*a^3*d + 3*C*a^2*b*d - B*a*b^2*
d - A*b^3*d)*arctan(sqrt(d*x + c)*b/sqrt(-b^2*c + a*b*d))/((b^4*c - a*b^3*d
```

```
)*sqrt(-b^2*c + a*b*d)) + (sqrt(d*x + c)*D*a^3*d - sqrt(d*x + c)*C*a^2*b*d
+ sqrt(d*x + c)*B*a*b^2*d - sqrt(d*x + c)*A*b^3*d)/((b^4*c - a*b^3*d)*((d*x
 + c)*b - b*c + a*d)) + 2/3*((d*x + c)^(3/2)*D*b^4*d^4 - 3*sqrt(d*x + c)*D*
b^4*c*d^4 - 6*sqrt(d*x + c)*D*a*b^3*d^5 + 3*sqrt(d*x + c)*C*b^4*d^5)/(b^6*d
^6)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2\sqrt{c+dx}}\,dx=\int \frac{A+B\,x+C\,x^2+x^3\,D}{{\left(a+b\,x\right)}^2\,\sqrt{c+d\,x}}\,dx$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^2*(c + d*x)^(1/2)),x)
[Out] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^2*(c + d*x)^(1/2)), x)
```

3.7 $\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3\sqrt{c+dx}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 279

$$
\begin{aligned}
\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3\sqrt{c+dx}}\,dx &= \frac{2D\sqrt{c+dx}}{b^3d}-\frac{\left(Ab^3-a\left(b^2B-abC+a^2D\right)\right)\sqrt{c+dx}}{2b^3(bc-ad)(a+bx)^2} \\
&\quad -\frac{\left(b^3(4Bc-3Ad)-ab^2(8cC+Bd)-9a^3dD+a^2b(5Cd+12cD)\right)\sqrt{c+dx}}{4b^3(bc-ad)^2(a+bx)} \\
&\quad -\frac{\left(b^3\left(8c^2C-4Bcd+3Ad^2\right)-15a^3d^2D+3a^2bd(Cd+12cD)-ab^2\left(8cCd-Bd^2+24c^2D\right)\right)\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c}}{\sqrt{bc-}}\right.}{4b^{7/2}(bc-ad)^{5/2}}
\end{aligned}
$$

```
[Out] -1/4*(b^3*(3*A*d^2-4*B*c*d+8*C*c^2)-15*a^3*d^2*D+3*a^2*b*d*(C*d+12*D*c)-a*b
^2*(-B*d^2+8*C*c*d+24*D*c^2))*arctanh(b^(1/2)*(d*x+c)^(1/2)/(-a*d+b*c)^(1/2
))/b^(7/2)/(-a*d+b*c)^(5/2)+2*D*(d*x+c)^(1/2)/b^3/d-1/2*(A*b^3-a*(B*b^2-C*a
*b+D*a^2))*(d*x+c)^(1/2)/b^3/(-a*d+b*c)/(b*x+a)^2-1/4*(b^3*(-3*A*d+4*B*c)-a
*b^2*(B*d+8*C*c)-9*a^3*d*D+a^2*b*(5*C*d+12*D*c))*(d*x+c)^(1/2)/b^3/(-a*d+b*
c)^2/(b*x+a)
```

Rubi [A] (verified)

Time = 0.49 (sec) , antiderivative size = 279, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.156, Rules used

$= \{1635, 911, 1171, 396, 214\}$

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3\sqrt{c+dx}}\,dx = -\frac{\sqrt{c+dx}(Ab^3-a(a^2D-abC+b^2B))}{2b^3(a+bx)^2(bc-ad)}$$

$$-\frac{\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)(-15a^3d^2D+3a^2bd(12cD+Cd)-ab^2(-Bd^2+24c^2D+8cCd)+b^3(3Ad^2-4Bc}{4b^{7/2}(bc-ad)^{5/2}}$$

$$-\frac{\sqrt{c+dx}(-9a^3dD+a^2b(12cD+5Cd)-ab^2(Bd+8cC)+b^3(4Bc-3Ad))}{4b^3(a+bx)(bc-ad)^2}$$

$$+\frac{2D\sqrt{c+dx}}{b^3d}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^3*Sqrt[c + d*x]),x]
```

```
[Out] (2*D*Sqrt[c + d*x])/(b^3*d) - ((A*b^3 - a*(b^2*B - a*b*C + a^2*D))*Sqrt[c +
 d*x])/(2*b^3*(b*c - a*d)*(a + b*x)^2) - ((b^3*(4*B*c - 3*A*d) - a*b^2*(8*c
*C + B*d) - 9*a^3*d*D + a^2*b*(5*C*d + 12*c*D))*Sqrt[c + d*x])/(4*b^3*(b*c
- a*d)^2*(a + b*x)) - ((b^3*(8*c^2*C - 4*B*c*d + 3*A*d^2) - 15*a^3*d^2*D +
3*a^2*b*d*(C*d + 12*c*D) - a*b^2*(8*c*C*d - B*d^2 + 24*c^2*D))*ArcTanh[(Sqr
t[b]*Sqrt[c + d*x])/Sqrt[b*c - a*d]])/(4*b^(7/2)*(b*c - a*d)^(5/2))
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 396

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_)*((c_) + (d_.)*(x_)^(n_)), x_Symbol] :> Si
mp[d*x*((a + b*x^n)^(p + 1)/(b*(n*(p + 1) + 1))), x] - Dist[(a*d - b*c*(n*(
p + 1) + 1))/(b*(n*(p + 1) + 1)), Int[(a + b*x^n)^p, x], x] /; FreeQ[{a, b,
 c, d, n}, x] && NeQ[b*c - a*d, 0] && NeQ[n*(p + 1) + 1, 0]
```

Rule 911

```
Int[((d_.) + (e_.)*(x_))^(m_)*((f_.) + (g_.)*(x_))^(n_)*((a_.) + (b_.)*(x_)
 + (c_.)*(x_)^2)^(p_.), x_Symbol] :> With[{q = Denominator[m]}, Dist[q/e, S
ubst[Int[x^(q*(m + 1) - 1)*((e*f - d*g)/e + g*(x^q/e))^n*((c*d^2 - b*d*e +
a*e^2)/e^2 - (2*c*d - b*e)*(x^q/e^2) + c*(x^(2*q)/e^2))^p, x], x, (d + e*x)
^(1/q)], x]] /; FreeQ[{a, b, c, d, e, f, g}, x] && NeQ[e*f - d*g, 0] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IntegersQ[n, p] && Fra
ctionQ[m]
```

Rule 1171

```
Int[((d_) + (e_.)*(x_)^2)^(q_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.),
x_Symbol] :> With[{Qx = PolynomialQuotient[(a + b*x^2 + c*x^4)^p, d + e*x^2
```

```
, x], R = Coeff[PolynomialRemainder[(a + b*x^2 + c*x^4)^p, d + e*x^2, x], x
, 0]}, Simp[(-R)*x*((d + e*x^2)^(q + 1)/(2*d*(q + 1))), x] + Dist[1/(2*d*(q
 + 1)), Int[(d + e*x^2)^(q + 1)*ExpandToSum[2*d*(q + 1)*Qx + R*(2*q + 3), x
], x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2
 - b*d*e + a*e^2, 0] && IGtQ[p, 0] && LtQ[q, -1]
```

Rule 1635

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol] :
> With[{Qx = PolynomialQuotient[Px, a + b*x, x], R = PolynomialRemainder[Px
, a + b*x, x]}, Simp[R*(a + b*x)^(m + 1)*((c + d*x)^(n + 1)/((m + 1)*(b*c -
 a*d))), x] + Dist[1/((m + 1)*(b*c - a*d)), Int[(a + b*x)^(m + 1)*(c + d*x)
^n*ExpandToSum[(m + 1)*(b*c - a*d)*Qx - d*R*(m + n + 2), x], x], x]] /; Fre
eQ[{a, b, c, d, n}, x] && PolyQ[Px, x] && ILtQ[m, -1] && GtQ[Expon[Px, x],
2]
```

Rubi steps

$$\text{integral} = -\frac{\left(Ab^3-a\left(b^2B-abC+a^2D\right)\right)\sqrt{c+dx}}{2b^3(bc-ad)(a+bx)^2} - \frac{\int \frac{-\frac{b^3(4Bc-3Ad)-ab^2(4cC+Bd)-a^3dD+a^2b(Cd+4cD)}{2b^3}-\frac{2(bc-ad)(bC-aD)x}{b^2}-2\left(c-\frac{ad}{b}\right)Dx^2}{(a+bx)^2\sqrt{c+dx}}\,dx}{2(bc-ad)}$$

$$= -\frac{\left(Ab^3-a\left(b^2B-abC+a^2D\right)\right)\sqrt{c+dx}}{2b^3(bc-ad)(a+bx)^2} - \frac{\text{Subst}\Biggl(\int \frac{\frac{-2c^2\left(c-\frac{ad}{b}\right)D+\frac{2cd(bc-ad)(bC-aD)}{b^2}-\frac{d^2\left(b^3(4Bc-3Ad)-ab^2(4cC+Bd)-a^3dD+a^2b(Cd+4cD)\right)}{2b^3}}{d^2}-\frac{\bigl(-4c\left(c-\frac{ad}{b}\right)D+\frac{2d(bc-ad)(}{b^2}}{d^2}}{\left(\frac{-bc+ad}{d}+\frac{bx^2}{d}\right)^2}}{d(bc-ad)}$$

$$= -\frac{\left(Ab^3-a\left(b^2B-abC+a^2D\right)\right)\sqrt{c+dx}}{2b^3(bc-ad)(a+bx)^2} - \frac{\left(b^3(4Bc-3Ad)-ab^2(8cC+Bd)-9a^3dD+a^2b(5Cd+12cD)\right)\sqrt{c+dx}}{4b^3(bc-ad)^2(a+bx)} - \frac{\text{Subst}\left(\int \frac{\frac{1}{2}\left(4Bc-\frac{8c^2C}{d}-3Ad+\frac{a(8cC-Bd)}{b}+\frac{8c^3D}{d^2}+\frac{7a^3dD}{b^3}-\frac{3a^2(Cd+4cD)}{b^2}\right)-\frac{4(bc-ad)^2Dx^2}{b^2d^2}}{\frac{-bc+ad}{d}+\frac{bx^2}{d}}\,dx,x,\sqrt{c+dx}\right)}{2(bc-ad)^2}$$

$$= \frac{2D\sqrt{c+dx}}{b^3d} - \frac{(Ab^3 - a(b^2B - abC + a^2D))\sqrt{c+dx}}{2b^3(bc-ad)(a+bx)^2}$$
$$- \frac{(b^3(4Bc - 3Ad) - ab^2(8cC + Bd) - 9a^3dD + a^2b(5Cd + 12cD))\sqrt{c+dx}}{4b^3(bc-ad)^2(a+bx)}$$
$$+ \frac{(b^3(8c^2C - 4Bcd + 3Ad^2) - 15a^3d^2D + 3a^2bd(Cd + 12cD) - ab^2(8cCd - Bd^2 + 24c^2D))\,\mathrm{Subs}}{4b^3d(bc-ad)^2}$$

$$= \frac{2D\sqrt{c+dx}}{b^3d} - \frac{(Ab^3 - a(b^2B - abC + a^2D))\sqrt{c+dx}}{2b^3(bc-ad)(a+bx)^2}$$
$$- \frac{(b^3(4Bc - 3Ad) - ab^2(8cC + Bd) - 9a^3dD + a^2b(5Cd + 12cD))\sqrt{c+dx}}{4b^3(bc-ad)^2(a+bx)}$$
$$- \frac{(b^3(8c^2C - 4Bcd + 3Ad^2) - 15a^3d^2D + 3a^2bd(Cd + 12cD) - ab^2(8cCd - Bd^2 + 24c^2D))\,\mathrm{tanh}}{4b^{7/2}(bc-ad)^{5/2}}$$

Mathematica [A] (verified)

Time = 0.86 (sec) , antiderivative size = 298, normalized size of antiderivative = 1.07

$$\int \frac{A + Bx + Cx^2 + Dx^3}{(a+bx)^3\sqrt{c+dx}}\,dx$$
$$= \frac{\frac{\sqrt{b}\sqrt{c+dx}(15a^4d^2D + Ab^3d(-2bc+5ad+3bdx) + 4b^4cx(-Bd+2cDx) + a^3bd(-3Cd-26cD+25dDx) + ab^3(Bd(-2c+dx) + 8cx(Cd+2cD-2dDx)) +}{d(bc-ad)^2(a+bx)^2}}{}$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^3*Sqrt[c + d*x]),x]
```

```
[Out] ((Sqrt[b]*Sqrt[c + d*x]*(15*a^4*d^2*D + A*b^3*d*(-2*b*c + 5*a*d + 3*b*d*x)
+ 4*b^4*c*x*(-(B*d) + 2*c*D*x) + a^3*b*d*(-3*C*d - 26*c*D + 25*d*D*x) + a*b
^3*(B*d*(-2*c + d*x) + 8*c*x*(C*d + 2*c*D - 2*d*D*x)) + a^2*b^2*(8*c^2*D +
c*(6*C*d - 44*d*D*x) - d^2*(B + 5*C*x - 8*D*x^2))))/(d*(b*c - a*d)^2*(a + b
*x)^2) + ((b^3*(8*c^2*C - 4*B*c*d + 3*A*d^2) - 15*a^3*d^2*D + 3*a^2*b*d*(C*
d + 12*c*D) + a*b^2*(-8*c*C*d + B*d^2 - 24*c^2*D))*ArcTan[(Sqrt[b]*Sqrt[c +
 d*x])/Sqrt[-(b*c) + a*d]])/(-(b*c) + a*d)^(5/2))/(4*b^(7/2))
```

Maple [A] (verified)

Time = 1.84 (sec) , antiderivative size = 289, normalized size of antiderivative = 1.04

method	result
pseudoelliptic	$\frac{5((\frac{3Ab^4x}{5}+a(\frac{Bx}{5}+A)b^3-\frac{a^2(-8Dx^2+5Cx+B)b^2}{5}-\frac{3a^3(-\frac{25Dx}{3}+C)b}{5}+3Da^4)d^2-\frac{2bc((2Bx+A)b^3+a(8Dx^2-4Cx+B)b^2-}{5}}{}$
derivativedivides	$\frac{\frac{2D\sqrt{dx+c}}{b^3}+\frac{2d(\frac{\frac{bd(3Ab^3d+Bab^2d-4Bb^3c-5Ca^2bd+8Cab^2c+9a^3dD-12Da^2bc)(dx+c)^{\frac{3}{2}}}{8a^2d^2-16abcd+8b^2c^2}+\frac{(5Ab^3d-Bab^2d-4Bb^3c-3Ca^2bd+8}{8ad-}}{((dx+c)b+ad-bc)^2}}{}}{}$
default	$\frac{\frac{2D\sqrt{dx+c}}{b^3}+\frac{2d(\frac{\frac{bd(3Ab^3d+Bab^2d-4Bb^3c-5Ca^2bd+8Cab^2c+9a^3dD-12Da^2bc)(dx+c)^{\frac{3}{2}}}{8a^2d^2-16abcd+8b^2c^2}+\frac{(5Ab^3d-Bab^2d-4Bb^3c-3Ca^2bd+8}{8ad-}}{((dx+c)b+ad-bc)^2}}{}}{}$

```
[In] int((D*x^3+C*x^2+B*x+A)/(b*x+a)^3/(d*x+c)^(1/2),x,method=_RETURNVERBOSE)
```

```
[Out] 1/4*(5*((3/5*A*b^4*x+a*(1/5*B*x+A)*b^3-1/5*a^2*(-8*D*x^2+5*C*x+B)*b^2-3/5*a
^3*(-25/3*D*x+C)*b+3*D*a^4)*d^2-2/5*b*c*((2*B*x+A)*b^3+a*(8*D*x^2-4*C*x+B)*
b^2-3*a^2*(-22/3*D*x+C)*b+13*D*a^3)*d+8/5*D*b^2*c^2*(b*x+a)^2)*((a*d-b*c)*b
)^(1/2)*(d*x+c)^(1/2)+3*(b*x+a)^2*arctan(b*(d*x+c)^(1/2)/((a*d-b*c)*b)^(1/2
))*((b^3*A+1/3*a*b^2*B+C*a^2*b-5*D*a^3)*d^2-4/3*b*c*(B*b^2+2*C*a*b-9*D*a^2)
*d+8/3*b^2*c^2*(C*b-3*D*a))*d)/((a*d-b*c)*b)^(1/2)/(a*d-b*c)^2/b^3/(b*x+a)^
2/d
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 824 vs. 2(258) = 516.

Time = 0.32 (sec) , antiderivative size = 1661, normalized size of antiderivative = 5.95

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3\sqrt{c+dx}}\,dx = \text{Too large to display}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^3/(d*x+c)^(1/2),x, algorithm="fricas"
)
```

```
[Out] [1/8*(((15*D*a^5 - 3*C*a^4*b - B*a^3*b^2 - 3*A*a^2*b^3)*d^3 - 4*(9*D*a^4*b*
c - (2*C*a^3*b^2 + B*a^2*b^3)*c)*d^2 + ((15*D*a^3*b^2 - 3*C*a^2*b^3 - B*a*b
^4 - 3*A*b^5)*d^3 - 4*(9*D*a^2*b^3*c - (2*C*a*b^4 + B*b^5)*c)*d^2 + 8*(3*D*
a*b^4*c^2 - C*b^5*c^2)*d)*x^2 + 8*(3*D*a^3*b^2*c^2 - C*a^2*b^3*c^2)*d + 2*(
(15*D*a^4*b - 3*C*a^3*b^2 - B*a^2*b^3 - 3*A*a*b^4)*d^3 - 4*(9*D*a^3*b^2*c -
 (2*C*a^2*b^3 + B*a*b^4)*c)*d^2 + 8*(3*D*a^2*b^3*c^2 - C*a*b^4*c^2)*d)*x)*s
qrt(b^2*c - a*b*d)*log((b*d*x + 2*b*c - a*d + 2*sqrt(b^2*c - a*b*d)*sqrt(d*
```

```
x + c))/(b*x + a)) + 2*(8*D*a^2*b^4*c^3 - (15*D*a^5*b - 3*C*a^4*b^2 - B*a^3
*b^3 + 5*A*a^2*b^4)*d^3 + (41*D*a^4*b^2*c - (9*C*a^3*b^3 - B*a^2*b^4 - 7*A*
a*b^5)*c)*d^2 + 8*(D*b^6*c^3 - 3*D*a*b^5*c^2*d + 3*D*a^2*b^4*c*d^2 - D*a^3*
b^3*d^3)*x^2 - 2*(17*D*a^3*b^3*c^2 - (3*C*a^2*b^4 - B*a*b^5 - A*b^6)*c^2)*d
 + (16*D*a*b^5*c^3 - (25*D*a^4*b^2 - 5*C*a^3*b^3 + B*a^2*b^4 + 3*A*a*b^5)*d
^3 + (69*D*a^3*b^3*c - (13*C*a^2*b^4 - 5*B*a*b^5 - 3*A*b^6)*c)*d^2 - 4*(15*
D*a^2*b^4*c^2 - (2*C*a*b^5 - B*b^6)*c^2)*d)*x)*sqrt(d*x + c))/(a^2*b^7*c^3*
d - 3*a^3*b^6*c^2*d^2 + 3*a^4*b^5*c*d^3 - a^5*b^4*d^4 + (b^9*c^3*d - 3*a*b^
8*c^2*d^2 + 3*a^2*b^7*c*d^3 - a^3*b^6*d^4)*x^2 + 2*(a*b^8*c^3*d - 3*a^2*b^7
*c^2*d^2 + 3*a^3*b^6*c*d^3 - a^4*b^5*d^4)*x), -1/4*(((15*D*a^5 - 3*C*a^4*b
- B*a^3*b^2 - 3*A*a^2*b^3)*d^3 - 4*(9*D*a^4*b*c - (2*C*a^3*b^2 + B*a^2*b^3)
*c)*d^2 + ((15*D*a^3*b^2 - 3*C*a^2*b^3 - B*a*b^4 - 3*A*b^5)*d^3 - 4*(9*D*a^
2*b^3*c - (2*C*a*b^4 + B*b^5)*c)*d^2 + 8*(3*D*a*b^4*c^2 - C*b^5*c^2)*d)*x^2
 + 8*(3*D*a^3*b^2*c^2 - C*a^2*b^3*c^2)*d + 2*((15*D*a^4*b - 3*C*a^3*b^2 - B
*a^2*b^3 - 3*A*a*b^4)*d^3 - 4*(9*D*a^3*b^2*c - (2*C*a^2*b^3 + B*a*b^4)*c)*d
^2 + 8*(3*D*a^2*b^3*c^2 - C*a*b^4*c^2)*d)*x)*sqrt(-b^2*c + a*b*d)*arctan(sq
rt(-b^2*c + a*b*d)*sqrt(d*x + c)/(b*d*x + b*c)) - (8*D*a^2*b^4*c^3 - (15*D*
a^5*b - 3*C*a^4*b^2 - B*a^3*b^3 + 5*A*a^2*b^4)*d^3 + (41*D*a^4*b^2*c - (9*C
*a^3*b^3 - B*a^2*b^4 - 7*A*a*b^5)*c)*d^2 + 8*(D*b^6*c^3 - 3*D*a*b^5*c^2*d +
 3*D*a^2*b^4*c*d^2 - D*a^3*b^3*d^3)*x^2 - 2*(17*D*a^3*b^3*c^2 - (3*C*a^2*b^
4 - B*a*b^5 - A*b^6)*c^2)*d + (16*D*a*b^5*c^3 - (25*D*a^4*b^2 - 5*C*a^3*b^3
 + B*a^2*b^4 + 3*A*a*b^5)*d^3 + (69*D*a^3*b^3*c - (13*C*a^2*b^4 - 5*B*a*b^5
 - 3*A*b^6)*c)*d^2 - 4*(15*D*a^2*b^4*c^2 - (2*C*a*b^5 - B*b^6)*c^2)*d)*x)*s
qrt(d*x + c))/(a^2*b^7*c^3*d - 3*a^3*b^6*c^2*d^2 + 3*a^4*b^5*c*d^3 - a^5*b^
4*d^4 + (b^9*c^3*d - 3*a*b^8*c^2*d^2 + 3*a^2*b^7*c*d^3 - a^3*b^6*d^4)*x^2 +
 2*(a*b^8*c^3*d - 3*a^2*b^7*c^2*d^2 + 3*a^3*b^6*c*d^3 - a^4*b^5*d^4)*x)]
```

Sympy [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3\sqrt{c+dx}}\,dx=\text{Timed out}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(b*x+a)**3/(d*x+c)**(1/2),x)
```

```
[Out] Timed out
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3\sqrt{c+dx}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^3/(d*x+c)^(1/2),x, algorithm="maxima"
)
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 529 vs. $2(258) = 516$.

Time = 0.30 (sec) , antiderivative size = 529, normalized size of antiderivative = 1.90

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3\sqrt{c+dx}}\,dx =$$

$$-\frac{(24\,Dab^2c^2-8\,Cb^3c^2-36\,Da^2bcd+8\,Cab^2cd+4\,Bb^3cd+15\,Da^3d^2-3\,Ca^2bd^2-Bab^2d^2-3\,Ab^3d^2)\,a}{4\,(b^5c^2-2\,ab^4cd+a^2b^3d^2)\sqrt{-b^2c+abd}}$$

$$-\frac{12\,(dx+c)^{\frac{3}{2}}Da^2b^2cd-8\,(dx+c)^{\frac{3}{2}}Cab^3cd+4\,(dx+c)^{\frac{3}{2}}Bb^4cd-12\,\sqrt{dx+c}Da^2b^2c^2d+8\,\sqrt{dx+c}Cab^3}{}$$

$$+\frac{2\,\sqrt{dx+c}D}{b^3d}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^3/(d*x+c)^(1/2),x, algorithm="giac")
```

```
[Out] -1/4*(24*D*a*b^2*c^2 - 8*C*b^3*c^2 - 36*D*a^2*b*c*d + 8*C*a*b^2*c*d + 4*B*b
^3*c*d + 15*D*a^3*d^2 - 3*C*a^2*b*d^2 - B*a*b^2*d^2 - 3*A*b^3*d^2)*arctan(s
qrt(d*x + c)*b/sqrt(-b^2*c + a*b*d))/((b^5*c^2 - 2*a*b^4*c*d + a^2*b^3*d^2)
*sqrt(-b^2*c + a*b*d)) - 1/4*(12*(d*x + c)^(3/2)*D*a^2*b^2*c*d - 8*(d*x + c
)^(3/2)*C*a*b^3*c*d + 4*(d*x + c)^(3/2)*B*b^4*c*d - 12*sqrt(d*x + c)*D*a^2*
b^2*c^2*d + 8*sqrt(d*x + c)*C*a*b^3*c^2*d - 4*sqrt(d*x + c)*B*b^4*c^2*d - 9
*(d*x + c)^(3/2)*D*a^3*b*d^2 + 5*(d*x + c)^(3/2)*C*a^2*b^2*d^2 - (d*x + c)^
(3/2)*B*a*b^3*d^2 - 3*(d*x + c)^(3/2)*A*b^4*d^2 + 19*sqrt(d*x + c)*D*a^3*b*
c*d^2 - 11*sqrt(d*x + c)*C*a^2*b^2*c*d^2 + 3*sqrt(d*x + c)*B*a*b^3*c*d^2 +
5*sqrt(d*x + c)*A*b^4*c*d^2 - 7*sqrt(d*x + c)*D*a^4*d^3 + 3*sqrt(d*x + c)*C
*a^3*b*d^3 + sqrt(d*x + c)*B*a^2*b^2*d^3 - 5*sqrt(d*x + c)*A*a*b^3*d^3)/((b
^5*c^2 - 2*a*b^4*c*d + a^2*b^3*d^2)*((d*x + c)*b - b*c + a*d)^2) + 2*sqrt(d
*x + c)*D/(b^3*d)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3\sqrt{c+dx}}\,dx=\int \frac{A+B\,x+C\,x^2+x^3\,D}{{\left(a+b\,x\right)}^3\,\sqrt{c+d\,x}} \,dx$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^3*(c + d*x)^(1/2)),x)
[Out] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^3*(c + d*x)^(1/2)), x)
```

3.8 $\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4\sqrt{c+dx}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 375

$$\begin{aligned}\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4\sqrt{c+dx}}\,dx &= -\frac{(Ab^3-a(b^2B-abC+a^2D))\sqrt{c+dx}}{3b^3(bc-ad)(a+bx)^3}\\ &\quad-\frac{(b^3(6Bc-5Ad)-ab^2(12cC+Bd)-13a^3dD+a^2b(7Cd+18cD))\sqrt{c+dx}}{12b^3(bc-ad)^2(a+bx)^2}\\ &\quad-\frac{(b^3(8c^2C-6Bcd+5Ad^2)-11a^3d^2D+a^2bd(Cd+30cD)-ab^2(4cCd-Bd^2+24c^2D))\sqrt{c+dx}}{8b^3(bc-ad)^3(a+bx)}\\ &\quad+\frac{(5a^3d^3D+a^2bd^2(Cd-18cD)-ab^2d(4cCd-Bd^2-24c^2D)+b^3(8c^2Cd-6Bcd^2+5Ad^3-16c^3D))\,\text{arc}}{8b^{7/2}(bc-ad)^{7/2}}\end{aligned}$$

```
[Out] 1/8*(5*a^3*d^3*D+a^2*b*d^2*(C*d-18*D*c)-a*b^2*d*(-B*d^2+4*C*c*d-24*D*c^2)+b
^3*(5*A*d^3-6*B*c*d^2+8*C*c^2*d-16*D*c^3))*arctanh(b^(1/2)*(d*x+c)^(1/2)/(-
a*d+b*c)^(1/2))/b^(7/2)/(-a*d+b*c)^(7/2)-1/3*(A*b^3-a*(B*b^2-C*a*b+D*a^2))*
(d*x+c)^(1/2)/b^3/(-a*d+b*c)/(b*x+a)^3-1/12*(b^3*(-5*A*d+6*B*c)-a*b^2*(B*d+
12*C*c)-13*a^3*d*D+a^2*b*(7*C*d+18*D*c))*(d*x+c)^(1/2)/b^3/(-a*d+b*c)^2/(b*
x+a)^2-1/8*(b^3*(5*A*d^2-6*B*c*d+8*C*c^2)-11*a^3*d^2*D+a^2*b*d*(C*d+30*D*c)
-a*b^2*(-B*d^2+4*C*c*d+24*D*c^2))*(d*x+c)^(1/2)/b^3/(-a*d+b*c)^3/(b*x+a)
```

Rubi [A] (verified)

Time = 0.56 (sec) , antiderivative size = 375, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}} = 0.156$, Rules used = {1635, 911, 1171, 393, 214}

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4\sqrt{c+dx}}\,dx = -\frac{\sqrt{c+dx}(Ab^3-a(a^2D-abC+b^2B))}{3b^3(a+bx)^3(bc-ad)}$$

$$+\frac{\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)(5a^3d^3D+a^2bd^2(Cd-18cD)-ab^2d(-Bd^2-24c^2D+4cCd)+b^3(5Ad^3-6Bcd^2}{8b^{7/2}(bc-ad)^{7/2}}$$

$$-\frac{\sqrt{c+dx}(-11a^3d^2D+a^2bd(30cD+Cd)-ab^2(-Bd^2+24c^2D+4cCd)+b^3(5Ad^2-6Bcd+8c^2C))}{8b^3(a+bx)(bc-ad)^3}$$

$$-\frac{\sqrt{c+dx}(-13a^3dD+a^2b(18cD+7Cd)-ab^2(Bd+12cC)+b^3(6Bc-5Ad))}{12b^3(a+bx)^2(bc-ad)^2}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^4*Sqrt[c + d*x]),x]
```

```
[Out] -1/3*((A*b^3 - a*(b^2*B - a*b*C + a^2*D))*Sqrt[c + d*x])/(b^3*(b*c - a*d)*(
a + b*x)^3) - ((b^3*(6*B*c - 5*A*d) - a*b^2*(12*c*C + B*d) - 13*a^3*d*D + a
^2*b*(7*C*d + 18*c*D))*Sqrt[c + d*x])/(12*b^3*(b*c - a*d)^2*(a + b*x)^2) -
((b^3*(8*c^2*C - 6*B*c*d + 5*A*d^2) - 11*a^3*d^2*D + a^2*b*d*(C*d + 30*c*D)
 - a*b^2*(4*c*C*d - B*d^2 + 24*c^2*D))*Sqrt[c + d*x])/(8*b^3*(b*c - a*d)^3*
(a + b*x)) + ((5*a^3*d^3*D + a^2*b*d^2*(C*d - 18*c*D) - a*b^2*d*(4*c*C*d -
B*d^2 - 24*c^2*D) + b^3*(8*c^2*C*d - 6*B*c*d^2 + 5*A*d^3 - 16*c^3*D))*ArcTa
nh[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[b*c - a*d]])/(8*b^(7/2)*(b*c - a*d)^(7/2))
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 393

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_)*((c_) + (d_.)*(x_)^(n_)), x_Symbol] :> Si
mp[(-(b*c - a*d))*x*((a + b*x^n)^(p + 1)/(a*b*n*(p + 1))), x] - Dist[(a*d -
 b*c*(n*(p + 1) + 1))/(a*b*n*(p + 1)), Int[(a + b*x^n)^(p + 1), x], x] /; F
reeQ[{a, b, c, d, n, p}, x] && NeQ[b*c - a*d, 0] && (LtQ[p, -1] || ILtQ[1/n
 + p, 0])
```

Rule 911

```
Int[((d_.) + (e_.)*(x_))^(m_)*((f_.) + (g_.)*(x_))^(n_)*((a_.) + (b_.)*(x_)
 + (c_.)*(x_)^2)^(p_.), x_Symbol] :> With[{q = Denominator[m]}, Dist[q/e, S
ubst[Int[x^(q*(m + 1) - 1)*((e*f - d*g)/e + g*(x^q/e))^n*((c*d^2 - b*d*e +
a*e^2)/e^2 - (2*c*d - b*e)*(x^q/e^2) + c*(x^(2*q)/e^2))^p, x], x, (d + e*x)
```

```
^(1/q)], x]] /; FreeQ[{a, b, c, d, e, f, g}, x] && NeQ[e*f - d*g, 0] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IntegersQ[n, p] && Fra
ctionQ[m]
```

Rule 1171

```
Int[((d_) + (e_.)*(x_)^2)^(q_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.),
x_Symbol] :> With[{Qx = PolynomialQuotient[(a + b*x^2 + c*x^4)^p, d + e*x^2
, x], R = Coeff[PolynomialRemainder[(a + b*x^2 + c*x^4)^p, d + e*x^2, x], x
, 0]}, Simp[(-R)*x*((d + e*x^2)^(q + 1)/(2*d*(q + 1))), x] + Dist[1/(2*d*(q
 + 1)), Int[(d + e*x^2)^(q + 1)*ExpandToSum[2*d*(q + 1)*Qx + R*(2*q + 3), x
], x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2
 - b*d*e + a*e^2, 0] && IGtQ[p, 0] && LtQ[q, -1]
```

Rule 1635

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol] :
> With[{Qx = PolynomialQuotient[Px, a + b*x, x], R = PolynomialRemainder[Px
, a + b*x, x]}, Simp[R*(a + b*x)^(m + 1)*((c + d*x)^(n + 1)/((m + 1)*(b*c -
 a*d))), x] + Dist[1/((m + 1)*(b*c - a*d)), Int[(a + b*x)^(m + 1)*(c + d*x)
^n*ExpandToSum[(m + 1)*(b*c - a*d)*Qx - d*R*(m + n + 2), x], x], x]] /; Fre
eQ[{a, b, c, d, n}, x] && PolyQ[Px, x] && ILtQ[m, -1] && GtQ[Expon[Px, x],
2]
```

Rubi steps

$$\text{integral} = -\frac{\left(Ab^3-a\left(b^2B-abC+a^2D\right)\right)\sqrt{c+dx}}{3b^3(bc-ad)(a+bx)^3} - \frac{\int \frac{-\frac{b^3(6Bc-5Ad)-ab^2(6cC+Bd)-a^3dD+a^2b(Cd+6cD)}{2b^3}-\frac{3(bc-ad)(bC-aD)x}{b^2}-3\left(c-\frac{ad}{b}\right)Dx^2}{(a+bx)^3\sqrt{c+dx}}\,dx}{3(bc-ad)}$$

$$= -\frac{\left(Ab^3-a\left(b^2B-abC+a^2D\right)\right)\sqrt{c+dx}}{3b^3(bc-ad)(a+bx)^3} - \frac{2\text{Subst}\left(\int \frac{\frac{-3c^2\left(c-\frac{ad}{b}\right)D+\frac{3cd(bc-ad)(bC-aD)}{b^2}-\frac{d^2\left(b^3(6Bc-5Ad)-ab^2(6cC+Bd)-a^3dD+a^2b(Cd+6cD)\right)}{2b^3}}{d^2}-\frac{\left(-6c\left(c-\frac{ad}{b}\right)D+\frac{3d(bc-ad}{b}\right.}{d^2}}{\left(\frac{-bc+ad}{d}+\frac{bx^2}{d}\right)^3}\right.}{3d(bc-ad)}$$

$$= -\frac{(Ab^3 - a(b^2B - abC + a^2D))\sqrt{c+dx}}{3b^3(bc-ad)(a+bx)^3} - \frac{(b^3(6Bc - 5Ad) - ab^2(12cC + Bd) - 13a^3dD + a^2b(7Cd + 18cD))\sqrt{c+dx}}{12b^3(bc-ad)^2(a+bx)^2} - \frac{\text{Subst}\left(\int \frac{\frac{3\left(ab^2d^2(4cC-Bd)+3a^3d^3D-a^2bd^2(Cd+6cD)-b^3\left(8c^2Cd-6Bcd^2+5Ad^3-8c^3D\right)\right)}{2b^3d^2} - \frac{12(bc-ad)^2Dx^2}{b^2d^2}}{\left(\frac{-bc+ad}{d}+\frac{bx^2}{d}\right)^2}dx, x, \sqrt{c+dx}\right)}{6(bc-ad)^2}$$

$$= -\frac{(Ab^3 - a(b^2B - abC + a^2D))\sqrt{c+dx}}{3b^3(bc-ad)(a+bx)^3} - \frac{(b^3(6Bc - 5Ad) - ab^2(12cC + Bd) - 13a^3dD + a^2b(7Cd + 18cD))\sqrt{c+dx}}{12b^3(bc-ad)^2(a+bx)^2} - \frac{(b^3(8c^2C - 6Bcd + 5Ad^2) - 11a^3d^2D + a^2bd(Cd + 30cD) - ab^2(4cCd - Bd^2 + 24c^2D))\sqrt{c+}}{8b^3(bc-ad)^3(a+bx)} - \frac{(5a^3d^3D + a^2bd^2(Cd - 18cD) - ab^2d(4cCd - Bd^2 - 24c^2D) + b^3(8c^2Cd - 6Bcd^2 + 5Ad^3 - 16}{8b^3d(bc-ad)^3}$$

$$= -\frac{(Ab^3 - a(b^2B - abC + a^2D))\sqrt{c+dx}}{3b^3(bc-ad)(a+bx)^3} - \frac{(b^3(6Bc - 5Ad) - ab^2(12cC + Bd) - 13a^3dD + a^2b(7Cd + 18cD))\sqrt{c+dx}}{12b^3(bc-ad)^2(a+bx)^2} - \frac{(b^3(8c^2C - 6Bcd + 5Ad^2) - 11a^3d^2D + a^2bd(Cd + 30cD) - ab^2(4cCd - Bd^2 + 24c^2D))\sqrt{c+}}{8b^3(bc-ad)^3(a+bx)} + \frac{(5a^3d^3D + a^2bd^2(Cd - 18cD) - ab^2d(4cCd - Bd^2 - 24c^2D) + b^3(8c^2Cd - 6Bcd^2 + 5Ad^3 - 16}{8b^{7/2}(bc-ad)^{7/2}}$$

Mathematica [A] (verified)

Time = 1.23 (sec) , antiderivative size = 409, normalized size of antiderivative = 1.09

$$\int \frac{A + Bx + Cx^2 + Dx^3}{(a+bx)^4\sqrt{c+dx}}\,dx = -\frac{\sqrt{c+dx}(-15a^5d^2D + 6b^5cx(2Bc + 4cCx - 3Bdx) + a^4bd(-3Cd + 44cD - 40dDx) + ab^4(-12cx(-2}{} - \frac{(-5a^3d^3D + a^2bd^2(-Cd + 18cD) - ab^2d(-4cCd + Bd^2 + 24c^2D) + b^3(-8c^2Cd + 6Bcd^2 - 5Ad^3 + 16}{8b^{7/2}(-bc+ad)^{7/2}}$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^4*Sqrt[c + d*x]),x]
```

```
[Out] -1/24*(Sqrt[c + d*x]*(-15*a^5*d^2*D + 6*b^5*c*x*(2*B*c + 4*c*C*x - 3*B*d*x)
 + a^4*b*d*(-3*C*d + 44*c*D - 40*d*D*x) + a*b^4*(-12*c*x*(-2*c*C + C*d*x +
```

```
6*c*D*x) + B*(4*c^2 - 50*c*d*x + 3*d^2*x^2)) + A*b^3*(33*a^2*d^2 + 2*a*b*d*
(-13*c + 20*d*x) + b^2*(8*c^2 - 10*c*d*x + 15*d^2*x^2)) - a^3*b^2*(44*c^2*D
 - 2*c*d*(5*C + 59*D*x) + d^2*(3*B + 8*C*x + 33*D*x^2)) + a^2*b^3*(d^2*x*(8
*B + 3*C*x) + 4*c^2*(2*C - 27*D*x) + 2*c*d*(-8*B + 7*C*x + 45*D*x^2))))/(b^
3*(b*c - a*d)^3*(a + b*x)^3) - ((-5*a^3*d^3*D + a^2*b*d^2*(-(C*d) + 18*c*D)
 - a*b^2*d*(-4*c*C*d + B*d^2 + 24*c^2*D) + b^3*(-8*c^2*C*d + 6*B*c*d^2 - 5*
A*d^3 + 16*c^3*D))*ArcTan[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[-(b*c) + a*d]])/(8*b
^(7/2)*(-(b*c) + a*d)^(7/2))
```

Maple [A] (verified)

Time = 1.83 (sec) , antiderivative size = 382, normalized size of antiderivative = 1.02

method	result
pseudoelliptic	$\frac{5\left(\left(A\,d^{3}-\frac{6}{5}B c\,d^{2}+\frac{8}{5}C\,c^{2}d-\frac{16}{5}D c^{3}\right)b^{3}+\frac{a d\left(B\,d^{2}-4C c d+24D c^{2}\right)b^{2}}{5}+\frac{a^{2}b\,d^{2}(C d-18D c)}{5}+a^{3}d^{3}D\right)(b x+a)^{3}\arctan\left(\frac{b\sqrt{d x+c}}{\sqrt{(a d-b c)b}}\right)}{8}+$
derivativedivides	$\frac{d\left(5A\,b^{3}d^{2}+B a\,b^{2}d^{2}-6B\,b^{3}c d+a^{2}b C\,d^{2}-4C a\,b^{2}c d+8C\,b^{3}c^{2}-11a^{3}d^{2}D+30D a^{2}b c d-24D a\,b^{2}c^{2}\right)(d x+c)^{\frac{5}{2}}}{8b\left(a^{3}d^{3}-3a^{2}b c\,d^{2}+3a\,b^{2}c^{2}d-b^{3}c^{3}\right)}+$
default	$\frac{d\left(5A\,b^{3}d^{2}+B a\,b^{2}d^{2}-6B\,b^{3}c d+a^{2}b C\,d^{2}-4C a\,b^{2}c d+8C\,b^{3}c^{2}-11a^{3}d^{2}D+30D a^{2}b c d-24D a\,b^{2}c^{2}\right)(d x+c)^{\frac{5}{2}}}{8b\left(a^{3}d^{3}-3a^{2}b c\,d^{2}+3a\,b^{2}c^{2}d-b^{3}c^{3}\right)}+$

```
[In] int((D*x^3+C*x^2+B*x+A)/(b*x+a)^4/(d*x+c)^(1/2),x,method=_RETURNVERBOSE)
```

```
[Out] 11/8/((a*d-b*c)*b)^(1/2)*(5/11*((A*d^3-6/5*B*c*d^2+8/5*C*c^2*d-16/5*D*c^3)*
b^3+1/5*a*d*(B*d^2-4*C*c*d+24*D*c^2)*b^2+1/5*a^2*b*d^2*(C*d-18*D*c)+a^3*d^3
*D)*(b*x+a)^3*arctan(b*(d*x+c)^(1/2)/((a*d-b*c)*b)^(1/2))+((a*d-b*c)*b)^(1/
2)*(d*x+c)^(1/2)*((5/11*A*d^2*x^2-10/33*x*c*(9/5*B*x+A)*d+8/33*(3*C*x^2+3/2
*B*x+A)*c^2)*b^5-26/33*a*(-20/13*(3/40*B*x+A)*x*d^2+c*(6/13*C*x^2+25/13*B*x
+A)*d-2/13*c^2*(-18*D*x^2+6*C*x+B))*b^4+a^2*((A+8/33*B*x+1/11*C*x^2)*d^2-16
/33*(-45/8*D*x^2-7/8*C*x+B)*c*d+8/33*(-27/2*D*x+C)*c^2)*b^3-1/11*((11*D*x^2
+8/3*C*x+B)*d^2-10/3*c*(59/5*D*x+C)*d+44/3*D*c^2)*a^3*b^2-1/11*a^4*((40/3*D
*x+C)*d-44/3*D*c)*d*b-5/11*D*a^5*d^2))/(b*x+a)^3/(a*d-b*c)^3/b^3
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1216 vs. $2(354) = 708$.

Time = 0.37 (sec) , antiderivative size = 2446, normalized size of antiderivative = 6.52

$$\int \frac{A + Bx + Cx^2 + Dx^3}{(a + bx)^4 \sqrt{c + dx}}\, dx = \text{Too large to display}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^4/(d*x+c)^(1/2),x, algorithm="fricas"
)
```

```
[Out] [1/48*(3*(16*D*a^3*b^3*c^3 - (5*D*a^6 + C*a^5*b + B*a^4*b^2 + 5*A*a^3*b^3)*
d^3 + (16*D*b^6*c^3 - (5*D*a^3*b^3 + C*a^2*b^4 + B*a*b^5 + 5*A*b^6)*d^3 + 2
*(9*D*a^2*b^4*c + (2*C*a*b^5 + 3*B*b^6)*c)*d^2 - 8*(3*D*a*b^5*c^2 + C*b^6*c
^2)*d)*x^3 + 2*(9*D*a^5*b*c + (2*C*a^4*b^2 + 3*B*a^3*b^3)*c)*d^2 + 3*(16*D*
a*b^5*c^3 - (5*D*a^4*b^2 + C*a^3*b^3 + B*a^2*b^4 + 5*A*a*b^5)*d^3 + 2*(9*D*
a^3*b^3*c + (2*C*a^2*b^4 + 3*B*a*b^5)*c)*d^2 - 8*(3*D*a^2*b^4*c^2 + C*a*b^5
*c^2)*d)*x^2 - 8*(3*D*a^4*b^2*c^2 + C*a^3*b^3*c^2)*d + 3*(16*D*a^2*b^4*c^3
- (5*D*a^5*b + C*a^4*b^2 + B*a^3*b^3 + 5*A*a^2*b^4)*d^3 + 2*(9*D*a^4*b^2*c
+ (2*C*a^3*b^3 + 3*B*a^2*b^4)*c)*d^2 - 8*(3*D*a^3*b^3*c^2 + C*a^2*b^4*c^2)*
d)*x)*sqrt(b^2*c - a*b*d)*log((b*d*x + 2*b*c - a*d - 2*sqrt(b^2*c - a*b*d)*
sqrt(d*x + c))/(b*x + a)) + 2*(44*D*a^3*b^4*c^3 - 4*(2*C*a^2*b^5 + B*a*b^6
+ 2*A*b^7)*c^3 - 3*(5*D*a^6*b + C*a^5*b^2 + B*a^4*b^3 - 11*A*a^3*b^4)*d^3 +
 (59*D*a^5*b^2*c + (13*C*a^4*b^3 - 13*B*a^3*b^4 - 59*A*a^2*b^5)*c)*d^2 + 3*
(24*D*a*b^6*c^3 - 8*C*b^7*c^3 - (11*D*a^4*b^3 - C*a^3*b^4 - B*a^2*b^5 - 5*A
*a*b^6)*d^3 + (41*D*a^3*b^4*c - (5*C*a^2*b^5 + 7*B*a*b^6 + 5*A*b^7)*c)*d^2
- 6*(9*D*a^2*b^5*c^2 - (2*C*a*b^6 + B*b^7)*c^2)*d)*x^2 - 2*(44*D*a^4*b^3*c^
2 + (C*a^3*b^4 - 10*B*a^2*b^5 - 17*A*a*b^6)*c^2)*d + 2*(54*D*a^2*b^5*c^3 -
6*(2*C*a*b^6 + B*b^7)*c^3 - 4*(5*D*a^5*b^2 + C*a^4*b^3 - B*a^3*b^4 - 5*A*a^
2*b^5)*d^3 + (79*D*a^4*b^3*c + (11*C*a^3*b^4 - 29*B*a^2*b^5 - 25*A*a*b^6)*c
)*d^2 - (113*D*a^3*b^4*c^2 - (5*C*a^2*b^5 + 31*B*a*b^6 + 5*A*b^7)*c^2)*d)*x
)*sqrt(d*x + c))/(a^3*b^8*c^4 - 4*a^4*b^7*c^3*d + 6*a^5*b^6*c^2*d^2 - 4*a^6
*b^5*c*d^3 + a^7*b^4*d^4 + (b^11*c^4 - 4*a*b^10*c^3*d + 6*a^2*b^9*c^2*d^2 -
 4*a^3*b^8*c*d^3 + a^4*b^7*d^4)*x^3 + 3*(a*b^10*c^4 - 4*a^2*b^9*c^3*d + 6*a
^3*b^8*c^2*d^2 - 4*a^4*b^7*c*d^3 + a^5*b^6*d^4)*x^2 + 3*(a^2*b^9*c^4 - 4*a^
3*b^8*c^3*d + 6*a^4*b^7*c^2*d^2 - 4*a^5*b^6*c*d^3 + a^6*b^5*d^4)*x), 1/24*(
3*(16*D*a^3*b^3*c^3 - (5*D*a^6 + C*a^5*b + B*a^4*b^2 + 5*A*a^3*b^3)*d^3 + (
16*D*b^6*c^3 - (5*D*a^3*b^3 + C*a^2*b^4 + B*a*b^5 + 5*A*b^6)*d^3 + 2*(9*D*a
^2*b^4*c + (2*C*a*b^5 + 3*B*b^6)*c)*d^2 - 8*(3*D*a*b^5*c^2 + C*b^6*c^2)*d)*
x^3 + 2*(9*D*a^5*b*c + (2*C*a^4*b^2 + 3*B*a^3*b^3)*c)*d^2 + 3*(16*D*a*b^5*c
^3 - (5*D*a^4*b^2 + C*a^3*b^3 + B*a^2*b^4 + 5*A*a*b^5)*d^3 + 2*(9*D*a^3*b^3
*c + (2*C*a^2*b^4 + 3*B*a*b^5)*c)*d^2 - 8*(3*D*a^2*b^4*c^2 + C*a*b^5*c^2)*d
)*x^2 - 8*(3*D*a^4*b^2*c^2 + C*a^3*b^3*c^2)*d + 3*(16*D*a^2*b^4*c^3 - (5*D*
a^5*b + C*a^4*b^2 + B*a^3*b^3 + 5*A*a^2*b^4)*d^3 + 2*(9*D*a^4*b^2*c + (2*C*
a^3*b^3 + 3*B*a^2*b^4)*c)*d^2 - 8*(3*D*a^3*b^3*c^2 + C*a^2*b^4*c^2)*d)*x)*s
```

```
qrt(-b^2*c + a*b*d)*arctan(sqrt(-b^2*c + a*b*d)*sqrt(d*x + c)/(b*d*x + b*c)
) + (44*D*a^3*b^4*c^3 - 4*(2*C*a^2*b^5 + B*a*b^6 + 2*A*b^7)*c^3 - 3*(5*D*a^
6*b + C*a^5*b^2 + B*a^4*b^3 - 11*A*a^3*b^4)*d^3 + (59*D*a^5*b^2*c + (13*C*a
^4*b^3 - 13*B*a^3*b^4 - 59*A*a^2*b^5)*c)*d^2 + 3*(24*D*a*b^6*c^3 - 8*C*b^7*
c^3 - (11*D*a^4*b^3 - C*a^3*b^4 - B*a^2*b^5 - 5*A*a*b^6)*d^3 + (41*D*a^3*b^
4*c - (5*C*a^2*b^5 + 7*B*a*b^6 + 5*A*b^7)*c)*d^2 - 6*(9*D*a^2*b^5*c^2 - (2*
C*a*b^6 + B*b^7)*c^2)*d)*x^2 - 2*(44*D*a^4*b^3*c^2 + (C*a^3*b^4 - 10*B*a^2*
b^5 - 17*A*a*b^6)*c^2)*d + 2*(54*D*a^2*b^5*c^3 - 6*(2*C*a*b^6 + B*b^7)*c^3
- 4*(5*D*a^5*b^2 + C*a^4*b^3 - B*a^3*b^4 - 5*A*a^2*b^5)*d^3 + (79*D*a^4*b^3
*c + (11*C*a^3*b^4 - 29*B*a^2*b^5 - 25*A*a*b^6)*c)*d^2 - (113*D*a^3*b^4*c^2
 - (5*C*a^2*b^5 + 31*B*a*b^6 + 5*A*b^7)*c^2)*d)*x)*sqrt(d*x + c))/(a^3*b^8*
c^4 - 4*a^4*b^7*c^3*d + 6*a^5*b^6*c^2*d^2 - 4*a^6*b^5*c*d^3 + a^7*b^4*d^4 +
 (b^11*c^4 - 4*a*b^10*c^3*d + 6*a^2*b^9*c^2*d^2 - 4*a^3*b^8*c*d^3 + a^4*b^7
*d^4)*x^3 + 3*(a*b^10*c^4 - 4*a^2*b^9*c^3*d + 6*a^3*b^8*c^2*d^2 - 4*a^4*b^7
*c*d^3 + a^5*b^6*d^4)*x^2 + 3*(a^2*b^9*c^4 - 4*a^3*b^8*c^3*d + 6*a^4*b^7*c^
2*d^2 - 4*a^5*b^6*c*d^3 + a^6*b^5*d^4)*x)]
```

Sympy [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4\sqrt{c+dx}}\,dx = \text{Timed out}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(b*x+a)**4/(d*x+c)**(1/2),x)
```

```
[Out] Timed out
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4\sqrt{c+dx}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^4/(d*x+c)^(1/2),x, algorithm="maxima"
)
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 976 vs. $2(354) = 708$.

Time = 0.31 (sec) , antiderivative size = 976, normalized size of antiderivative = 2.60

$$\int \frac{A + Bx + Cx^2 + Dx^3}{(a + bx)^4\sqrt{c + dx}}\,dx$$

$$= \frac{(16\,Db^3c^3 - 24\,Dab^2c^2d - 8\,Cb^3c^2d + 18\,Da^2bcd^2 + 4\,Cab^2cd^2 + 6\,Bb^3cd^2 - 5\,Da^3d^3 - Ca^2bd^3 - Bab^2d^3}{8\,(b^6c^3 - 3\,ab^5c^2d + 3\,a^2b^4cd^2 - a^3b^3d^3)\sqrt{-b^2c + abd}}$$

$$+ \frac{72\,(dx + c)^{\frac{5}{2}}Dab^4c^2d - 24\,(dx + c)^{\frac{5}{2}}Cb^5c^2d - 144\,(dx + c)^{\frac{3}{2}}Dab^4c^3d + 48\,(dx + c)^{\frac{3}{2}}Cb^5c^3d + 72\,\sqrt{dx +}}{}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^4/(d*x+c)^(1/2),x, algorithm="giac")
```

```
[Out] 1/8*(16*D*b^3*c^3 - 24*D*a*b^2*c^2*d - 8*C*b^3*c^2*d + 18*D*a^2*b*c*d^2 + 4
*C*a*b^2*c*d^2 + 6*B*b^3*c*d^2 - 5*D*a^3*d^3 - C*a^2*b*d^3 - B*a*b^2*d^3 -
5*A*b^3*d^3)*arctan(sqrt(d*x + c)*b/sqrt(-b^2*c + a*b*d))/((b^6*c^3 - 3*a*b
^5*c^2*d + 3*a^2*b^4*c*d^2 - a^3*b^3*d^3)*sqrt(-b^2*c + a*b*d)) + 1/24*(72*
(d*x + c)^(5/2)*D*a*b^4*c^2*d - 24*(d*x + c)^(5/2)*C*b^5*c^2*d - 144*(d*x +
 c)^(3/2)*D*a*b^4*c^3*d + 48*(d*x + c)^(3/2)*C*b^5*c^3*d + 72*sqrt(d*x + c)
*D*a*b^4*c^4*d - 24*sqrt(d*x + c)*C*b^5*c^4*d - 90*(d*x + c)^(5/2)*D*a^2*b^
3*c*d^2 + 12*(d*x + c)^(5/2)*C*a*b^4*c*d^2 + 18*(d*x + c)^(5/2)*B*b^5*c*d^2
 + 288*(d*x + c)^(3/2)*D*a^2*b^3*c^2*d^2 - 48*(d*x + c)^(3/2)*C*a*b^4*c^2*d
^2 - 48*(d*x + c)^(3/2)*B*b^5*c^2*d^2 - 198*sqrt(d*x + c)*D*a^2*b^3*c^3*d^2
 + 36*sqrt(d*x + c)*C*a*b^4*c^3*d^2 + 30*sqrt(d*x + c)*B*b^5*c^3*d^2 + 33*(
d*x + c)^(5/2)*D*a^3*b^2*d^3 - 3*(d*x + c)^(5/2)*C*a^2*b^3*d^3 - 3*(d*x + c
)^(5/2)*B*a*b^4*d^3 - 15*(d*x + c)^(5/2)*A*b^5*d^3 - 184*(d*x + c)^(3/2)*D*
a^3*b^2*c*d^3 - 8*(d*x + c)^(3/2)*C*a^2*b^3*c*d^3 + 56*(d*x + c)^(3/2)*B*a*
b^4*c*d^3 + 40*(d*x + c)^(3/2)*A*b^5*c*d^3 + 195*sqrt(d*x + c)*D*a^3*b^2*c^
2*d^3 + 3*sqrt(d*x + c)*C*a^2*b^3*c^2*d^3 - 57*sqrt(d*x + c)*B*a*b^4*c^2*d^
3 - 33*sqrt(d*x + c)*A*b^5*c^2*d^3 + 40*(d*x + c)^(3/2)*D*a^4*b*d^4 + 8*(d*
x + c)^(3/2)*C*a^3*b^2*d^4 - 8*(d*x + c)^(3/2)*B*a^2*b^3*d^4 - 40*(d*x + c)
^(3/2)*A*a*b^4*d^4 - 84*sqrt(d*x + c)*D*a^4*b*c*d^4 - 18*sqrt(d*x + c)*C*a^
3*b^2*c*d^4 + 24*sqrt(d*x + c)*B*a^2*b^3*c*d^4 + 66*sqrt(d*x + c)*A*a*b^4*c
*d^4 + 15*sqrt(d*x + c)*D*a^5*d^5 + 3*sqrt(d*x + c)*C*a^4*b*d^5 + 3*sqrt(d*
x + c)*B*a^3*b^2*d^5 - 33*sqrt(d*x + c)*A*a^2*b^3*d^5)/((b^6*c^3 - 3*a*b^5*
c^2*d + 3*a^2*b^4*c*d^2 - a^3*b^3*d^3)*((d*x + c)*b - b*c + a*d)^3)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4\sqrt{c+dx}}\,dx=\int \frac{A+B\,x+C\,x^2+x^3\,D}{{\left(a+b\,x\right)}^4\,\sqrt{c+d\,x}}\,dx$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^4*(c + d*x)^(1/2)),x)
[Out] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^4*(c + d*x)^(1/2)), x)
```

3.9 $\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^5\sqrt{c+dx}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 495

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^5\sqrt{c+dx}}\,dx = -\frac{(Ab^3-a(b^2B-abC+a^2D))\sqrt{c+dx}}{4b^3(bc-ad)(a+bx)^4}$$

$$-\frac{(b^3(8Bc-7Ad)-ab^2(16cC+Bd)-17a^3dD+3a^2b(3Cd+8cD))\sqrt{c+dx}}{24b^3(bc-ad)^2(a+bx)^3}$$

$$-\frac{(b^3(48c^2C-40Bcd+35Ad^2)-59a^3d^2D+3a^2bd(Cd+56cD)-ab^2(16cCd-5Bd^2+144c^2D))\sqrt{c+}}{96b^3(bc-ad)^3(a+bx)^2}$$

$$+\frac{(5a^3d^3D+3a^2bd^2(Cd-8cD)-ab^2d(16cCd-5Bd^2-48c^2D)+b^3(48c^2Cd-40Bcd^2+35Ad^3-64c}{64b^3(bc-ad)^4(a+bx)}$$

$$-\frac{d(5a^3d^3D+3a^2bd^2(Cd-8cD)-ab^2d(16cCd-5Bd^2-48c^2D)+b^3(48c^2Cd-40Bcd^2+35Ad^3-64}{64b^{7/2}(bc-ad)^{9/2}}$$

```
[Out] -1/64*d*(5*a^3*d^3*D+3*a^2*b*d^2*(C*d-8*D*c)-a*b^2*d*(-5*B*d^2+16*C*c*d-48*
D*c^2)+b^3*(35*A*d^3-40*B*c*d^2+48*C*c^2*d-64*D*c^3))*arctanh(b^(1/2)*(d*x+
c)^(1/2)/(-a*d+b*c)^(1/2))/b^(7/2)/(-a*d+b*c)^(9/2)-1/4*(A*b^3-a*(B*b^2-C*a
*b+D*a^2))*(d*x+c)^(1/2)/b^3/(-a*d+b*c)/(b*x+a)^4-1/24*(b^3*(-7*A*d+8*B*c)-
a*b^2*(B*d+16*C*c)-17*a^3*d*D+3*a^2*b*(3*C*d+8*D*c))*(d*x+c)^(1/2)/b^3/(-a*
d+b*c)^2/(b*x+a)^3-1/96*(b^3*(35*A*d^2-40*B*c*d+48*C*c^2)-59*a^3*d^2*D+3*a^
2*b*d*(C*d+56*D*c)-a*b^2*(-5*B*d^2+16*C*c*d+144*D*c^2))*(d*x+c)^(1/2)/b^3/(
-a*d+b*c)^3/(b*x+a)^2+1/64*(5*a^3*d^3*D+3*a^2*b*d^2*(C*d-8*D*c)-a*b^2*d*(-5
*B*d^2+16*C*c*d-48*D*c^2)+b^3*(35*A*d^3-40*B*c*d^2+48*C*c^2*d-64*D*c^3))*(d
*x+c)^(1/2)/b^3/(-a*d+b*c)^4/(b*x+a)
```

Rubi [A] (verified)

Time = 0.68 (sec) , antiderivative size = 495, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.188$, Rules used = {1635, 911, 1171, 393, 205, 214}

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^5\sqrt{c+dx}}\,dx = -\frac{\sqrt{c+dx}(Ab^3-a(a^2D-abC+b^2B))}{4b^3(a+bx)^4(bc-ad)}$$

$$-\frac{d\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)(5a^3d^3D+3a^2bd^2(Cd-8cD)-ab^2d(-5Bd^2-48c^2D+16cCd)+b^3(35Ad^3-40B}{64b^{7/2}(bc-ad)^{9/2}}$$

$$-\frac{\sqrt{c+dx}(-59a^3d^2D+3a^2bd(56cD+Cd)-ab^2(-5Bd^2+144c^2D+16cCd)+b^3(35Ad^2-40Bcd+48c}{96b^3(a+bx)^2(bc-ad)^3}$$

$$+\frac{\sqrt{c+dx}(5a^3d^3D+3a^2bd^2(Cd-8cD)-ab^2d(-5Bd^2-48c^2D+16cCd)+b^3(35Ad^3-40Bcd^2-64c^3}{64b^3(a+bx)(bc-ad)^4}$$

$$-\frac{\sqrt{c+dx}(-17a^3dD+3a^2b(8cD+3Cd)-ab^2(Bd+16cC)+b^3(8Bc-7Ad))}{24b^3(a+bx)^3(bc-ad)^2}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^5*Sqrt[c + d*x]),x]
```

```
[Out] -1/4*((A*b^3 - a*(b^2*B - a*b*C + a^2*D))*Sqrt[c + d*x])/(b^3*(b*c - a*d)*(
a + b*x)^4) - ((b^3*(8*B*c - 7*A*d) - a*b^2*(16*c*C + B*d) - 17*a^3*d*D + 3
*a^2*b*(3*C*d + 8*c*D))*Sqrt[c + d*x])/(24*b^3*(b*c - a*d)^2*(a + b*x)^3) -
 ((b^3*(48*c^2*C - 40*B*c*d + 35*A*d^2) - 59*a^3*d^2*D + 3*a^2*b*d*(C*d + 5
6*c*D) - a*b^2*(16*c*C*d - 5*B*d^2 + 144*c^2*D))*Sqrt[c + d*x])/(96*b^3*(b*
c - a*d)^3*(a + b*x)^2) + ((5*a^3*d^3*D + 3*a^2*b*d^2*(C*d - 8*c*D) - a*b^2
*d*(16*c*C*d - 5*B*d^2 - 48*c^2*D) + b^3*(48*c^2*C*d - 40*B*c*d^2 + 35*A*d^
3 - 64*c^3*D))*Sqrt[c + d*x])/(64*b^3*(b*c - a*d)^4*(a + b*x)) - (d*(5*a^3*
d^3*D + 3*a^2*b*d^2*(C*d - 8*c*D) - a*b^2*d*(16*c*C*d - 5*B*d^2 - 48*c^2*D)
 + b^3*(48*c^2*C*d - 40*B*c*d^2 + 35*A*d^3 - 64*c^3*D))*ArcTanh[(Sqrt[b]*Sq
rt[c + d*x])/Sqrt[b*c - a*d]])/(64*b^(7/2)*(b*c - a*d)^(9/2))
```

Rule 205

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[(-x)*((a + b*x^n)^(p +
 1)/(a*n*(p + 1))), x] + Dist[(n*(p + 1) + 1)/(a*n*(p + 1)), Int[(a + b*x^n
)^(p + 1), x], x] /; FreeQ[{a, b}, x] && IGtQ[n, 0] && LtQ[p, -1] && (Integ
erQ[2*p] || (n == 2 && IntegerQ[4*p]) || (n == 2 && IntegerQ[3*p]) || Denom
inator[p + 1/n] < Denominator[p])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 393

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_)*((c_) + (d_.)*(x_)^(n_)), x_Symbol] :> Si
mp[(-(b*c - a*d))*x*((a + b*x^n)^(p + 1)/(a*b*n*(p + 1))), x] - Dist[(a*d -
 b*c*(n*(p + 1) + 1))/(a*b*n*(p + 1)), Int[(a + b*x^n)^(p + 1), x], x] /; F
reeQ[{a, b, c, d, n, p}, x] && NeQ[b*c - a*d, 0] && (LtQ[p, -1] || ILtQ[1/n
 + p, 0])
```

Rule 911

```
Int[((d_.) + (e_.)*(x_))^(m_)*((f_.) + (g_.)*(x_))^(n_)*((a_.) + (b_.)*(x_)
 + (c_.)*(x_)^2)^(p_.), x_Symbol] :> With[{q = Denominator[m]}, Dist[q/e, S
ubst[Int[x^(q*(m + 1) - 1)*((e*f - d*g)/e + g*(x^q/e))^n*((c*d^2 - b*d*e +
a*e^2)/e^2 - (2*c*d - b*e)*(x^q/e^2) + c*(x^(2*q)/e^2))^p, x], x, (d + e*x)
^(1/q)], x]] /; FreeQ[{a, b, c, d, e, f, g}, x] && NeQ[e*f - d*g, 0] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IntegersQ[n, p] && Fra
ctionQ[m]
```

Rule 1171

```
Int[((d_) + (e_.)*(x_)^2)^(q_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.),
x_Symbol] :> With[{Qx = PolynomialQuotient[(a + b*x^2 + c*x^4)^p, d + e*x^2
, x], R = Coeff[PolynomialRemainder[(a + b*x^2 + c*x^4)^p, d + e*x^2, x], x
, 0]}, Simp[(-R)*x*((d + e*x^2)^(q + 1)/(2*d*(q + 1))), x] + Dist[1/(2*d*(q
 + 1)), Int[(d + e*x^2)^(q + 1)*ExpandToSum[2*d*(q + 1)*Qx + R*(2*q + 3), x
], x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2
 - b*d*e + a*e^2, 0] && IGtQ[p, 0] && LtQ[q, -1]
```

Rule 1635

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol] :
> With[{Qx = PolynomialQuotient[Px, a + b*x, x], R = PolynomialRemainder[Px
, a + b*x, x]}, Simp[R*(a + b*x)^(m + 1)*((c + d*x)^(n + 1)/((m + 1)*(b*c -
 a*d))), x] + Dist[1/((m + 1)*(b*c - a*d)), Int[(a + b*x)^(m + 1)*(c + d*x)
^n*ExpandToSum[(m + 1)*(b*c - a*d)*Qx - d*R*(m + n + 2), x], x], x]] /; Fre
eQ[{a, b, c, d, n}, x] && PolyQ[Px, x] && ILtQ[m, -1] && GtQ[Expon[Px, x],
2]
```

Rubi steps

$$\begin{aligned}\text{integral} &= -\frac{(Ab^3-a(b^2B-abC+a^2D))\sqrt{c+dx}}{4b^3(bc-ad)(a+bx)^4}\\ &\quad-\frac{\int\frac{-\frac{b^3(8Bc-7Ad)-ab^2(8cC+Bd)-a^3dD+a^2b(Cd+8cD)}{2b^3}-\frac{4(bc-ad)(bC-aD)x}{b^2}-4\left(c-\frac{ad}{b}\right)Dx^2}{(a+bx)^4\sqrt{c+dx}}\,dx}{4(bc-ad)}\end{aligned}$$

$$= -\frac{(Ab^3 - a(b^2B - abC + a^2D))\sqrt{c+dx}}{4b^3(bc-ad)(a+bx)^4} - \frac{\text{Subst}\left(\int \frac{\frac{-4c^2\left(c-\frac{ad}{b}\right)D+\frac{4cd(bc-ad)(bC-aD)}{b^2}-\frac{d^2\left(b^3(8Bc-7Ad)-ab^2(8cC+Bd)-a^3dD+a^2b(Cd+8cD)\right)}{2b^3}}{d^2}-\frac{\left(-8c\left(c-\frac{ad}{b}\right)D+\frac{4d(bc-ad)(}{b^2}}{d^2}}{\left(\frac{-bc+ad}{d}+\frac{bx^2}{d}\right)^4}\right.}{2d(bc-ad)}$$

$$= -\frac{(Ab^3 - a(b^2B - abC + a^2D))\sqrt{c+dx}}{4b^3(bc-ad)(a+bx)^4} - \frac{(b^3(8Bc-7Ad) - ab^2(16cC+Bd) - 17a^3dD + 3a^2b(3Cd+8cD))\sqrt{c+dx}}{24b^3(bc-ad)^2(a+bx)^3} - \frac{\text{Subst}\left(\int \frac{\frac{1}{2}\left(40Bc-\frac{48c^2C}{d}-35Ad+\frac{a(16cC-5Bd)}{b}+\frac{48c^3D}{d^2}+\frac{11a^3dD}{b^3}-\frac{3a^2(Cd+8cD)}{b^2}\right)-\frac{24(bc-ad)^2Dx^2}{b^2d^2}}{\left(\frac{-bc+ad}{d}+\frac{bx^2}{d}\right)^3}dx, x, \sqrt{c+dx}\right)}{12(bc-ad)^2}$$

$$= -\frac{(Ab^3 - a(b^2B - abC + a^2D))\sqrt{c+dx}}{4b^3(bc-ad)(a+bx)^4} - \frac{(b^3(8Bc-7Ad) - ab^2(16cC+Bd) - 17a^3dD + 3a^2b(3Cd+8cD))\sqrt{c+dx}}{24b^3(bc-ad)^2(a+bx)^3} - \frac{(b^3(48c^2C - 40Bcd + 35Ad^2) - 59a^3d^2D + 3a^2bd(Cd+56cD) - ab^2(16cCd - 5Bd^2 + 144c^2D)}{96b^3(bc-ad)^3(a+bx)^2} - \frac{(5a^3d^3D + 3a^2bd^2(Cd-8cD) - ab^2d(16cCd - 5Bd^2 - 48c^2D) + b^3(48c^2Cd - 40Bcd^2 + 35Ad^3}{32b^3d(bc-ad)^3}$$

$$= -\frac{(Ab^3 - a(b^2B - abC + a^2D))\sqrt{c+dx}}{4b^3(bc-ad)(a+bx)^4} - \frac{(b^3(8Bc-7Ad) - ab^2(16cC+Bd) - 17a^3dD + 3a^2b(3Cd+8cD))\sqrt{c+dx}}{24b^3(bc-ad)^2(a+bx)^3} - \frac{(b^3(48c^2C - 40Bcd + 35Ad^2) - 59a^3d^2D + 3a^2bd(Cd+56cD) - ab^2(16cCd - 5Bd^2 + 144c^2D)}{96b^3(bc-ad)^3(a+bx)^2} + \frac{(5a^3d^3D + 3a^2bd^2(Cd-8cD) - ab^2d(16cCd - 5Bd^2 - 48c^2D) + b^3(48c^2Cd - 40Bcd^2 + 35Ad^3}{64b^3(bc-ad)^4(a+bx)} + \frac{(5a^3d^3D + 3a^2bd^2(Cd-8cD) - ab^2d(16cCd - 5Bd^2 - 48c^2D) + b^3(48c^2Cd - 40Bcd^2 + 35Ad^3}{64b^3(bc-ad)^4}$$

$$= -\frac{(Ab^3 - a(b^2B - abC + a^2D))\sqrt{c+dx}}{4b^3(bc-ad)(a+bx)^4}$$

$$-\frac{(b^3(8Bc-7Ad) - ab^2(16cC+Bd) - 17a^3dD + 3a^2b(3Cd+8cD))\sqrt{c+dx}}{24b^3(bc-ad)^2(a+bx)^3}$$

$$-\frac{(b^3(48c^2C - 40Bcd + 35Ad^2) - 59a^3d^2D + 3a^2bd(Cd+56cD) - ab^2(16cCd - 5Bd^2 + 144c^2D}{96b^3(bc-ad)^3(a+bx)^2}$$

$$+\frac{(5a^3d^3D + 3a^2bd^2(Cd-8cD) - ab^2d(16cCd - 5Bd^2 - 48c^2D) + b^3(48c^2Cd - 40Bcd^2 + 35Ad}{64b^3(bc-ad)^4(a+bx)}$$

$$-\frac{d(5a^3d^3D + 3a^2bd^2(Cd-8cD) - ab^2d(16cCd - 5Bd^2 - 48c^2D) + b^3(48c^2Cd - 40Bcd^2 + 35A}{64b^{7/2}(bc-ad)^{9/2}}$$

Mathematica [A] (verified)

Time = 3.20 (sec) , antiderivative size = 602, normalized size of antiderivative = 1.22

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^5\sqrt{c+dx}}\,dx =$$

$$-\frac{\sqrt{c+dx}(15a^6d^3D + a^5bd^2(9Cd - 62cD + 55dDx) + 8b^6cx(6cx(2cC - 3Cdx + 4cDx) + B(8c^2 - 10c}{}$$

$$+\frac{d(5a^3d^3D + 3a^2bd^2(Cd-8cD) + ab^2d(-16cCd + 5Bd^2 + 48c^2D) + b^3(48c^2Cd - 40Bcd^2 + 35Ad^3 - 6}{64b^{7/2}(-bc+ad)^{9/2}}$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^5*Sqrt[c + d*x]),x]
```

```
[Out] -1/192*(Sqrt[c + d*x]*(15*a^6*d^3*D + a^5*b*d^2*(9*C*d - 62*c*D + 55*d*D*x)
 + 8*b^6*c*x*(6*c*x*(2*c*C - 3*C*d*x + 4*c*D*x) + B*(8*c^2 - 10*c*d*x + 15*
d^2*x^2)) + a^3*b^3*(48*c^3*D + c^2*(-88*C*d + 296*d*D*x) + 2*c*d^2*(73*B -
 26*C*x - 119*D*x^2) - d^3*x*(73*B + 33*C*x + 15*D*x^2)) + a^2*b^4*(-(d^3*x
^2*(55*B + 9*C*x)) + 16*c^3*(C + 12*D*x) - 24*c^2*d*(3*B + 15*C*x - 8*D*x^2
) + 2*c*d^2*x*(310*B + 91*C*x + 36*D*x^2)) + a^4*b^2*d*(104*c^2*D - 6*c*d*(
7*C + 38*D*x) + d^2*(15*B + 33*C*x + 73*D*x^2)) + A*b^3*(-279*a^3*d^3 + a^2
*b*d^2*(326*c - 511*d*x) + a*b^2*d*(-200*c^2 + 252*c*d*x - 385*d^2*x^2) + b
^3*(48*c^3 - 56*c^2*d*x + 70*c*d^2*x^2 - 105*d^3*x^3)) + a*b^5*(B*(16*c^3 -
 296*c^2*d*x + 450*c*d^2*x^2 - 15*d^3*x^3) + 16*c*x*(3*C*d^2*x^2 + 2*c^2*(2
*C + 9*D*x) - c*d*x*(35*C + 9*D*x))))))/(b^3*(b*c - a*d)^4*(a + b*x)^4) + (d
*(5*a^3*d^3*D + 3*a^2*b*d^2*(C*d - 8*c*D) + a*b^2*d*(-16*c*C*d + 5*B*d^2 +
48*c^2*D) + b^3*(48*c^2*C*d - 40*B*c*d^2 + 35*A*d^3 - 64*c^3*D))*ArcTan[(Sq
rt[b]*Sqrt[c + d*x])/Sqrt[-(b*c) + a*d]])/(64*b^(7/2)*(-(b*c) + a*d)^(9/2))
```

Maple [A] (verified)

Time = 1.92 (sec) , antiderivative size = 540, normalized size of antiderivative = 1.09

method	result
pseudoelliptic	$\frac{35(bx+a)^4 \left(\left(A d^3-\frac{8}{7} Bc d^2+\frac{48}{35} C c^2 d-\frac{64}{35} Dc^3\right) b^3+\frac{a\left(B d^2-\frac{16}{5} Ccd+\frac{48}{5} Dc^2\right) d b^2}{7}+\frac{3a^2 b d^2 (Cd-8Dc)}{35}+\frac{a^3 d^3 D}{7}\right) d \arctan\left(\frac{b\sqrt{dx+c}}{\sqrt{(ad-bc}}\right.}{64}$
derivativedivides	$2d \left(\frac{\left(35A b^3 d^3+5Ba b^2 d^3-40B b^3 c d^2+3a^2 bC d^3-16Ca b^2 c d^2+48C b^3 c^2 d+5a^3 d^3 D-24Da^2 bc d^2+48Da b^2 c^2 d-64Db^3 c^3\right)(dx+c)^{\frac{7}{2}}}{128a^4 d^4-512a^3 bc d^3+768a^2 b^2 c^2 d^2-512a b^3 c^3 d+128b^4 c^4} \right.$
default	$2d \left(\frac{\left(35A b^3 d^3+5Ba b^2 d^3-40B b^3 c d^2+3a^2 bC d^3-16Ca b^2 c d^2+48C b^3 c^2 d+5a^3 d^3 D-24Da^2 bc d^2+48Da b^2 c^2 d-64Db^3 c^3\right)(dx+c)^{\frac{7}{2}}}{128a^4 d^4-512a^3 bc d^3+768a^2 b^2 c^2 d^2-512a b^3 c^3 d+128b^4 c^4} \right.$

```
[In] int((D*x^3+C*x^2+B*x+A)/(b*x+a)^5/(d*x+c)^(1/2),x,method=_RETURNVERBOSE)
```

```
[Out] 93/64*(35/93*(b*x+a)^4*((A*d^3-8/7*B*c*d^2+48/35*C*c^2*d-64/35*D*c^3)*b^3+1
/7*a*(B*d^2-16/5*C*c*d+48/5*D*c^2)*d*b^2+3/35*a^2*b*d^2*(C*d-8*D*c)+1/7*a^3
*d^3*D)*d*arctan(b*(d*x+c)^(1/2)/((a*d-b*c)*b)^(1/2))+((a*d-b*c)*b)^(1/2)*(
1/93*(35*A*d^3*x^3-70/3*(12/7*B*x+A)*x^2*c*d^2+56/3*x*c^2*(18/7*C*x^2+10/7*
B*x+A)*d-16*c^3*(4*D*x^3+2*C*x^2+4/3*B*x+A))*b^6+200/279*a*(77/40*x^2*(3/77
*B*x+A)*d^3-63/50*(4/21*C*x^2+25/14*B*x+A)*x*c*d^2+c^2*(18/25*D*x^3+14/5*C*
x^2+37/25*B*x+A)*d-2/25*c^3*(18*D*x^2+4*C*x+B))*b^5-326/279*a^2*(-511/326*x
*(9/511*C*x^2+55/511*B*x+A)*d^3+c*(36/163*D*x^3+91/163*C*x^2+310/163*B*x+A)
*d^2-36/163*(-8/3*D*x^2+5*C*x+B)*c^2*d+8/163*c^3*(12*D*x+C))*b^4+a^3*((5/93
*D*x^3+11/93*C*x^2+73/279*B*x+A)*d^3-146/279*c*(-119/73*D*x^2-26/73*C*x+B)*
d^2+88/279*c^2*(-37/11*D*x+C)*d-16/93*D*c^3)*b^3-5/93*a^4*((73/15*D*x^2+11/
5*C*x+B)*d^2-14/5*c*(38/7*D*x+C)*d+104/15*D*c^2)*d*b^2-1/31*a^5*((55/9*D*x+
C)*d-62/9*D*c)*d^2*b-5/93*D*a^6*d^3)*(d*x+c)^(1/2))/((a*d-b*c)*b)^(1/2)/(b*
x+a)^4/(a*d-b*c)^4/b^3
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1805 vs. 2(469) = 938.

Time = 0.49 (sec) , antiderivative size = 3624, normalized size of antiderivative = 7.32

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^5\sqrt{c+dx}}\,dx = \text{Too large to display}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^5/(d*x+c)^(1/2),x, algorithm="fricas"
)
```

```
[Out] [-1/384*(3*(64*D*a^4*b^3*c^3*d - (5*D*a^7 + 3*C*a^6*b + 5*B*a^5*b^2 + 35*A*
a^4*b^3)*d^4 + (64*D*b^7*c^3*d - (5*D*a^3*b^4 + 3*C*a^2*b^5 + 5*B*a*b^6 + 3
5*A*b^7)*d^4 + 8*(3*D*a^2*b^5*c + (2*C*a*b^6 + 5*B*b^7)*c)*d^3 - 48*(D*a*b^
6*c^2 + C*b^7*c^2)*d^2)*x^4 + 8*(3*D*a^6*b*c + (2*C*a^5*b^2 + 5*B*a^4*b^3)*
c)*d^3 + 4*(64*D*a*b^6*c^3*d - (5*D*a^4*b^3 + 3*C*a^3*b^4 + 5*B*a^2*b^5 + 3
5*A*a*b^6)*d^4 + 8*(3*D*a^3*b^4*c + (2*C*a^2*b^5 + 5*B*a*b^6)*c)*d^3 - 48*(
D*a^2*b^5*c^2 + C*a*b^6*c^2)*d^2)*x^3 - 48*(D*a^5*b^2*c^2 + C*a^4*b^3*c^2)*
d^2 + 6*(64*D*a^2*b^5*c^3*d - (5*D*a^5*b^2 + 3*C*a^4*b^3 + 5*B*a^3*b^4 + 35
*A*a^2*b^5)*d^4 + 8*(3*D*a^4*b^3*c + (2*C*a^3*b^4 + 5*B*a^2*b^5)*c)*d^3 - 4
8*(D*a^3*b^4*c^2 + C*a^2*b^5*c^2)*d^2)*x^2 + 4*(64*D*a^3*b^4*c^3*d - (5*D*a
^6*b + 3*C*a^5*b^2 + 5*B*a^4*b^3 + 35*A*a^3*b^4)*d^4 + 8*(3*D*a^5*b^2*c + (
2*C*a^4*b^3 + 5*B*a^3*b^4)*c)*d^3 - 48*(D*a^4*b^3*c^2 + C*a^3*b^4*c^2)*d^2)
*x)*sqrt(b^2*c - a*b*d)*log((b*d*x + 2*b*c - a*d - 2*sqrt(b^2*c - a*b*d)*sq
rt(d*x + c))/(b*x + a)) + 2*(48*D*a^3*b^5*c^4 + 16*(C*a^2*b^6 + B*a*b^7 + 3
*A*b^8)*c^4 - 3*(5*D*a^7*b + 3*C*a^6*b^2 + 5*B*a^5*b^3 - 93*A*a^4*b^4)*d^4
+ (77*D*a^6*b^2*c + (51*C*a^5*b^3 - 131*B*a^4*b^4 - 605*A*a^3*b^5)*c)*d^3 +
 3*(64*D*b^8*c^4 + (5*D*a^4*b^4 + 3*C*a^3*b^5 + 5*B*a^2*b^6 + 35*A*a*b^7)*d
^4 - (29*D*a^3*b^5*c + (19*C*a^2*b^6 + 45*B*a*b^7 + 35*A*b^8)*c)*d^3 + 8*(9
*D*a^2*b^6*c^2 + (8*C*a*b^7 + 5*B*b^8)*c^2)*d^2 - 16*(7*D*a*b^7*c^3 + 3*C*b
^8*c^3)*d)*x^3 - 2*(83*D*a^5*b^3*c^2 - (23*C*a^4*b^4 + 109*B*a^3*b^5 + 263*
A*a^2*b^6)*c^2)*d^2 + (288*D*a*b^7*c^4 + 96*C*b^8*c^4 - (73*D*a^5*b^3 - 33*
C*a^4*b^4 - 55*B*a^3*b^5 - 385*A*a^2*b^6)*d^4 + (311*D*a^4*b^4*c - 5*(43*C*
a^3*b^5 + 101*B*a^2*b^6 + 91*A*a*b^7)*c)*d^3 - 2*(215*D*a^3*b^5*c^2 - (371*
C*a^2*b^6 + 265*B*a*b^7 + 35*A*b^8)*c^2)*d^2 - 16*(6*D*a^2*b^6*c^3 + (41*C*
a*b^7 + 5*B*b^8)*c^3)*d)*x^2 + 8*(7*D*a^4*b^4*c^3 - (13*C*a^3*b^5 + 11*B*a^
2*b^6 + 31*A*a*b^7)*c^3)*d + (192*D*a^2*b^6*c^4 + 64*(C*a*b^7 + B*b^8)*c^4
- (55*D*a^6*b^2 + 33*C*a^5*b^3 - 73*B*a^4*b^4 - 511*A*a^3*b^5)*d^4 + (283*D
*a^5*b^3*c + (85*C*a^4*b^4 - 693*B*a^3*b^5 - 763*A*a^2*b^6)*c)*d^3 - 4*(131
*D*a^4*b^4*c^2 - (77*C*a^3*b^5 + 229*B*a^2*b^6 + 77*A*a*b^7)*c^2)*d^2 + 8*(
13*D*a^3*b^5*c^3 - (53*C*a^2*b^6 + 45*B*a*b^7 + 7*A*b^8)*c^3)*d)*x)*sqrt(d*
x + c))/(a^4*b^9*c^5 - 5*a^5*b^8*c^4*d + 10*a^6*b^7*c^3*d^2 - 10*a^7*b^6*c^
2*d^3 + 5*a^8*b^5*c*d^4 - a^9*b^4*d^5 + (b^13*c^5 - 5*a*b^12*c^4*d + 10*a^2
*b^11*c^3*d^2 - 10*a^3*b^10*c^2*d^3 + 5*a^4*b^9*c*d^4 - a^5*b^8*d^5)*x^4 +
4*(a*b^12*c^5 - 5*a^2*b^11*c^4*d + 10*a^3*b^10*c^3*d^2 - 10*a^4*b^9*c^2*d^3
 + 5*a^5*b^8*c*d^4 - a^6*b^7*d^5)*x^3 + 6*(a^2*b^11*c^5 - 5*a^3*b^10*c^4*d
+ 10*a^4*b^9*c^3*d^2 - 10*a^5*b^8*c^2*d^3 + 5*a^6*b^7*c*d^4 - a^7*b^6*d^5)*
x^2 + 4*(a^3*b^10*c^5 - 5*a^4*b^9*c^4*d + 10*a^5*b^8*c^3*d^2 - 10*a^6*b^7*c
^2*d^3 + 5*a^7*b^6*c*d^4 - a^8*b^5*d^5)*x), -1/192*(3*(64*D*a^4*b^3*c^3*d -
 (5*D*a^7 + 3*C*a^6*b + 5*B*a^5*b^2 + 35*A*a^4*b^3)*d^4 + (64*D*b^7*c^3*d -
 (5*D*a^3*b^4 + 3*C*a^2*b^5 + 5*B*a*b^6 + 35*A*b^7)*d^4 + 8*(3*D*a^2*b^5*c
+ (2*C*a*b^6 + 5*B*b^7)*c)*d^3 - 48*(D*a*b^6*c^2 + C*b^7*c^2)*d^2)*x^4 + 8*
(3*D*a^6*b*c + (2*C*a^5*b^2 + 5*B*a^4*b^3)*c)*d^3 + 4*(64*D*a*b^6*c^3*d - (
5*D*a^4*b^3 + 3*C*a^3*b^4 + 5*B*a^2*b^5 + 35*A*a*b^6)*d^4 + 8*(3*D*a^3*b^4*
c + (2*C*a^2*b^5 + 5*B*a*b^6)*c)*d^3 - 48*(D*a^2*b^5*c^2 + C*a*b^6*c^2)*d^2
)*x^3 - 48*(D*a^5*b^2*c^2 + C*a^4*b^3*c^2)*d^2 + 6*(64*D*a^2*b^5*c^3*d - (5
```

```
*D*a^5*b^2 + 3*C*a^4*b^3 + 5*B*a^3*b^4 + 35*A*a^2*b^5)*d^4 + 8*(3*D*a^4*b^3
*c + (2*C*a^3*b^4 + 5*B*a^2*b^5)*c)*d^3 - 48*(D*a^3*b^4*c^2 + C*a^2*b^5*c^2
)*d^2)*x^2 + 4*(64*D*a^3*b^4*c^3*d - (5*D*a^6*b + 3*C*a^5*b^2 + 5*B*a^4*b^3
 + 35*A*a^3*b^4)*d^4 + 8*(3*D*a^5*b^2*c + (2*C*a^4*b^3 + 5*B*a^3*b^4)*c)*d^
3 - 48*(D*a^4*b^3*c^2 + C*a^3*b^4*c^2)*d^2)*x)*sqrt(-b^2*c + a*b*d)*arctan(
sqrt(-b^2*c + a*b*d)*sqrt(d*x + c)/(b*d*x + b*c)) + (48*D*a^3*b^5*c^4 + 16*
(C*a^2*b^6 + B*a*b^7 + 3*A*b^8)*c^4 - 3*(5*D*a^7*b + 3*C*a^6*b^2 + 5*B*a^5*
b^3 - 93*A*a^4*b^4)*d^4 + (77*D*a^6*b^2*c + (51*C*a^5*b^3 - 131*B*a^4*b^4 -
 605*A*a^3*b^5)*c)*d^3 + 3*(64*D*b^8*c^4 + (5*D*a^4*b^4 + 3*C*a^3*b^5 + 5*B
*a^2*b^6 + 35*A*a*b^7)*d^4 - (29*D*a^3*b^5*c + (19*C*a^2*b^6 + 45*B*a*b^7 +
 35*A*b^8)*c)*d^3 + 8*(9*D*a^2*b^6*c^2 + (8*C*a*b^7 + 5*B*b^8)*c^2)*d^2 - 1
6*(7*D*a*b^7*c^3 + 3*C*b^8*c^3)*d)*x^3 - 2*(83*D*a^5*b^3*c^2 - (23*C*a^4*b^
4 + 109*B*a^3*b^5 + 263*A*a^2*b^6)*c^2)*d^2 + (288*D*a*b^7*c^4 + 96*C*b^8*c
^4 - (73*D*a^5*b^3 - 33*C*a^4*b^4 - 55*B*a^3*b^5 - 385*A*a^2*b^6)*d^4 + (31
1*D*a^4*b^4*c - 5*(43*C*a^3*b^5 + 101*B*a^2*b^6 + 91*A*a*b^7)*c)*d^3 - 2*(2
15*D*a^3*b^5*c^2 - (371*C*a^2*b^6 + 265*B*a*b^7 + 35*A*b^8)*c^2)*d^2 - 16*(
6*D*a^2*b^6*c^3 + (41*C*a*b^7 + 5*B*b^8)*c^3)*d)*x^2 + 8*(7*D*a^4*b^4*c^3 -
 (13*C*a^3*b^5 + 11*B*a^2*b^6 + 31*A*a*b^7)*c^3)*d + (192*D*a^2*b^6*c^4 + 6
4*(C*a*b^7 + B*b^8)*c^4 - (55*D*a^6*b^2 + 33*C*a^5*b^3 - 73*B*a^4*b^4 - 511
*A*a^3*b^5)*d^4 + (283*D*a^5*b^3*c + (85*C*a^4*b^4 - 693*B*a^3*b^5 - 763*A*
a^2*b^6)*c)*d^3 - 4*(131*D*a^4*b^4*c^2 - (77*C*a^3*b^5 + 229*B*a^2*b^6 + 77
*A*a*b^7)*c^2)*d^2 + 8*(13*D*a^3*b^5*c^3 - (53*C*a^2*b^6 + 45*B*a*b^7 + 7*A
*b^8)*c^3)*d)*x)*sqrt(d*x + c))/(a^4*b^9*c^5 - 5*a^5*b^8*c^4*d + 10*a^6*b^7
*c^3*d^2 - 10*a^7*b^6*c^2*d^3 + 5*a^8*b^5*c*d^4 - a^9*b^4*d^5 + (b^13*c^5 -
 5*a*b^12*c^4*d + 10*a^2*b^11*c^3*d^2 - 10*a^3*b^10*c^2*d^3 + 5*a^4*b^9*c*d
^4 - a^5*b^8*d^5)*x^4 + 4*(a*b^12*c^5 - 5*a^2*b^11*c^4*d + 10*a^3*b^10*c^3*
d^2 - 10*a^4*b^9*c^2*d^3 + 5*a^5*b^8*c*d^4 - a^6*b^7*d^5)*x^3 + 6*(a^2*b^11
*c^5 - 5*a^3*b^10*c^4*d + 10*a^4*b^9*c^3*d^2 - 10*a^5*b^8*c^2*d^3 + 5*a^6*b
^7*c*d^4 - a^7*b^6*d^5)*x^2 + 4*(a^3*b^10*c^5 - 5*a^4*b^9*c^4*d + 10*a^5*b^
8*c^3*d^2 - 10*a^6*b^7*c^2*d^3 + 5*a^7*b^6*c*d^4 - a^8*b^5*d^5)*x)]
```

Sympy [F(-1)]

Timed out.

$$\int \frac{A + Bx + Cx^2 + Dx^3}{(a + bx)^5 \sqrt{c + dx}} \, dx = \text{Timed out}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(b*x+a)**5/(d*x+c)**(1/2),x)
```

```
[Out] Timed out
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^5\sqrt{c+dx}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^5/(d*x+c)^(1/2),x, algorithm="maxima"
)
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1512 vs. $2(469) = 938$.

Time = 0.32 (sec) , antiderivative size = 1512, normalized size of antiderivative = 3.05

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^5\sqrt{c+dx}}\,dx = \text{Too large to display}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^5/(d*x+c)^(1/2),x, algorithm="giac")
```

```
[Out] -1/64*(64*D*b^3*c^3*d - 48*D*a*b^2*c^2*d^2 - 48*C*b^3*c^2*d^2 + 24*D*a^2*b*
c*d^3 + 16*C*a*b^2*c*d^3 + 40*B*b^3*c*d^3 - 5*D*a^3*d^4 - 3*C*a^2*b*d^4 - 5
*B*a*b^2*d^4 - 35*A*b^3*d^4)*arctan(sqrt(d*x + c)*b/sqrt(-b^2*c + a*b*d))/(
(b^7*c^4 - 4*a*b^6*c^3*d + 6*a^2*b^5*c^2*d^2 - 4*a^3*b^4*c*d^3 + a^4*b^3*d^
4)*sqrt(-b^2*c + a*b*d)) - 1/192*(192*(d*x + c)^(7/2)*D*b^6*c^3*d - 576*(d*
x + c)^(5/2)*D*b^6*c^4*d + 576*(d*x + c)^(3/2)*D*b^6*c^5*d - 192*sqrt(d*x +
 c)*D*b^6*c^6*d - 144*(d*x + c)^(7/2)*D*a*b^5*c^2*d^2 - 144*(d*x + c)^(7/2)
*C*b^6*c^2*d^2 + 720*(d*x + c)^(5/2)*D*a*b^5*c^3*d^2 + 528*(d*x + c)^(5/2)*
C*b^6*c^3*d^2 - 1008*(d*x + c)^(3/2)*D*a*b^5*c^4*d^2 - 624*(d*x + c)^(3/2)*
C*b^6*c^4*d^2 + 432*sqrt(d*x + c)*D*a*b^5*c^5*d^2 + 240*sqrt(d*x + c)*C*b^6
*c^5*d^2 + 72*(d*x + c)^(7/2)*D*a^2*b^4*c*d^3 + 48*(d*x + c)^(7/2)*C*a*b^5*
c*d^3 + 120*(d*x + c)^(7/2)*B*b^6*c*d^3 - 24*(d*x + c)^(5/2)*D*a^2*b^4*c^2*
d^3 - 704*(d*x + c)^(5/2)*C*a*b^5*c^2*d^3 - 440*(d*x + c)^(5/2)*B*b^6*c^2*d
^3 + 24*(d*x + c)^(3/2)*D*a^2*b^4*c^3*d^3 + 1328*(d*x + c)^(3/2)*C*a*b^5*c^
3*d^3 + 584*(d*x + c)^(3/2)*B*b^6*c^3*d^3 - 72*sqrt(d*x + c)*D*a^2*b^4*c^4*
d^3 - 672*sqrt(d*x + c)*C*a*b^5*c^4*d^3 - 264*sqrt(d*x + c)*B*b^6*c^4*d^3 -
 15*(d*x + c)^(7/2)*D*a^3*b^3*d^4 - 9*(d*x + c)^(7/2)*C*a^2*b^4*d^4 - 15*(d
*x + c)^(7/2)*B*a*b^5*d^4 - 105*(d*x + c)^(7/2)*A*b^6*d^4 - 193*(d*x + c)^(
5/2)*D*a^3*b^3*c*d^4 + 209*(d*x + c)^(5/2)*C*a^2*b^4*c*d^4 + 495*(d*x + c)^
(5/2)*B*a*b^5*c*d^4 + 385*(d*x + c)^(5/2)*A*b^6*c*d^4 + 727*(d*x + c)^(3/2)
```

```
*D*a^3*b^3*c^2*d^4 - 751*(d*x + c)^(3/2)*C*a^2*b^4*c^2*d^4 - 1241*(d*x + c)
^(3/2)*B*a*b^5*c^2*d^4 - 511*(d*x + c)^(3/2)*A*b^6*c^2*d^4 - 471*sqrt(d*x +
 c)*D*a^3*b^3*c^3*d^4 + 567*sqrt(d*x + c)*C*a^2*b^4*c^3*d^4 + 777*sqrt(d*x
+ c)*B*a*b^5*c^3*d^4 + 279*sqrt(d*x + c)*A*b^6*c^3*d^4 + 73*(d*x + c)^(5/2)
*D*a^4*b^2*d^5 - 33*(d*x + c)^(5/2)*C*a^3*b^3*d^5 - 55*(d*x + c)^(5/2)*B*a^
2*b^4*d^5 - 385*(d*x + c)^(5/2)*A*a*b^5*d^5 - 374*(d*x + c)^(3/2)*D*a^4*b^2
*c*d^5 + 14*(d*x + c)^(3/2)*C*a^3*b^3*c*d^5 + 730*(d*x + c)^(3/2)*B*a^2*b^4
*c*d^5 + 1022*(d*x + c)^(3/2)*A*a*b^5*c*d^5 + 405*sqrt(d*x + c)*D*a^4*b^2*c
^2*d^5 - 69*sqrt(d*x + c)*C*a^3*b^3*c^2*d^5 - 747*sqrt(d*x + c)*B*a^2*b^4*c
^2*d^5 - 837*sqrt(d*x + c)*A*a*b^5*c^2*d^5 + 55*(d*x + c)^(3/2)*D*a^5*b*d^6
 + 33*(d*x + c)^(3/2)*C*a^4*b^2*d^6 - 73*(d*x + c)^(3/2)*B*a^3*b^3*d^6 - 51
1*(d*x + c)^(3/2)*A*a^2*b^4*d^6 - 117*sqrt(d*x + c)*D*a^5*b*c*d^6 - 75*sqrt
(d*x + c)*C*a^4*b^2*c*d^6 + 219*sqrt(d*x + c)*B*a^3*b^3*c*d^6 + 837*sqrt(d*
x + c)*A*a^2*b^4*c*d^6 + 15*sqrt(d*x + c)*D*a^6*d^7 + 9*sqrt(d*x + c)*C*a^5
*b*d^7 + 15*sqrt(d*x + c)*B*a^4*b^2*d^7 - 279*sqrt(d*x + c)*A*a^3*b^3*d^7)/
((b^7*c^4 - 4*a*b^6*c^3*d + 6*a^2*b^5*c^2*d^2 - 4*a^3*b^4*c*d^3 + a^4*b^3*d
^4)*((d*x + c)*b - b*c + a*d)^4)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^5\sqrt{c+dx}}\,dx=\int \frac{A+B\,x+C\,x^2+x^3\,D}{(a+b\,x)^5\,\sqrt{c+d\,x}}\,dx$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^5*(c + d*x)^(1/2)),x)
```

```
[Out] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^5*(c + d*x)^(1/2)), x)
```

3.10 $\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 434

$$
\begin{aligned}
&\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx=\frac{2(bc-ad)^3\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{d^7\sqrt{c+dx}}\\
&-\frac{2(bc-ad)^2\left(ad(2cCd-Bd^2-3c^2D)-b(5c^2Cd-4Bcd^2+3Ad^3-6c^3D)\right)\sqrt{c+dx}}{d^7}\\
&-\frac{2(bc-ad)\left(a^2d^2(Cd-3cD)-abd(8cCd-3Bd^2-15c^2D)+b^2(10c^2Cd-6Bcd^2+3Ad^3-15c^3D)\right)(c+}{3d^7}\\
&+\frac{2(a^3d^3D+3a^2bd^2(Cd-4cD)-3ab^2d(4cCd-Bd^2-10c^2D)+b^3(10c^2Cd-4Bcd^2+Ad^3-20c^3D))(c}{5d^7}\\
&+\frac{2b(3a^2d^2D+3abd(Cd-5cD)-b^2(5cCd-Bd^2-15c^2D))(c+dx)^{7/2}}{7d^7}\\
&+\frac{2b^2(bCd-6bcD+3adD)(c+dx)^{9/2}}{9d^7}+\frac{2b^3D(c+dx)^{11/2}}{11d^7}
\end{aligned}
$$

```
[Out] -2/3*(-a*d+b*c)*(a^2*d^2*(C*d-3*D*c)-a*b*d*(-3*B*d^2+8*C*c*d-15*D*c^2)+b^2*
(3*A*d^3-6*B*c*d^2+10*C*c^2*d-15*D*c^3))*(d*x+c)^(3/2)/d^7+2/5*(a^3*d^3*D+3
*a^2*b*d^2*(C*d-4*D*c)-3*a*b^2*d*(-B*d^2+4*C*c*d-10*D*c^2)+b^3*(A*d^3-4*B*c
*d^2+10*C*c^2*d-20*D*c^3))*(d*x+c)^(5/2)/d^7+2/7*b*(3*a^2*d^2*D+3*a*b*d*(C*
d-5*D*c)-b^2*(-B*d^2+5*C*c*d-15*D*c^2))*(d*x+c)^(7/2)/d^7+2/9*b^2*(C*b*d+3*
D*a*d-6*D*b*c)*(d*x+c)^(9/2)/d^7+2/11*b^3*D*(d*x+c)^(11/2)/d^7+2*(-a*d+b*c)
^3*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)/d^7/(d*x+c)^(1/2)-2*(-a*d+b*c)^2*(a*d*(-B*
d^2+2*C*c*d-3*D*c^2)-b*(3*A*d^3-4*B*c*d^2+5*C*c^2*d-6*D*c^3))*(d*x+c)^(1/2)
/d^7
```

Rubi [A] (verified)

Time = 0.25 (sec) , antiderivative size = 434, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}} = 0.031$, Rules used = {1634}

$$
\begin{aligned}
&\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx = \\
&-\frac{2(c+dx)^{3/2}(bc-ad)\left(a^2d^2(Cd-3cD)-abd(-3Bd^2-15c^2D+8cCd)+b^2(3Ad^3-6Bcd^2-15c^3D+10\right.}{3d^7} \\
&+\frac{2b(c+dx)^{7/2}\left(3a^2d^2D+3abd(Cd-5cD)-\left(b^2(-Bd^2-15c^2D+5cCd)\right)\right)}{7d^7} \\
&+\frac{2(c+dx)^{5/2}\left(a^3d^3D+3a^2bd^2(Cd-4cD)-3ab^2d(-Bd^2-10c^2D+4cCd)+b^3(Ad^3-4Bcd^2-20c^3D+1\right.}{5d^7} \\
&-\frac{2\sqrt{c+dx}(bc-ad)^2\left(ad(-Bd^2-3c^2D+2cCd)-b(3Ad^3-4Bcd^2-6c^3D+5c^2Cd)\right)}{d^7} \\
&+\frac{2(bc-ad)^3\left(Ad^3-Bcd^2+c^3(-D)+c^2Cd\right)}{d^7\sqrt{c+dx}} \\
&+\frac{2b^2(c+dx)^{9/2}(3adD-6bcD+bCd)}{9d^7}+\frac{2b^3D(c+dx)^{11/2}}{11d^7}
\end{aligned}
$$

```
[In] Int[((a + b*x)^3*(A + B*x + C*x^2 + D*x^3))/(c + d*x)^(3/2),x]
```

```
[Out] (2*(b*c - a*d)^3*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D))/(d^7*Sqrt[c + d*x]) -
 (2*(b*c - a*d)^2*(a*d*(2*c*C*d - B*d^2 - 3*c^2*D) - b*(5*c^2*C*d - 4*B*c*d
^2 + 3*A*d^3 - 6*c^3*D))*Sqrt[c + d*x])/d^7 - (2*(b*c - a*d)*(a^2*d^2*(C*d
- 3*c*D) - a*b*d*(8*c*C*d - 3*B*d^2 - 15*c^2*D) + b^2*(10*c^2*C*d - 6*B*c*d
^2 + 3*A*d^3 - 15*c^3*D))*(c + d*x)^(3/2))/(3*d^7) + (2*(a^3*d^3*D + 3*a^2*
b*d^2*(C*d - 4*c*D) - 3*a*b^2*d*(4*c*C*d - B*d^2 - 10*c^2*D) + b^3*(10*c^2*
C*d - 4*B*c*d^2 + A*d^3 - 20*c^3*D))*(c + d*x)^(5/2))/(5*d^7) + (2*b*(3*a^2
*d^2*D + 3*a*b*d*(C*d - 5*c*D) - b^2*(5*c*C*d - B*d^2 - 15*c^2*D))*(c + d*x
)^(7/2))/(7*d^7) + (2*b^2*(b*C*d - 6*b*c*D + 3*a*d*D)*(c + d*x)^(9/2))/(9*d
^7) + (2*b^3*D*(c + d*x)^(11/2))/(11*d^7)
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} = \int \Bigg(& \frac{(-bc+ad)^3\,(c^2Cd-Bcd^2+Ad^3-c^3D)}{d^6(c+dx)^{3/2}} \\
& + \frac{(bc-ad)^2\,(-ad(2cCd-Bd^2-3c^2D)+b(5c^2Cd-4Bcd^2+3Ad^3-6c^3D))}{d^6\sqrt{c+dx}} \\
& + \frac{(bc-ad)\,(-a^2d^2(Cd-3cD)+abd(8cCd-3Bd^2-15c^2D)-b^2(10c^2Cd-6Bcd^2+3Ad^3-15c^3D))}{d^6} \\
& + \frac{(a^3d^3D+3a^2bd^2(Cd-4cD)-3ab^2d(4cCd-Bd^2-10c^2D)+b^3(10c^2Cd-4Bcd^2+Ad^3-20c^3D))\,(}{d^6} \\
& + \frac{b(3a^2d^2D+3abd(Cd-5cD)-b^2(5cCd-Bd^2-15c^2D))\,(c+dx)^{5/2}}{d^6} \\
& + \frac{b^2(bCd-6bcD+3adD)(c+dx)^{7/2}}{d^6} + \frac{b^3D(c+dx)^{9/2}}{d^6} \Bigg)\,dx \\
= {} & \frac{2(bc-ad)^3\,(c^2Cd-Bcd^2+Ad^3-c^3D)}{d^7\sqrt{c+dx}} \\
& - \frac{2(bc-ad)^2\,(ad(2cCd-Bd^2-3c^2D)-b(5c^2Cd-4Bcd^2+3Ad^3-6c^3D))\,\sqrt{c+dx}}{d^7} \\
& - \frac{2(bc-ad)\,(a^2d^2(Cd-3cD)-abd(8cCd-3Bd^2-15c^2D)+b^2(10c^2Cd-6Bcd^2+3Ad^3-15}{3d^7} \\
& + \frac{2(a^3d^3D+3a^2bd^2(Cd-4cD)-3ab^2d(4cCd-Bd^2-10c^2D)+b^3(10c^2Cd-4Bcd^2+Ad^3-2}{5d^7} \\
& + \frac{2b(3a^2d^2D+3abd(Cd-5cD)-b^2(5cCd-Bd^2-15c^2D))\,(c+dx)^{7/2}}{7d^7} \\
& + \frac{2b^2(bCd-6bcD+3adD)(c+dx)^{9/2}}{9d^7} + \frac{2b^3D(c+dx)^{11/2}}{11d^7}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.47 (sec) , antiderivative size = 500, normalized size of antiderivative = 1.15

$$\int \frac{(a+bx)^3\,(A+Bx+Cx^2+Dx^3)}{(c+dx)^{3/2}}\,dx = 2(231a^3d^3(48c^3D-8c^2d(5C-3Dx)+2cd^2(15B-x(10C+3D$$

```
[In] Integrate[((a + b*x)^3*(A + B*x + C*x^2 + D*x^3))/(c + d*x)^(3/2),x]

[Out] (2*(231*a^3*d^3*(48*c^3*D - 8*c^2*d*(5*C - 3*D*x) + 2*c*d^2*(15*B - x*(10*C
 + 3*D*x)) + d^3*(-15*A + x*(15*B + 5*C*x + 3*D*x^2))) + 99*a^2*b*d^2*(-384
*c^4*D + 48*c^3*d*(7*C - 4*D*x) - 8*c^2*d^2*(35*B - 3*x*(7*C + 2*D*x)) + 2*
c*d^3*(105*A - x*(70*B + 3*x*(7*C + 4*D*x))) + d^4*x*(105*A + x*(35*B + 3*x
*(7*C + 5*D*x)))) + 33*a*b^2*d*(1280*c^5*D - 128*c^4*d*(9*C - 5*D*x) + 16*c
^3*d^2*(63*B - 2*x*(18*C + 5*D*x)) + 8*c^2*d^3*(-105*A + x*(63*B + 2*x*(9*C
 + 5*D*x))) + d^5*x^2*(105*A + x*(63*B + 5*x*(9*C + 7*D*x))) - 2*c*d^4*x*(2
```

```
10*A + x*(63*B + x*(36*C + 25*D*x)))) + b^3*(-15360*c^6*D + 1280*c^5*d*(11*
C - 6*D*x) - 128*c^4*d^2*(99*B - 5*x*(11*C + 3*D*x)) + 16*c^3*d^3*(693*A -
2*x*(198*B + 5*x*(11*C + 6*D*x))) + d^6*x^3*(693*A + 5*x*(99*B + 7*x*(11*C
+ 9*D*x))) + 8*c^2*d^4*x*(693*A + x*(198*B + 5*x*(22*C + 15*D*x))) - 2*c*d^
5*x^2*(693*A + x*(396*B + 5*x*(55*C + 42*D*x))))))/(3465*d^7*Sqrt[c + d*x])
```

Maple [A] (verified)

Time = 1.77 (sec) , antiderivative size = 440, normalized size of antiderivative = 1.01

method	result
pseudoelliptic	$-\frac{2((-\frac{x^{3}(\frac{5}{11}Dx^{3}+\frac{5}{9}Cx^{2}+\frac{5}{7}Bx+A)b^{3}}{5}-ax^{2}(\frac{1}{3}Dx^{3}+\frac{3}{7}Cx^{2}+\frac{3}{5}Bx+A)b^{2}-3a^{2}(A+\frac{1}{7}Dx^{3}+\frac{1}{5}Cx^{2}+\frac{1}{3}Bx)xb+a^{3}(-\frac{1}{5}D}{}$
gosper	$-\frac{2(-315Db^{3}x^{6}d^{6}-385Cb^{3}d^{6}x^{5}-1155Dab^{2}d^{6}x^{5}+420Db^{3}cd^{5}x^{5}-495Bb^{3}d^{6}x^{4}-1485Cab^{2}d^{6}x^{4}+550Cb^{3}cd^{5}x^{4}-1485}{}$
trager	$-\frac{2(-315Db^{3}x^{6}d^{6}-385Cb^{3}d^{6}x^{5}-1155Dab^{2}d^{6}x^{5}+420Db^{3}cd^{5}x^{5}-495Bb^{3}d^{6}x^{4}-1485Cab^{2}d^{6}x^{4}+550Cb^{3}cd^{5}x^{4}-1485}{}$
derivativedivides	Expression too large to display
default	Expression too large to display

```
[In] int((b*x+a)^3*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x,method=_RETURNVERBOSE)
```

```
[Out] -2/(d*x+c)^(1/2)*((-1/5*x^3*(5/11*D*x^3+5/9*C*x^2+5/7*B*x+A)*b^3-a*x^2*(1/3
*D*x^3+3/7*C*x^2+3/5*B*x+A)*b^2-3*a^2*(A+1/7*D*x^3+1/5*C*x^2+1/3*B*x)*x*b+a
^3*(-1/5*D*x^3-1/3*C*x^2-B*x+A))*d^6-6*c*(-1/15*x^2*(10/33*D*x^3+25/63*C*x^
2+4/7*B*x+A)*b^3-2/3*a*(5/42*D*x^3+6/35*C*x^2+3/10*B*x+A)*x*b^2+a^2*(-4/35*
D*x^3-1/5*C*x^2-2/3*B*x+A)*b+1/3*a^3*(-1/5*D*x^2-2/3*C*x+B))*d^5+8*(-1/5*(2
5/231*D*x^3+10/63*C*x^2+2/7*B*x+A)*x*b^3+a*(-2/21*D*x^3-6/35*C*x^2-3/5*B*x+
A)*b^2+a^2*(-6/35*D*x^2-3/5*C*x+B)*b+1/3*a^3*(-3/5*D*x+C))*c^2*d^4-16/5*c^3
*((-20/231*D*x^3-10/63*C*x^2-4/7*B*x+A)*b^3+3*a*(-10/63*D*x^2-4/7*C*x+B)*b^
2+3*a^2*(-4/7*D*x+C)*b+D*a^3)*d^3+128/35*b*c^4*((-5/33*D*x^2-5/9*C*x+B)*b^2
+3*a*(-5/9*D*x+C)*b+3*D*a^2)*d^2-256/63*((-6/11*D*x+C)*b+3*D*a)*b^2*c^5*d+1
024/231*D*b^3*c^6)/d^7
```

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 677, normalized size of antiderivative = 1.56

$$\int \frac{(a+bx)^3\,(A+Bx+Cx^2+Dx^3)}{(c+dx)^{3/2}}\,dx = \frac{2\,(315\,Db^3d^6x^6-15360\,Db^3c^6-3465\,Aa^3d^6-9240\,(Ca^3+3\,Ba^2}{}$$

```
[In] integrate((b*x+a)^3*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x, algorithm="fricas"
)
```

```
[Out] 2/3465*(315*D*b^3*d^6*x^6 - 15360*D*b^3*c^6 - 3465*A*a^3*d^6 - 9240*(C*a^3
+ 3*B*a^2*b + 3*A*a*b^2)*c^2*d^4 + 6930*(B*a^3 + 3*A*a^2*b)*c*d^5 - 35*(12*
D*b^3*c*d^5 - 11*(3*D*a*b^2 + C*b^3)*d^6)*x^5 + 5*(120*D*b^3*c^2*d^4 + 99*(
3*D*a^2*b + 3*C*a*b^2 + B*b^3)*d^6 - 110*(3*D*a*b^2*c + C*b^3*c)*d^5)*x^4 +
 11088*(D*a^3*c^3 + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^3)*d^3 - (960*D*b^3*c
^3*d^3 - 693*(D*a^3 + 3*C*a^2*b + 3*B*a*b^2 + A*b^3)*d^6 + 792*(3*D*a^2*b*c
 + (3*C*a*b^2 + B*b^3)*c)*d^5 - 880*(3*D*a*b^2*c^2 + C*b^3*c^2)*d^4)*x^3 -
12672*(3*D*a^2*b*c^4 + (3*C*a*b^2 + B*b^3)*c^4)*d^2 + (1920*D*b^3*c^4*d^2 +
 1155*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*d^6 - 1386*(D*a^3*c + (3*C*a^2*b + 3*
B*a*b^2 + A*b^3)*c)*d^5 + 1584*(3*D*a^2*b*c^2 + (3*C*a*b^2 + B*b^3)*c^2)*d^
4 - 1760*(3*D*a*b^2*c^3 + C*b^3*c^3)*d^3)*x^2 + 14080*(3*D*a*b^2*c^5 + C*b^
3*c^5)*d - (7680*D*b^3*c^5*d + 4620*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c*d^5 -
 3465*(B*a^3 + 3*A*a^2*b)*d^6 - 5544*(D*a^3*c^2 + (3*C*a^2*b + 3*B*a*b^2 +
A*b^3)*c^2)*d^4 + 6336*(3*D*a^2*b*c^3 + (3*C*a*b^2 + B*b^3)*c^3)*d^3 - 7040
*(3*D*a*b^2*c^4 + C*b^3*c^4)*d^2)*x)*sqrt(d*x + c)/(d^8*x + c*d^7)
```

Sympy [A] (verification not implemented)

Time = 70.97 (sec) , antiderivative size = 853, normalized size of antiderivative = 1.97

$$\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx=\begin{cases}\frac{2\left(\frac{Db^3(c+dx)^{\frac{11}{2}}}{11d^6}+\frac{(c+dx)^{\frac{9}{2}}\left(Cb^3d+3Dab^2d-6Db^3c\right)}{9d^6}+\frac{(c+dx)^{\frac{7}{2}}\left(Bb^3d^2+3Cab^2d^2-5Cb^3cd-\right.}{7d^6}\right.}{} \\ \frac{Aa^3x+\frac{Db^3x^7}{7}+\frac{x^6\left(Cb^3+3Dab^2\right)}{6}+\frac{x^5\left(Bb^3+3Cab^2+3Da^2b\right)}{5}+\frac{x^4\left(Ab^3+3Bab^2+3Ca^2b+Da\right.}{4}}{c^{\frac{3}{2}}}\end{cases}$$

```
[In] integrate((b*x+a)**3*(D*x**3+C*x**2+B*x+A)/(d*x+c)**(3/2),x)
```

```
[Out] Piecewise((2*(D*b**3*(c + d*x)**(11/2)/(11*d**6) + (c + d*x)**(9/2)*(C*b**3
*d + 3*D*a*b**2*d - 6*D*b**3*c)/(9*d**6) + (c + d*x)**(7/2)*(B*b**3*d**2 +
3*C*a*b**2*d**2 - 5*C*b**3*c*d + 3*D*a**2*b*d**2 - 15*D*a*b**2*c*d + 15*D*b
**3*c**2)/(7*d**6) + (c + d*x)**(5/2)*(A*b**3*d**3 + 3*B*a*b**2*d**3 - 4*B*
b**3*c*d**2 + 3*C*a**2*b*d**3 - 12*C*a*b**2*c*d**2 + 10*C*b**3*c**2*d + D*a
**3*d**3 - 12*D*a**2*b*c*d**2 + 30*D*a*b**2*c**2*d - 20*D*b**3*c**3)/(5*d**
6) + (c + d*x)**(3/2)*(3*A*a*b**2*d**4 - 3*A*b**3*c*d**3 + 3*B*a**2*b*d**4
- 9*B*a*b**2*c*d**3 + 6*B*b**3*c**2*d**2 + C*a**3*d**4 - 9*C*a**2*b*c*d**3
+ 18*C*a*b**2*c**2*d**2 - 10*C*b**3*c**3*d - 3*D*a**3*c*d**3 + 18*D*a**2*b*
c**2*d**2 - 30*D*a*b**2*c**3*d + 15*D*b**3*c**4)/(3*d**6) + sqrt(c + d*x)*(
3*A*a**2*b*d**5 - 6*A*a*b**2*c*d**4 + 3*A*b**3*c**2*d**3 + B*a**3*d**5 - 6*
B*a**2*b*c*d**4 + 9*B*a*b**2*c**2*d**3 - 4*B*b**3*c**3*d**2 - 2*C*a**3*c*d*
*4 + 9*C*a**2*b*c**2*d**3 - 12*C*a*b**2*c**3*d**2 + 5*C*b**3*c**4*d + 3*D*a
**3*c**2*d**3 - 12*D*a**2*b*c**3*d**2 + 15*D*a*b**2*c**4*d - 6*D*b**3*c**5)
/d**6 + (a*d - b*c)**3*(-A*d**3 + B*c*d**2 - C*c**2*d + D*c**3)/(d**6*sqrt(
c + d*x)))/d, Ne(d, 0)), ((A*a**3*x + D*b**3*x**7/7 + x**6*(C*b**3 + 3*D*a*
b**2)/6 + x**5*(B*b**3 + 3*C*a*b**2 + 3*D*a**2*b)/5 + x**4*(A*b**3 + 3*B*a*
```

```
b**2 + 3*C*a**2*b + D*a**3)/4 + x**3*(3*A*a*b**2 + 3*B*a**2*b + C*a**3)/3 +
 x**2*(3*A*a**2*b + B*a**3)/2)/c**(3/2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 629, normalized size of antiderivative = 1.45

$$\int \frac{(a+bx)^3\,(A+Bx+Cx^2+Dx^3)}{(c+dx)^{3/2}}\,dx = \frac{2\left(\frac{315\,(dx+c)^{\frac{11}{2}}\,Db^3-385\,\left(6\,Db^3c-\left(3\,Dab^2+Cb^3\right)d\right)(dx+c)^{\frac{9}{2}}+495\,\left(15\,Db^3c^2-5\,\left(3\,Dab\right.\right.}{}\right.}{}$$

```
[In] integrate((b*x+a)^3*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x, algorithm="maxima"
)
```

```
[Out] 2/3465*((315*(d*x + c)^(11/2)*D*b^3 - 385*(6*D*b^3*c - (3*D*a*b^2 + C*b^3)*
d)*(d*x + c)^(9/2) + 495*(15*D*b^3*c^2 - 5*(3*D*a*b^2 + C*b^3)*c*d + (3*D*a
^2*b + 3*C*a*b^2 + B*b^3)*d^2)*(d*x + c)^(7/2) - 693*(20*D*b^3*c^3 - 10*(3*
D*a*b^2 + C*b^3)*c^2*d + 4*(3*D*a^2*b + 3*C*a*b^2 + B*b^3)*c*d^2 - (D*a^3 +
 3*C*a^2*b + 3*B*a*b^2 + A*b^3)*d^3)*(d*x + c)^(5/2) + 1155*(15*D*b^3*c^4 -
 10*(3*D*a*b^2 + C*b^3)*c^3*d + 6*(3*D*a^2*b + 3*C*a*b^2 + B*b^3)*c^2*d^2 -
 3*(D*a^3 + 3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c*d^3 + (C*a^3 + 3*B*a^2*b + 3*A
*a*b^2)*d^4)*(d*x + c)^(3/2) - 3465*(6*D*b^3*c^5 - 5*(3*D*a*b^2 + C*b^3)*c^
4*d + 4*(3*D*a^2*b + 3*C*a*b^2 + B*b^3)*c^3*d^2 - 3*(D*a^3 + 3*C*a^2*b + 3*
B*a*b^2 + A*b^3)*c^2*d^3 + 2*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c*d^4 - (B*a^3
 + 3*A*a^2*b)*d^5)*sqrt(d*x + c))/d^6 - 3465*(D*b^3*c^6 + A*a^3*d^6 - (3*D*
a*b^2 + C*b^3)*c^5*d + (3*D*a^2*b + 3*C*a*b^2 + B*b^3)*c^4*d^2 - (D*a^3 + 3
*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^3*d^3 + (C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^2
*d^4 - (B*a^3 + 3*A*a^2*b)*c*d^5)/(sqrt(d*x + c)*d^6))/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1067 vs. $2(412) = 824$.

Time = 0.31 (sec) , antiderivative size = 1067, normalized size of antiderivative = 2.46

$$\int \frac{(a+bx)^3\,(A+Bx+Cx^2+Dx^3)}{(c+dx)^{3/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((b*x+a)^3*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x, algorithm="giac")
```

```
[Out] -2*(D*b^3*c^6 - 3*D*a*b^2*c^5*d - C*b^3*c^5*d + 3*D*a^2*b*c^4*d^2 + 3*C*a*b
^2*c^4*d^2 + B*b^3*c^4*d^2 - D*a^3*c^3*d^3 - 3*C*a^2*b*c^3*d^3 - 3*B*a*b^2*
c^3*d^3 - A*b^3*c^3*d^3 + C*a^3*c^2*d^4 + 3*B*a^2*b*c^2*d^4 + 3*A*a*b^2*c^2
*d^4 - B*a^3*c*d^5 - 3*A*a^2*b*c*d^5 + A*a^3*d^6)/(sqrt(d*x + c)*d^7) + 2/3
```

```
465*(315*(d*x + c)^(11/2)*D*b^3*d^70 - 2310*(d*x + c)^(9/2)*D*b^3*c*d^70 +
7425*(d*x + c)^(7/2)*D*b^3*c^2*d^70 - 13860*(d*x + c)^(5/2)*D*b^3*c^3*d^70
+ 17325*(d*x + c)^(3/2)*D*b^3*c^4*d^70 - 20790*sqrt(d*x + c)*D*b^3*c^5*d^70
 + 1155*(d*x + c)^(9/2)*D*a*b^2*d^71 + 385*(d*x + c)^(9/2)*C*b^3*d^71 - 742
5*(d*x + c)^(7/2)*D*a*b^2*c*d^71 - 2475*(d*x + c)^(7/2)*C*b^3*c*d^71 + 2079
0*(d*x + c)^(5/2)*D*a*b^2*c^2*d^71 + 6930*(d*x + c)^(5/2)*C*b^3*c^2*d^71 -
34650*(d*x + c)^(3/2)*D*a*b^2*c^3*d^71 - 11550*(d*x + c)^(3/2)*C*b^3*c^3*d^
71 + 51975*sqrt(d*x + c)*D*a*b^2*c^4*d^71 + 17325*sqrt(d*x + c)*C*b^3*c^4*d
^71 + 1485*(d*x + c)^(7/2)*D*a^2*b*d^72 + 1485*(d*x + c)^(7/2)*C*a*b^2*d^72
 + 495*(d*x + c)^(7/2)*B*b^3*d^72 - 8316*(d*x + c)^(5/2)*D*a^2*b*c*d^72 - 8
316*(d*x + c)^(5/2)*C*a*b^2*c*d^72 - 2772*(d*x + c)^(5/2)*B*b^3*c*d^72 + 20
790*(d*x + c)^(3/2)*D*a^2*b*c^2*d^72 + 20790*(d*x + c)^(3/2)*C*a*b^2*c^2*d^
72 + 6930*(d*x + c)^(3/2)*B*b^3*c^2*d^72 - 41580*sqrt(d*x + c)*D*a^2*b*c^3*
d^72 - 41580*sqrt(d*x + c)*C*a*b^2*c^3*d^72 - 13860*sqrt(d*x + c)*B*b^3*c^3
*d^72 + 693*(d*x + c)^(5/2)*D*a^3*d^73 + 2079*(d*x + c)^(5/2)*C*a^2*b*d^73
+ 2079*(d*x + c)^(5/2)*B*a*b^2*d^73 + 693*(d*x + c)^(5/2)*A*b^3*d^73 - 3465
*(d*x + c)^(3/2)*D*a^3*c*d^73 - 10395*(d*x + c)^(3/2)*C*a^2*b*c*d^73 - 1039
5*(d*x + c)^(3/2)*B*a*b^2*c*d^73 - 3465*(d*x + c)^(3/2)*A*b^3*c*d^73 + 1039
5*sqrt(d*x + c)*D*a^3*c^2*d^73 + 31185*sqrt(d*x + c)*C*a^2*b*c^2*d^73 + 311
85*sqrt(d*x + c)*B*a*b^2*c^2*d^73 + 10395*sqrt(d*x + c)*A*b^3*c^2*d^73 + 11
55*(d*x + c)^(3/2)*C*a^3*d^74 + 3465*(d*x + c)^(3/2)*B*a^2*b*d^74 + 3465*(d
*x + c)^(3/2)*A*a*b^2*d^74 - 6930*sqrt(d*x + c)*C*a^3*c*d^74 - 20790*sqrt(d
*x + c)*B*a^2*b*c*d^74 - 20790*sqrt(d*x + c)*A*a*b^2*c*d^74 + 3465*sqrt(d*x
 + c)*B*a^3*d^75 + 10395*sqrt(d*x + c)*A*a^2*b*d^75)/d^77
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+bx)^3 (A+Bx+Cx^2+Dx^3)}{(c+dx)^{3/2}}\,dx = \int \frac{(a+b\,x)^3\,(A+B\,x+C\,x^2+x^3\,D)}{(c+d\,x)^{3/2}}\,dx$$

```
[In] int(((a + b*x)^3*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(3/2),x)
[Out] int(((a + b*x)^3*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(3/2), x)
```

3.11 $\int \frac{(a+bx)^2(A+Bx+Cx^2+Dx^3)}{(c+dx)^{3/2}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 322

$$
\begin{aligned}
\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx &= -\frac{2(bc-ad)^2\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{d^6\sqrt{c+dx}} \\
&+ \frac{2(bc-ad)\left(ad(2cCd-Bd^2-3c^2D)-b(4c^2Cd-3Bcd^2+2Ad^3-5c^3D)\right)\sqrt{c+dx}}{d^6} \\
&+ \frac{2(a^2d^2(Cd-3cD)-2abd(3cCd-Bd^2-6c^2D)+b^2(6c^2Cd-3Bcd^2+Ad^3-10c^3D))\,(c+dx)^{3/2}}{3d^6} \\
&+ \frac{2(a^2d^2D+2abd(Cd-4cD)-b^2(4cCd-Bd^2-10c^2D))\,(c+dx)^{5/2}}{5d^6} \\
&+ \frac{2b(bCd-5bcD+2adD)(c+dx)^{7/2}}{7d^6} + \frac{2b^2D(c+dx)^{9/2}}{9d^6}
\end{aligned}
$$

```
[Out] 2/3*(a^2*d^2*(C*d-3*D*c)-2*a*b*d*(-B*d^2+3*C*c*d-6*D*c^2)+b^2*(A*d^3-3*B*c*
d^2+6*C*c^2*d-10*D*c^3))*(d*x+c)^(3/2)/d^6+2/5*(a^2*d^2*D+2*a*b*d*(C*d-4*D*
c)-b^2*(-B*d^2+4*C*c*d-10*D*c^2))*(d*x+c)^(5/2)/d^6+2/7*b*(C*b*d+2*D*a*d-5*
D*b*c)*(d*x+c)^(7/2)/d^6+2/9*b^2*D*(d*x+c)^(9/2)/d^6-2*(-a*d+b*c)^2*(A*d^3-
B*c*d^2+C*c^2*d-D*c^3)/d^6/(d*x+c)^(1/2)+2*(-a*d+b*c)*(a*d*(-B*d^2+2*C*c*d-
3*D*c^2)-b*(2*A*d^3-3*B*c*d^2+4*C*c^2*d-5*D*c^3))*(d*x+c)^(1/2)/d^6
```

Rubi [A] (verified)

Time = 0.19 (sec) , antiderivative size = 322, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}} = 0.031$, Rules used = {1634}

$$\int \frac{(a+bx)^2 (A+Bx+Cx^2+Dx^3)}{(c+dx)^{3/2}} \, dx = \frac{2(c+dx)^{3/2} (a^2d^2(Cd-3cD)-2abd(-Bd^2-6c^2D+3cCd)+}{3d^6}$$
$$+\frac{2(c+dx)^{5/2} (a^2d^2D+2abd(Cd-4cD)-(b^2(-Bd^2-10c^2D+4cCd)))}{5d^6}$$
$$+\frac{2\sqrt{c+dx}(bc-ad) (ad(-Bd^2-3c^2D+2cCd)-b(2Ad^3-3Bcd^2-5c^3D+4c^2Cd))}{d^6}$$
$$-\frac{2(bc-ad)^2 (Ad^3-Bcd^2+c^3(-D)+c^2Cd)}{d^6\sqrt{c+dx}}$$
$$+\frac{2b(c+dx)^{7/2}(2adD-5bcD+bCd)}{7d^6}+\frac{2b^2D(c+dx)^{9/2}}{9d^6}$$

```
[In] Int[((a + b*x)^2*(A + B*x + C*x^2 + D*x^3))/(c + d*x)^(3/2),x]
```

```
[Out] (-2*(b*c - a*d)^2*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D))/(d^6*Sqrt[c + d*x])
+ (2*(b*c - a*d)*(a*d*(2*c*C*d - B*d^2 - 3*c^2*D) - b*(4*c^2*C*d - 3*B*c*d^
2 + 2*A*d^3 - 5*c^3*D))*Sqrt[c + d*x])/d^6 + (2*(a^2*d^2*(C*d - 3*c*D) - 2*
a*b*d*(3*c*C*d - B*d^2 - 6*c^2*D) + b^2*(6*c^2*C*d - 3*B*c*d^2 + A*d^3 - 10
*c^3*D))*(c + d*x)^(3/2))/(3*d^6) + (2*(a^2*d^2*D + 2*a*b*d*(C*d - 4*c*D) -
 b^2*(4*c*C*d - B*d^2 - 10*c^2*D))*(c + d*x)^(5/2))/(5*d^6) + (2*b*(b*C*d -
 5*b*c*D + 2*a*d*D)*(c + d*x)^(7/2))/(7*d^6) + (2*b^2*D*(c + d*x)^(9/2))/(9
*d^6)
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

$$\text{integral} = \int \left(\frac{(-bc+ad)^2 (c^2Cd-Bcd^2+Ad^3-c^3D)}{d^5(c+dx)^{3/2}} + \frac{(bc-ad) (ad(2cCd-Bd^2-3c^2D)-b(4c^2Cd-3Bcd^2+2Ad^3-5c^3D))}{d^5\sqrt{c+dx}} + \frac{(a^2d^2(Cd-3cD)-2abd(3cCd-Bd^2-6c^2D)+b^2(6c^2Cd-3Bcd^2+Ad^3-10c^3D)) \sqrt{c+dx}}{d^5} + \frac{(a^2d^2D+2abd(Cd-4cD)-b^2(4cCd-Bd^2-10c^2D)) (c+dx)^{3/2}}{d^5} + \frac{b(bCd-5bcD+2adD)(c+dx)^{5/2}}{d^5}+\frac{b^2D(c+dx)^{7/2}}{d^5}\right) dx$$

$$= -\frac{2(bc-ad)^2\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{d^6\sqrt{c+dx}} + \frac{2(bc-ad)\left(ad(2cCd-Bd^2-3c^2D)-b(4c^2Cd-3Bcd^2+2Ad^3-5c^3D)\right)\sqrt{c+dx}}{d^6} + \frac{2(a^2d^2(Cd-3cD)-2abd(3cCd-Bd^2-6c^2D)+b^2(6c^2Cd-3Bcd^2+Ad^3-10c^3D))\,(c+dx)}{3d^6} + \frac{2(a^2d^2D+2abd(Cd-4cD)-b^2(4cCd-Bd^2-10c^2D))\,(c+dx)^{5/2}}{5d^6} + \frac{2b(bCd-5bcD+2adD)(c+dx)^{7/2}}{7d^6} + \frac{2b^2D(c+dx)^{9/2}}{9d^6}$$

Mathematica [A] (verified)

Time = 0.32 (sec) , antiderivative size = 326, normalized size of antiderivative = 1.01

$$\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx = \frac{2(21a^2d^2(48c^3D-8c^2d(5C-3Dx)+2cd^2(15B-x(10C+3Dx)}{}$$

```
[In] Integrate[((a + b*x)^2*(A + B*x + C*x^2 + D*x^3))/(c + d*x)^(3/2),x]
```

```
[Out] (2*(21*a^2*d^2*(48*c^3*D - 8*c^2*d*(5*C - 3*D*x) + 2*c*d^2*(15*B - x*(10*C
+ 3*D*x)) + d^3*(-15*A + x*(15*B + 5*C*x + 3*D*x^2))) + 6*a*b*d*(-384*c^4*D
 + 48*c^3*d*(7*C - 4*D*x) - 8*c^2*d^2*(35*B - 3*x*(7*C + 2*D*x)) + 2*c*d^3*
(105*A - x*(70*B + 3*x*(7*C + 4*D*x))) + d^4*x*(105*A + x*(35*B + 3*x*(7*C
+ 5*D*x)))) + b^2*(1280*c^5*D - 128*c^4*d*(9*C - 5*D*x) + 16*c^3*d^2*(63*B
- 2*x*(18*C + 5*D*x)) + 8*c^2*d^3*(-105*A + x*(63*B + 2*x*(9*C + 5*D*x))) +
 d^5*x^2*(105*A + x*(63*B + 5*x*(9*C + 7*D*x))) - 2*c*d^4*x*(210*A + x*(63*
B + x*(36*C + 25*D*x))))))/(315*d^6*Sqrt[c + d*x])
```

Maple [A] (verified)

Time = 1.74 (sec) , antiderivative size = 289, normalized size of antiderivative = 0.90

method	result
pseudoelliptic	$\frac{((70Dx^5+90Cx^4+126x^3B+210Ax^2)b^2+1260a(A+\frac{1}{7}Dx^3+\frac{1}{5}Cx^2+\frac{1}{3}Bx)xb-630a^2(-\frac{1}{5}Dx^3-\frac{1}{3}Cx^2-Bx+A))d^5+2}{}$
gosper	$-\frac{2(-35Db^2x^5d^5-45Cb^2d^5x^4-90Dabd^5x^4+50Db^2cd^4x^4-63Bb^2d^5x^3-126Cabd^5x^3+72Cb^2cd^4x^3-63Da^2d^5x^3+1}{}$
trager	$-\frac{2(-35Db^2x^5d^5-45Cb^2d^5x^4-90Dabd^5x^4+50Db^2cd^4x^4-63Bb^2d^5x^3-126Cabd^5x^3+72Cb^2cd^4x^3-63Da^2d^5x^3+1}{}$
derivativedivides	$\frac{\frac{2Db^2(dx+c)^{\frac{9}{2}}}{9}-\frac{2(a^2Ad^5-2Aabcd^4+Ab^2c^2d^3-Ba^2cd^4+2Babc^2d^3-Bb^2c^3d^2+Ca^2c^2d^3-2Cabc^3d^2+Cb^2c^4d-Da^2c^3d^2+2Da}{\sqrt{dx+c}}}{}$
default	$\frac{\frac{2Db^2(dx+c)^{\frac{9}{2}}}{9}-\frac{2(a^2Ad^5-2Aabcd^4+Ab^2c^2d^3-Ba^2cd^4+2Babc^2d^3-Bb^2c^3d^2+Ca^2c^2d^3-2Cabc^3d^2+Cb^2c^4d-Da^2c^3d^2+2Da}{\sqrt{dx+c}}}{}$

```
[In] int((b*x+a)^2*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x,method=_RETURNVERBOSE)
```

```
[Out] 1/315*(((70*D*x^5+90*C*x^4+126*B*x^3+210*A*x^2)*b^2+1260*a*(A+1/7*D*x^3+1/5
*C*x^2+1/3*B*x)*x*b-630*a^2*(-1/5*D*x^3-1/3*C*x^2-B*x+A))*d^5+2520*(-1/3*(5
/42*D*x^3+6/35*C*x^2+3/10*B*x+A)*x*b^2+a*(-4/35*D*x^3-1/5*C*x^2-2/3*B*x+A)*
b+1/2*a^2*(-1/5*D*x^2-2/3*C*x+B))*c*d^4-1680*c^2*((-2/21*D*x^3-6/35*C*x^2-3
/5*B*x+A)*b^2+2*a*(-6/35*D*x^2-3/5*C*x+B)*b+a^2*(-3/5*D*x+C))*d^3+2016*c^3*
((-10/63*D*x^2-4/7*C*x+B)*b^2+2*a*(-4/7*D*x+C)*b+D*a^2)*d^2-2304*((-5/9*D*x
+C)*b+2*D*a)*b*c^4*d+2560*D*b^2*c^5)/(d*x+c)^(1/2)/d^6
```

Fricas [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 419, normalized size of antiderivative = 1.30

$$\int \frac{(a+bx)^2\,(A+Bx+Cx^2+Dx^3)}{(c+dx)^{3/2}}\,dx = \frac{2\,(35\,Db^2d^5x^5+1280\,Db^2c^5-315\,Aa^2d^5-840\,(Ca^2+2\,Bab+}{}$$

```
[In] integrate((b*x+a)^2*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x, algorithm="fricas"
)
```

```
[Out] 2/315*(35*D*b^2*d^5*x^5 + 1280*D*b^2*c^5 - 315*A*a^2*d^5 - 840*(C*a^2 + 2*B
*a*b + A*b^2)*c^2*d^3 + 630*(B*a^2 + 2*A*a*b)*c*d^4 - 5*(10*D*b^2*c*d^4 - 9
*(2*D*a*b + C*b^2)*d^5)*x^4 + (80*D*b^2*c^2*d^3 + 63*(D*a^2 + 2*C*a*b + B*b
^2)*d^5 - 72*(2*D*a*b*c + C*b^2*c)*d^4)*x^3 + 1008*(D*a^2*c^3 + (2*C*a*b +
B*b^2)*c^3)*d^2 - (160*D*b^2*c^3*d^2 - 105*(C*a^2 + 2*B*a*b + A*b^2)*d^5 +
126*(D*a^2*c + (2*C*a*b + B*b^2)*c)*d^4 - 144*(2*D*a*b*c^2 + C*b^2*c^2)*d^3
)*x^2 - 1152*(2*D*a*b*c^4 + C*b^2*c^4)*d + (640*D*b^2*c^4*d - 420*(C*a^2 +
```

```
2*B*a*b + A*b^2)*c*d^4 + 315*(B*a^2 + 2*A*a*b)*d^5 + 504*(D*a^2*c^2 + (2*C*
a*b + B*b^2)*c^2)*d^3 - 576*(2*D*a*b*c^3 + C*b^2*c^3)*d^2)*x)*sqrt(d*x + c)
/(d^7*x + c*d^6)
```

Sympy [A] (verification not implemented)

Time = 27.76 (sec) , antiderivative size = 534, normalized size of antiderivative = 1.66

$$\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx = \begin{cases} \frac{2\left(\frac{Db^2(c+dx)^{\frac{9}{2}}}{9d^5}+\frac{(c+dx)^{\frac{7}{2}}\left(Cb^2d+2Dabd-5Db^2c\right)}{7d^5}+\frac{(c+dx)^{\frac{5}{2}}\left(Bb^2d^2+2Cabd^2-4Cb^2cd+Da^2\right.}{5d^5}\right.}{} \\ \frac{Aa^2x+\frac{Db^2x^6}{6}+\frac{x^5\left(Cb^2+2Dab\right)}{5}+\frac{x^4\left(Bb^2+2Cab+Da^2\right)}{4}+\frac{x^3\left(Ab^2+2Bab+Ca^2\right)}{3}+\frac{x^2\cdot\left(2Aab+\right.}{2}}{c^{\frac{3}{2}}} \end{cases}$$

```
[In] integrate((b*x+a)**2*(D*x**3+C*x**2+B*x+A)/(d*x+c)**(3/2),x)
```

```
[Out] Piecewise((2*(D*b**2*(c + d*x)**(9/2)/(9*d**5) + (c + d*x)**(7/2)*(C*b**2*d
 + 2*D*a*b*d - 5*D*b**2*c)/(7*d**5) + (c + d*x)**(5/2)*(B*b**2*d**2 + 2*C*a
*b*d**2 - 4*C*b**2*c*d + D*a**2*d**2 - 8*D*a*b*c*d + 10*D*b**2*c**2)/(5*d**
5) + (c + d*x)**(3/2)*(A*b**2*d**3 + 2*B*a*b*d**3 - 3*B*b**2*c*d**2 + C*a**
2*d**3 - 6*C*a*b*c*d**2 + 6*C*b**2*c**2*d - 3*D*a**2*c*d**2 + 12*D*a*b*c**2
*d - 10*D*b**2*c**3)/(3*d**5) + sqrt(c + d*x)*(2*A*a*b*d**4 - 2*A*b**2*c*d*
*3 + B*a**2*d**4 - 4*B*a*b*c*d**3 + 3*B*b**2*c**2*d**2 - 2*C*a**2*c*d**3 +
6*C*a*b*c**2*d**2 - 4*C*b**2*c**3*d + 3*D*a**2*c**2*d**2 - 8*D*a*b*c**3*d +
 5*D*b**2*c**4)/d**5 + (a*d - b*c)**2*(-A*d**3 + B*c*d**2 - C*c**2*d + D*c*
*3)/(d**5*sqrt(c + d*x)))/d, Ne(d, 0)), ((A*a**2*x + D*b**2*x**6/6 + x**5*(
C*b**2 + 2*D*a*b)/5 + x**4*(B*b**2 + 2*C*a*b + D*a**2)/4 + x**3*(A*b**2 + 2
*B*a*b + C*a**2)/3 + x**2*(2*A*a*b + B*a**2)/2)/c**(3/2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 395, normalized size of antiderivative = 1.23

$$\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx = \frac{2\left(\frac{35\,(dx+c)^{\frac{9}{2}}Db^2-45\,\left(5\,Db^2c-\left(2\,Dab+Cb^2\right)d\right)(dx+c)^{\frac{7}{2}}+63\,\left(10\,Db^2c^2-4\,\left(2\,Dab+Cb^2\right.\right.}{}\right.}{}$$

```
[In] integrate((b*x+a)^2*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x, algorithm="maxima"
)
```

```
[Out] 2/315*((35*(d*x + c)^(9/2)*D*b^2 - 45*(5*D*b^2*c - (2*D*a*b + C*b^2)*d)*(d*
x + c)^(7/2) + 63*(10*D*b^2*c^2 - 4*(2*D*a*b + C*b^2)*c*d + (D*a^2 + 2*C*a*
b + B*b^2)*d^2)*(d*x + c)^(5/2) - 105*(10*D*b^2*c^3 - 6*(2*D*a*b + C*b^2)*c
```

```
^2*d + 3*(D*a^2 + 2*C*a*b + B*b^2)*c*d^2 - (C*a^2 + 2*B*a*b + A*b^2)*d^3)*(
d*x + c)^(3/2) + 315*(5*D*b^2*c^4 - 4*(2*D*a*b + C*b^2)*c^3*d + 3*(D*a^2 +
2*C*a*b + B*b^2)*c^2*d^2 - 2*(C*a^2 + 2*B*a*b + A*b^2)*c*d^3 + (B*a^2 + 2*A
*a*b)*d^4)*sqrt(d*x + c))/d^5 + 315*(D*b^2*c^5 - A*a^2*d^5 - (2*D*a*b + C*b
^2)*c^4*d + (D*a^2 + 2*C*a*b + B*b^2)*c^3*d^2 - (C*a^2 + 2*B*a*b + A*b^2)*c
^2*d^3 + (B*a^2 + 2*A*a*b)*c*d^4)/(sqrt(d*x + c)*d^5))/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 651 vs. $2(302) = 604$.

Time = 0.30 (sec) , antiderivative size = 651, normalized size of antiderivative = 2.02

$$\int \frac{(a+bx)^2\,(A+Bx+Cx^2+Dx^3)}{(c+dx)^{3/2}}\,dx = \frac{2\left(Db^2c^5 - 2\,Dabc^4d - Cb^2c^4d + Da^2c^3d^2 + 2\,Cabc^3d^2 + Bb^2c^3d\right.}{\sqrt{dx}}$$
$$+\frac{2\left(35\,(dx+c)^{\frac{9}{2}}Db^2d^{48} - 225\,(dx+c)^{\frac{7}{2}}Db^2cd^{48} + 630\,(dx+c)^{\frac{5}{2}}Db^2c^2d^{48} - 1050\,(dx+c)^{\frac{3}{2}}Db^2c^3d^{48} + 157\right.}{}$$

```
[In] integrate((b*x+a)^2*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x, algorithm="giac")
```

```
[Out] 2*(D*b^2*c^5 - 2*D*a*b*c^4*d - C*b^2*c^4*d + D*a^2*c^3*d^2 + 2*C*a*b*c^3*d^
2 + B*b^2*c^3*d^2 - C*a^2*c^2*d^3 - 2*B*a*b*c^2*d^3 - A*b^2*c^2*d^3 + B*a^2
*c*d^4 + 2*A*a*b*c*d^4 - A*a^2*d^5)/(sqrt(d*x + c)*d^6) + 2/315*(35*(d*x +
c)^(9/2)*D*b^2*d^48 - 225*(d*x + c)^(7/2)*D*b^2*c*d^48 + 630*(d*x + c)^(5/2
)*D*b^2*c^2*d^48 - 1050*(d*x + c)^(3/2)*D*b^2*c^3*d^48 + 1575*sqrt(d*x + c)
*D*b^2*c^4*d^48 + 90*(d*x + c)^(7/2)*D*a*b*d^49 + 45*(d*x + c)^(7/2)*C*b^2*
d^49 - 504*(d*x + c)^(5/2)*D*a*b*c*d^49 - 252*(d*x + c)^(5/2)*C*b^2*c*d^49
+ 1260*(d*x + c)^(3/2)*D*a*b*c^2*d^49 + 630*(d*x + c)^(3/2)*C*b^2*c^2*d^49
- 2520*sqrt(d*x + c)*D*a*b*c^3*d^49 - 1260*sqrt(d*x + c)*C*b^2*c^3*d^49 + 6
3*(d*x + c)^(5/2)*D*a^2*d^50 + 126*(d*x + c)^(5/2)*C*a*b*d^50 + 63*(d*x + c
)^(5/2)*B*b^2*d^50 - 315*(d*x + c)^(3/2)*D*a^2*c*d^50 - 630*(d*x + c)^(3/2)
*C*a*b*c*d^50 - 315*(d*x + c)^(3/2)*B*b^2*c*d^50 + 945*sqrt(d*x + c)*D*a^2*
c^2*d^50 + 1890*sqrt(d*x + c)*C*a*b*c^2*d^50 + 945*sqrt(d*x + c)*B*b^2*c^2*
d^50 + 105*(d*x + c)^(3/2)*C*a^2*d^51 + 210*(d*x + c)^(3/2)*B*a*b*d^51 + 10
5*(d*x + c)^(3/2)*A*b^2*d^51 - 630*sqrt(d*x + c)*C*a^2*c*d^51 - 1260*sqrt(d
*x + c)*B*a*b*c*d^51 - 630*sqrt(d*x + c)*A*b^2*c*d^51 + 315*sqrt(d*x + c)*B
*a^2*d^52 + 630*sqrt(d*x + c)*A*a*b*d^52)/d^54
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+bx)^2 (A+Bx+Cx^2+Dx^3)}{(c+dx)^{3/2}}\,dx = \int \frac{(a+b\,x)^2\,(A+B\,x+C\,x^2+x^3\,D)}{(c+d\,x)^{3/2}}\,dx$$

```
[In] int(((a + b*x)^2*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(3/2),x)
[Out] int(((a + b*x)^2*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(3/2), x)
```

3.12 $\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx$

Optimal result

Integrand size = 30, antiderivative size = 210

$$
\begin{aligned}
\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx &= \frac{2(bc-ad)\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{d^5\sqrt{c+dx}} \\
&- \frac{2(ad(2cCd-Bd^2-3c^2D)-b(3c^2Cd-2Bcd^2+Ad^3-4c^3D))\sqrt{c+dx}}{d^5} \\
&+ \frac{2(ad(Cd-3cD)-b(3cCd-Bd^2-6c^2D))\,(c+dx)^{3/2}}{3d^5} \\
&+ \frac{2(bCd-4bcD+adD)(c+dx)^{5/2}}{5d^5} + \frac{2bD(c+dx)^{7/2}}{7d^5}
\end{aligned}
$$

```
[Out] 2/3*(a*d*(C*d-3*D*c)-b*(-B*d^2+3*C*c*d-6*D*c^2))*(d*x+c)^(3/2)/d^5+2/5*(C*b
*d+D*a*d-4*D*b*c)*(d*x+c)^(5/2)/d^5+2/7*b*D*(d*x+c)^(7/2)/d^5+2*(-a*d+b*c)*
(A*d^3-B*c*d^2+C*c^2*d-D*c^3)/d^5/(d*x+c)^(1/2)-2*(a*d*(-B*d^2+2*C*c*d-3*D*
c^2)-b*(A*d^3-2*B*c*d^2+3*C*c^2*d-4*D*c^3))*(d*x+c)^(1/2)/d^5
```

Rubi [A] (verified)

Time = 0.12 (sec) , antiderivative size = 210, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.033, Rules used

$= \{1634\}$

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx = -\frac{2\sqrt{c+dx}\left(ad\left(-Bd^2-3c^2D+2cCd\right)-b\left(Ad^3-2Bcd^2-4c^3D+3c^2Cd\right)\right)}{d^5} + \frac{2(bc-ad)\left(Ad^3-Bcd^2+c^3(-D)+c^2Cd\right)}{d^5\sqrt{c+dx}} + \frac{2(c+dx)^{3/2}\left(ad(Cd-3cD)-b\left(-Bd^2-6c^2D+3cCd\right)\right)}{3d^5} + \frac{2(c+dx)^{5/2}(adD-4bcD+bCd)}{5d^5} + \frac{2bD(c+dx)^{7/2}}{7d^5}$$

```
[In] Int[((a + b*x)*(A + B*x + C*x^2 + D*x^3))/(c + d*x)^(3/2),x]
```

```
[Out] (2*(b*c - a*d)*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D))/(d^5*Sqrt[c + d*x]) - (
2*(a*d*(2*c*C*d - B*d^2 - 3*c^2*D) - b*(3*c^2*C*d - 2*B*c*d^2 + A*d^3 - 4*c
^3*D))*Sqrt[c + d*x])/d^5 + (2*(a*d*(C*d - 3*c*D) - b*(3*c*C*d - B*d^2 - 6*
c^2*D))*(c + d*x)^(3/2))/(3*d^5) + (2*(b*C*d - 4*b*c*D + a*d*D)*(c + d*x)^(
5/2))/(5*d^5) + (2*b*D*(c + d*x)^(7/2))/(7*d^5)
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \left(\frac{(-bc+ad)\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{d^4(c+dx)^{3/2}} + \frac{-ad\left(2cCd-Bd^2-3c^2D\right)+b\left(3c^2Cd-2Bcd^2+Ad^3-4c^3D\right)}{d^4\sqrt{c+dx}} + \frac{\left(ad(Cd-3cD)-b\left(3cCd-Bd^2-6c^2D\right)\right)\sqrt{c+dx}}{d^4} + \frac{(bCd-4bcD+adD)(c+dx)^{3/2}}{d^4} + \frac{bD(c+dx)^{5/2}}{d^4}\right)dx \\ &= \frac{2(bc-ad)\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{d^5\sqrt{c+dx}} - \frac{2\left(ad\left(2cCd-Bd^2-3c^2D\right)-b\left(3c^2Cd-2Bcd^2+Ad^3-4c^3D\right)\right)\sqrt{c+dx}}{d^5} + \frac{2\left(ad(Cd-3cD)-b\left(3cCd-Bd^2-6c^2D\right)\right)(c+dx)^{3/2}}{3d^5} + \frac{2(bCd-4bcD+adD)(c+dx)^{5/2}}{5d^5} + \frac{2bD(c+dx)^{7/2}}{7d^5}\end{aligned}$$

Mathematica [A] (verified)

Time = 0.16 (sec) , antiderivative size = 188, normalized size of antiderivative = 0.90

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx = \frac{14ad(48c^3D-8c^2d(5C-3Dx)+2cd^2(15B-x(10C+3Dx))+}{}$$

```
[In] Integrate[((a + b*x)*(A + B*x + C*x^2 + D*x^3))/(c + d*x)^(3/2),x]
```

```
[Out] (14*a*d*(48*c^3*D - 8*c^2*d*(5*C - 3*D*x) + 2*c*d^2*(15*B - x*(10*C + 3*D*x
)) + d^3*(-15*A + x*(15*B + 5*C*x + 3*D*x^2))) + b*(-768*c^4*D + 96*c^3*d*(
7*C - 4*D*x) + 16*c^2*d^2*(-35*B + 3*x*(7*C + 2*D*x)) + 4*c*d^3*(105*A - x*
(70*B + 3*x*(7*C + 4*D*x))) + 2*d^4*x*(105*A + x*(35*B + 3*x*(7*C + 5*D*x))
)))/(105*d^5*Sqrt[c + d*x])
```

Maple [A] (verified)

Time = 1.68 (sec) , antiderivative size = 173, normalized size of antiderivative = 0.82

method	result
pseudoelliptic	$-\frac{2\left(\left(-\frac{Dbx^4}{7}+\frac{(-Cb-Da)x^3}{5}+\frac{(-Bb-Ca)x^2}{3}+(-Ab-Ba)x+Aa\right)d^4-2\left(-\frac{4Dbx^3}{35}+\frac{(-Cb-Da)x^2}{5}+\frac{2(-Bb-Ca)x}{3}+Ab+\right.\right.}{\sqrt{dx+c}\,d^5}$
gosper	$-\frac{2(-15Dbx^4d^4-21Cbd^4x^3-21Dad^4x^3+24Dbcd^3x^3-35Bbd^4x^2-35Cad^4x^2+42Cbcd^3x^2+42Dacd^3x^2-48Dbc^2d}{}$
trager	$-\frac{2(-15Dbx^4d^4-21Cbd^4x^3-21Dad^4x^3+24Dbcd^3x^3-35Bbd^4x^2-35Cad^4x^2+42Cbcd^3x^2+42Dacd^3x^2-48Dbc^2d}{}$
derivativedivides	$\frac{\frac{2Db(dx+c)^{\frac{7}{2}}}{7}+\frac{2Cbd(dx+c)^{\frac{5}{2}}}{5}+\frac{2Dad(dx+c)^{\frac{5}{2}}}{5}-\frac{8Dbc(dx+c)^{\frac{5}{2}}}{5}+\frac{2Bbd^2(dx+c)^{\frac{3}{2}}}{3}+\frac{2Cad^2(dx+c)^{\frac{3}{2}}}{3}-2Cbcd(dx+c)^{\frac{3}{2}}-2Dacd(}{}$
default	$\frac{\frac{2Db(dx+c)^{\frac{7}{2}}}{7}+\frac{2Cbd(dx+c)^{\frac{5}{2}}}{5}+\frac{2Dad(dx+c)^{\frac{5}{2}}}{5}-\frac{8Dbc(dx+c)^{\frac{5}{2}}}{5}+\frac{2Bbd^2(dx+c)^{\frac{3}{2}}}{3}+\frac{2Cad^2(dx+c)^{\frac{3}{2}}}{3}-2Cbcd(dx+c)^{\frac{3}{2}}-2Dacd(}{}$

```
[In] int((b*x+a)*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x,method=_RETURNVERBOSE)
```

```
[Out] -2/(d*x+c)^(1/2)*((-1/7*D*b*x^4+1/5*(-C*b-D*a)*x^3+1/3*(-B*b-C*a)*x^2+(-A*b
-B*a)*x+A*a)*d^4-2*(-4/35*D*b*x^3+1/5*(-C*b-D*a)*x^2+2/3*(-B*b-C*a)*x+A*b+B
*a)*c*d^3+8/3*c^2*(-6/35*D*b*x^2+3/5*(-C*b-D*a)*x+B*b+C*a)*d^2-16/5*c^3*(-4
/7*D*b*x+C*b+D*a)*d+128/35*D*b*c^4)/d^5
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 213, normalized size of antiderivative = 1.01

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx = \frac{2\left(15\,Dbd^4x^4 - 384\,Dbc^4 - 105\,Aad^4 - 280\,(Ca+Bb)c^2d^2 + 210\,(\right.}{}$$

```
[In] integrate((b*x+a)*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x, algorithm="fricas")
```

```
[Out] 2/105*(15*D*b*d^4*x^4 - 384*D*b*c^4 - 105*A*a*d^4 - 280*(C*a + B*b)*c^2*d^2
 + 210*(B*a + A*b)*c*d^3 - 3*(8*D*b*c*d^3 - 7*(D*a + C*b)*d^4)*x^3 + (48*D*
b*c^2*d^2 + 35*(C*a + B*b)*d^4 - 42*(D*a*c + C*b*c)*d^3)*x^2 + 336*(D*a*c^3
 + C*b*c^3)*d - (192*D*b*c^3*d + 140*(C*a + B*b)*c*d^3 - 105*(B*a + A*b)*d^
4 - 168*(D*a*c^2 + C*b*c^2)*d^2)*x)*sqrt(d*x + c)/(d^6*x + c*d^5)
```

Sympy [A] (verification not implemented)

Time = 7.84 (sec) , antiderivative size = 280, normalized size of antiderivative = 1.33

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx = \begin{cases} \frac{2\left(\frac{Db(c+dx)^{\frac{7}{2}}}{7d^4}+\frac{(c+dx)^{\frac{5}{2}}(Cbd+Dad-4Dbc)}{5d^4}+\frac{(c+dx)^{\frac{3}{2}}\left(Bbd^2+Cad^2-3Cbcd-3Dacd+6Dbc^2\right)}{3d^4}+\right.}{} \\ \frac{Aax+\frac{Dbx^5}{5}+\frac{x^4(Cb+Da)}{4}+\frac{x^3(Bb+Ca)}{3}+\frac{x^2(Ab+Ba)}{2}}{c^{\frac{3}{2}}} \end{cases}$$

```
[In] integrate((b*x+a)*(D*x**3+C*x**2+B*x+A)/(d*x+c)**(3/2),x)
```

```
[Out] Piecewise((2*(D*b*(c + d*x)**(7/2)/(7*d**4) + (c + d*x)**(5/2)*(C*b*d + D*a
*d - 4*D*b*c)/(5*d**4) + (c + d*x)**(3/2)*(B*b*d**2 + C*a*d**2 - 3*C*b*c*d
- 3*D*a*c*d + 6*D*b*c**2)/(3*d**4) + sqrt(c + d*x)*(A*b*d**3 + B*a*d**3 - 2
*B*b*c*d**2 - 2*C*a*c*d**2 + 3*C*b*c**2*d + 3*D*a*c**2*d - 4*D*b*c**3)/d**4
 + (a*d - b*c)*(-A*d**3 + B*c*d**2 - C*c**2*d + D*c**3)/(d**4*sqrt(c + d*x)
))/d, Ne(d, 0)), ((A*a*x + D*b*x**5/5 + x**4*(C*b + D*a)/4 + x**3*(B*b + C*
a)/3 + x**2*(A*b + B*a)/2)/c**(3/2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.21 (sec) , antiderivative size = 206, normalized size of antiderivative = 0.98

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx = \frac{2\left(\frac{15\,(dx+c)^{\frac{7}{2}}Db-21\,(4\,Dbc-(Da+Cb)d)(dx+c)^{\frac{5}{2}}+35\,\left(6\,Dbc^2-3\,(Da+Cb)cd+(Ca+Bb\right.}{d^4}\right.}{}$$

```
[In] integrate((b*x+a)*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x, algorithm="maxima")
```

```
[Out] 2/105*((15*(d*x + c)^(7/2)*D*b - 21*(4*D*b*c - (D*a + C*b)*d)*(d*x + c)^(5/
2) + 35*(6*D*b*c^2 - 3*(D*a + C*b)*c*d + (C*a + B*b)*d^2)*(d*x + c)^(3/2) -
 105*(4*D*b*c^3 - 3*(D*a + C*b)*c^2*d + 2*(C*a + B*b)*c*d^2 - (B*a + A*b)*d
^3)*sqrt(d*x + c))/d^4 - 105*(D*b*c^4 + A*a*d^4 - (D*a + C*b)*c^3*d + (C*a
+ B*b)*c^2*d^2 - (B*a + A*b)*c*d^3)/(sqrt(d*x + c)*d^4))/d
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 323, normalized size of antiderivative = 1.54

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx =$$

$$-\frac{2\left(Dbc^4-Dac^3d-Cbc^3d+Cac^2d^2+Bbc^2d^2-Bacd^3-Abcd^3+Aad^4\right)}{\sqrt{dx+c}d^5} + \frac{2\left(15\,(dx+c)^{\frac{7}{2}}Dbd^{30}-84\,(dx+c)^{\frac{5}{2}}Dbcd^{30}+210\,(dx+c)^{\frac{3}{2}}Dbc^2d^{30}-420\,\sqrt{dx+c}Dbc^3d^{30}+21\,(dx+c)\right.}{}$$

```
[In] integrate((b*x+a)*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x, algorithm="giac")
```

```
[Out] -2*(D*b*c^4 - D*a*c^3*d - C*b*c^3*d + C*a*c^2*d^2 + B*b*c^2*d^2 - B*a*c*d^3
 - A*b*c*d^3 + A*a*d^4)/(sqrt(d*x + c)*d^5) + 2/105*(15*(d*x + c)^(7/2)*D*b
*d^30 - 84*(d*x + c)^(5/2)*D*b*c*d^30 + 210*(d*x + c)^(3/2)*D*b*c^2*d^30 -
420*sqrt(d*x + c)*D*b*c^3*d^30 + 21*(d*x + c)^(5/2)*D*a*d^31 + 21*(d*x + c)
^(5/2)*C*b*d^31 - 105*(d*x + c)^(3/2)*D*a*c*d^31 - 105*(d*x + c)^(3/2)*C*b*
c*d^31 + 315*sqrt(d*x + c)*D*a*c^2*d^31 + 315*sqrt(d*x + c)*C*b*c^2*d^31 +
35*(d*x + c)^(3/2)*C*a*d^32 + 35*(d*x + c)^(3/2)*B*b*d^32 - 210*sqrt(d*x +
c)*C*a*c*d^32 - 210*sqrt(d*x + c)*B*b*c*d^32 + 105*sqrt(d*x + c)*B*a*d^33 +
 105*sqrt(d*x + c)*A*b*d^33)/d^35
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{3/2}}\,dx = \int \frac{(a+b\,x)\,\left(A+B\,x+C\,x^2+x^3\,D\right)}{{\left(c+d\,x\right)}^{3/2}} \,dx$$

```
[In] int(((a + b*x)*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(3/2),x)
```

```
[Out] int(((a + b*x)*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(3/2), x)
```

3.13 $\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{3/2}}\,dx$

Optimal result

Integrand size = 25, antiderivative size = 113

$$\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{3/2}}\,dx = -\frac{2(c^2Cd-Bcd^2+Ad^3-c^3D)}{d^4\sqrt{c+dx}} - \frac{2(2cCd-Bd^2-3c^2D)\sqrt{c+dx}}{d^4} + \frac{2(Cd-3cD)(c+dx)^{3/2}}{3d^4} + \frac{2D(c+dx)^{5/2}}{5d^4}$$

```
[Out] 2/3*(C*d-3*D*c)*(d*x+c)^(3/2)/d^4+2/5*D*(d*x+c)^(5/2)/d^4-2*(A*d^3-B*c*d^2+
C*c^2*d-D*c^3)/d^4/(d*x+c)^(1/2)-2*(-B*d^2+2*C*c*d-3*D*c^2)*(d*x+c)^(1/2)/d
^4
```

Rubi [A] (verified)

Time = 0.06 (sec) , antiderivative size = 113, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.040, Rules used = {1864}

$$\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{3/2}}\,dx = -\frac{2(Ad^3-Bcd^2+c^3(-D)+c^2Cd)}{d^4\sqrt{c+dx}} - \frac{2\sqrt{c+dx}(-Bd^2-3c^2D+2cCd)}{d^4} + \frac{2(c+dx)^{3/2}(Cd-3cD)}{3d^4} + \frac{2D(c+dx)^{5/2}}{5d^4}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/(c + d*x)^(3/2),x]
```

```
[Out] (-2*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D))/(d^4*Sqrt[c + d*x]) - (2*(2*c*C*d
- B*d^2 - 3*c^2*D)*Sqrt[c + d*x])/d^4 + (2*(C*d - 3*c*D)*(c + d*x)^(3/2))/(
3*d^4) + (2*D*(c + d*x)^(5/2))/(5*d^4)
```

Rule 1864

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_.))^(p_.), x_Symbol] :> Int[ExpandIntegrand
[Pq*(a + b*x^n)^p, x], x] /; FreeQ[{a, b, n}, x] && PolyQ[Pq, x] && (IGtQ[p
, 0] || EqQ[n, 1])
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \left(\frac{c^2Cd - Bcd^2 + Ad^3 - c^3D}{d^3(c+dx)^{3/2}} + \frac{-2cCd + Bd^2 + 3c^2D}{d^3\sqrt{c+dx}} + \frac{(Cd-3cD)\sqrt{c+dx}}{d^3} + \frac{D(c+dx)^{3/2}}{d^3}\right) dx \\ &= -\frac{2(c^2Cd - Bcd^2 + Ad^3 - c^3D)}{d^4\sqrt{c+dx}} - \frac{2(2cCd - Bd^2 - 3c^2D)\sqrt{c+dx}}{d^4} + \frac{2(Cd-3cD)(c+dx)^{3/2}}{3d^4} + \frac{2D(c+dx)^{5/2}}{5d^4}\end{aligned}$$

Mathematica [A] (verified)

Time = 0.06 (sec) , antiderivative size = 82, normalized size of antiderivative = 0.73

$$\int \frac{A + Bx + Cx^2 + Dx^3}{(c+dx)^{3/2}} dx = \frac{2(48c^3D - 8c^2d(5C - 3Dx) + 2cd^2(15B - x(10C + 3Dx)) + d^3(-15A + x(}{15d^4\sqrt{c+dx}}$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/(c + d*x)^(3/2),x]
[Out] (2*(48*c^3*D - 8*c^2*d*(5*C - 3*D*x) + 2*c*d^2*(15*B - x*(10*C + 3*D*x)) +
d^3*(-15*A + x*(15*B + 5*C*x + 3*D*x^2))))/(15*d^4*Sqrt[c + d*x])
```

Maple [A] (verified)

Time = 1.61 (sec) , antiderivative size = 74, normalized size of antiderivative = 0.65

method	result	si
pseudoelliptic	$\frac{\left(6D x^{3}+10C x^{2}+30B x -30A \right) d^{3}+60c \left(-\frac{1}{5}D x^{2}-\frac{2}{3}C x +B \right) d^{2}-80c^{2} \left(-\frac{3D x}{5}+C \right) d +96D c^{3}}{15\sqrt{d x +c}\, d^{4}}$	7
gosper	$-\frac{2\left(-3D x^{3} d^{3}-5C\, d^{3} x^{2}+6D c\, d^{2} x^{2}-15B\, d^{3} x +20C c\, d^{2} x -24D c^{2} d x +15A\, d^{3}-30B c\, d^{2}+40C\, c^{2} d -48D c^{3}\right)}{15\sqrt{d x +c}\, d^{4}}$	9
trager	$-\frac{2\left(-3D x^{3} d^{3}-5C\, d^{3} x^{2}+6D c\, d^{2} x^{2}-15B\, d^{3} x +20C c\, d^{2} x -24D c^{2} d x +15A\, d^{3}-30B c\, d^{2}+40C\, c^{2} d -48D c^{3}\right)}{15\sqrt{d x +c}\, d^{4}}$	9
derivativedivides	$\frac{\frac{2D \left(d x +c \right)^{\frac{5}{2}}}{5}+\frac{2C d \left(d x +c \right)^{\frac{3}{2}}}{3}-2D c \left(d x +c \right)^{\frac{3}{2}}+2B\, d^{2} \sqrt{d x +c}-4C c d \sqrt{d x +c}+6D c^{2} \sqrt{d x +c}-\frac{2\left(A\, d^{3}-B c\, d^{2}+C\, c^{2} d -D c^{3}\right)}{\sqrt{d x +c}}}{d^{4}}$	1
default	$\frac{\frac{2D \left(d x +c \right)^{\frac{5}{2}}}{5}+\frac{2C d \left(d x +c \right)^{\frac{3}{2}}}{3}-2D c \left(d x +c \right)^{\frac{3}{2}}+2B\, d^{2} \sqrt{d x +c}-4C c d \sqrt{d x +c}+6D c^{2} \sqrt{d x +c}-\frac{2\left(A\, d^{3}-B c\, d^{2}+C\, c^{2} d -D c^{3}\right)}{\sqrt{d x +c}}}{d^{4}}$	1

```
[In] int((D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x,method=_RETURNVERBOSE)
```

```
[Out] 1/15*((6*D*x^3+10*C*x^2+30*B*x-30*A)*d^3+60*c*(-1/5*D*x^2-2/3*C*x+B)*d^2-80
*c^2*(-3/5*D*x+C)*d+96*D*c^3)/(d*x+c)^(1/2)/d^4
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 100, normalized size of antiderivative = 0.88

$$\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{3/2}}\,dx = \frac{2\,(3\,Dd^3x^3+48\,Dc^3-40\,Cc^2d+30\,Bcd^2-15\,Ad^3-(6\,Dcd^2-5\,Cd^3)x^2+(}{15\,(d^5x+cd^4)}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x, algorithm="fricas")
```

```
[Out] 2/15*(3*D*d^3*x^3 + 48*D*c^3 - 40*C*c^2*d + 30*B*c*d^2 - 15*A*d^3 - (6*D*c*
d^2 - 5*C*d^3)*x^2 + (24*D*c^2*d - 20*C*c*d^2 + 15*B*d^3)*x)*sqrt(d*x + c)/
(d^5*x + c*d^4)
```

Sympy [A] (verification not implemented)

Time = 1.99 (sec) , antiderivative size = 144, normalized size of antiderivative = 1.27

$$\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{3/2}}\,dx = \begin{cases} \frac{2\left(\frac{D(c+dx)^{\frac{5}{2}}}{5d^3}+\frac{(c+dx)^{\frac{3}{2}}(Cd-3Dc)}{3d^3}+\frac{\sqrt{c+dx}\left(Bd^2-2Ccd+3Dc^2\right)}{d^3}+\frac{-Ad^3+Bcd^2-Cc^2d+Dc^3}{d^3\sqrt{c+dx}}\right)}{d} & \text{for } d \neq \\ \frac{Ax+\frac{Bx^2}{2}+\frac{Cx^3}{3}+\frac{Dx^4}{4}}{c^{\frac{3}{2}}} & \text{otherwis} \end{cases}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(d*x+c)**(3/2),x)
```

```
[Out] Piecewise((2*(D*(c + d*x)**(5/2)/(5*d**3) + (c + d*x)**(3/2)*(C*d - 3*D*c)/
(3*d**3) + sqrt(c + d*x)*(B*d**2 - 2*C*c*d + 3*D*c**2)/d**3 + (-A*d**3 + B*
c*d**2 - C*c**2*d + D*c**3)/(d**3*sqrt(c + d*x)))/d, Ne(d, 0)), ((A*x + B*x
**2/2 + C*x**3/3 + D*x**4/4)/c**(3/2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 102, normalized size of antiderivative = 0.90

$$\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{3/2}}\,dx = \frac{2\left(\frac{3\,(dx+c)^{\frac{5}{2}}D-5\,(3\,Dc-Cd)(dx+c)^{\frac{3}{2}}+15\,\left(3\,Dc^2-2\,Ccd+Bd^2\right)\sqrt{dx+c}}{d^3}+\frac{15\,(Dc^3-Cc^2d+Bcd^2-Ad^3}{\sqrt{dx+c}d^3}\right.}{15\,d}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x, algorithm="maxima")
```

```
[Out] 2/15*((3*(d*x + c)^(5/2)*D - 5*(3*D*c - C*d)*(d*x + c)^(3/2) + 15*(3*D*c^2
- 2*C*c*d + B*d^2)*sqrt(d*x + c))/d^3 + 15*(D*c^3 - C*c^2*d + B*c*d^2 - A*d
^3)/(sqrt(d*x + c)*d^3))/d
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 127, normalized size of antiderivative = 1.12

$$\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{3/2}}\,dx = \frac{2\,(Dc^3-Cc^2d+Bcd^2-Ad^3)}{\sqrt{dx+c}d^4} + \frac{2\Big(3\,(dx+c)^{\frac{5}{2}}Dd^{16}-15\,(dx+c)^{\frac{3}{2}}Dcd^{16}+45\,\sqrt{dx+c}Dc^2d^{16}+5\,(dx+c)^{\frac{3}{2}}Cd^{17}-30\,\sqrt{dx+c}Ccd^{17}+15}{15\,d^{20}}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(d*x+c)^(3/2),x, algorithm="giac")
```

```
[Out] 2*(D*c^3 - C*c^2*d + B*c*d^2 - A*d^3)/(sqrt(d*x + c)*d^4) + 2/15*(3*(d*x +
c)^(5/2)*D*d^16 - 15*(d*x + c)^(3/2)*D*c*d^16 + 45*sqrt(d*x + c)*D*c^2*d^16
 + 5*(d*x + c)^(3/2)*C*d^17 - 30*sqrt(d*x + c)*C*c*d^17 + 15*sqrt(d*x + c)*
B*d^18)/d^20
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{3/2}}\,dx = \int \frac{A+B\,x+C\,x^2+x^3\,D}{{\left(c+d\,x\right)}^{3/2}}\,dx$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/(c + d*x)^(3/2),x)
```

```
[Out] int((A + B*x + C*x^2 + x^3*D)/(c + d*x)^(3/2), x)
```

3.14 $\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{3/2}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 193

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{3/2}}\,dx=\frac{2(c^2Cd-Bcd^2+Ad^3-c^3D)}{d^3(bc-ad)\sqrt{c+dx}}-\frac{2cD\sqrt{c+dx}}{bd^3}+\frac{2(bCd-bcD-adD)\sqrt{c+dx}}{b^2d^3}+\frac{2D(c+dx)^{3/2}}{3bd^3}-\frac{2(Ab^3-a(b^2B-abC+a^2D))\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{5/2}(bc-ad)^{3/2}}$$

```
[Out] 2/3*D*(d*x+c)^(3/2)/b/d^3-2*(A*b^3-a*(B*b^2-C*a*b+D*a^2))*arctanh(b^(1/2)*(
d*x+c)^(1/2)/(-a*d+b*c)^(1/2))/b^(5/2)/(-a*d+b*c)^(3/2)+2*(A*d^3-B*c*d^2+C*
c^2*d-D*c^3)/d^3/(-a*d+b*c)/(d*x+c)^(1/2)-2*c*D*(d*x+c)^(1/2)/b/d^3+2*(C*b*
d-D*a*d-D*b*c)*(d*x+c)^(1/2)/b^2/d^3
```

Rubi [A] (verified)

Time = 0.17 (sec) , antiderivative size = 193, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {1633, 45, 65, 214}

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{3/2}}\,dx=-\frac{2(Ab^3-a(a^2D-abC+b^2B))\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{5/2}(bc-ad)^{3/2}}+\frac{2(Ad^3-Bcd^2+c^3(-D)+c^2Cd)}{d^3\sqrt{c+dx}(bc-ad)}+\frac{2\sqrt{c+dx}(-adD-bcD+bCd)}{b^2d^3}+\frac{2D(c+dx)^{3/2}}{3bd^3}-\frac{2cD\sqrt{c+dx}}{bd^3}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/((a + b*x)*(c + d*x)^(3/2)),x]
```

```
[Out] (2*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D))/(d^3*(b*c - a*d)*Sqrt[c + d*x]) - (
2*c*D*Sqrt[c + d*x])/(b*d^3) + (2*(b*C*d - b*c*D - a*d*D)*Sqrt[c + d*x])/(b
^2*d^3) + (2*D*(c + d*x)^(3/2))/(3*b*d^3) - (2*(A*b^3 - a*(b^2*B - a*b*C +
a^2*D))*ArcTanh[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[b*c - a*d]])/(b^(5/2)*(b*c - a
*d)^(3/2))
```

Rule 45

```
Int[((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol] :> Int
[ExpandIntegrand[(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c, d, n},
x] && NeQ[b*c - a*d, 0] && IGtQ[m, 0] && ( !IntegerQ[n] || (EqQ[c, 0] && Le
Q[7*m + 4*n + 4, 0]) || LtQ[9*m + 5*(n + 1), 0] || GtQ[m + n + 2, 0])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 1633

```
Int[((Px_)*((c_.) + (d_.)*(x_))^(n_.))/((a_.) + (b_.)*(x_)), x_Symbol] :> I
nt[ExpandIntegrand[1/Sqrt[c + d*x], Px*((c + d*x)^(n + 1/2)/(a + b*x)), x],
 x] /; FreeQ[{a, b, c, d, n}, x] && PolyQ[Px, x] && ILtQ[n + 1/2, 0] && GtQ
[Expon[Px, x], 2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \left(\frac{c^2Cd - Bcd^2 + Ad^3 - c^3D}{d^2(-bc+ad)(c+dx)^{3/2}} + \frac{bCd - bcD - adD}{b^2d^2\sqrt{c+dx}} + \frac{Dx}{bd\sqrt{c+dx}} \right. \\
&\qquad \left. + \frac{Ab^3 - a(b^2B - abC + a^2D)}{b^2(bc-ad)(a+bx)\sqrt{c+dx}}\right) dx \\
&= \frac{2(c^2Cd - Bcd^2 + Ad^3 - c^3D)}{d^3(bc-ad)\sqrt{c+dx}} + \frac{2(bCd - bcD - adD)\sqrt{c+dx}}{b^2d^3} \\
&\quad + \frac{D\int \frac{x}{\sqrt{c+dx}}\,dx}{bd} + \frac{(Ab^3 - a(b^2B - abC + a^2D))\int \frac{1}{(a+bx)\sqrt{c+dx}}\,dx}{b^2(bc-ad)}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{2(c^2Cd - Bcd^2 + Ad^3 - c^3D)}{d^3(bc-ad)\sqrt{c+dx}} + \frac{2(bCd - bcD - adD)\sqrt{c+dx}}{b^2d^3} \\
&\quad + \frac{D\int \left(-\frac{c}{d\sqrt{c+dx}} + \frac{\sqrt{c+dx}}{d}\right) dx}{bd} \\
&\quad + \frac{(2(Ab^3 - a(b^2B - abC + a^2D)))\,\text{Subst}\left(\int \frac{1}{a-\frac{bc}{d}+\frac{bx^2}{d}}\,dx, x, \sqrt{c+dx}\right)}{b^2d(bc-ad)} \\
&= \frac{2(c^2Cd - Bcd^2 + Ad^3 - c^3D)}{d^3(bc-ad)\sqrt{c+dx}} - \frac{2cD\sqrt{c+dx}}{bd^3} + \frac{2(bCd - bcD - adD)\sqrt{c+dx}}{b^2d^3} \\
&\quad + \frac{2D(c+dx)^{3/2}}{3bd^3} - \frac{2(Ab^3 - a(b^2B - abC + a^2D))\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{5/2}(bc-ad)^{3/2}}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.35 (sec) , antiderivative size = 191, normalized size of antiderivative = 0.99

$$
\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{3/2}}\,dx = \frac{-6a^2d^2D(c+dx) + 2abd(c+dx)(3Cd - 2cD + dDx) + 2b^2(-3Ad^3 + 8c^3D}{3b^2d^3(-bc+ad)\sqrt{c+dx}}
$$

$$
- \frac{2(Ab^3 - a(b^2B - abC + a^2D))\arctan\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{-bc+ad}}\right)}{b^{5/2}(-bc+ad)^{3/2}}
$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/((a + b*x)*(c + d*x)^(3/2)),x]
```

```
[Out] (-6*a^2*d^2*D*(c + d*x) + 2*a*b*d*(c + d*x)*(3*C*d - 2*c*D + d*D*x) + 2*b^2
*(-3*A*d^3 + 8*c^3*D + c^2*(-6*C*d + 4*d*D*x) + c*d^2*(3*B - x*(3*C + D*x))
))/(3*b^2*d^3*(-(b*c) + a*d)*Sqrt[c + d*x]) - (2*(A*b^3 - a*(b^2*B - a*b*C
+ a^2*D))*ArcTan[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[-(b*c) + a*d]])/(b^(5/2)*(-(b
*c) + a*d)^(3/2))
```

Maple [A] (verified)

Time = 1.72 (sec) , antiderivative size = 161, normalized size of antiderivative = 0.83

method	result
pseudoelliptic	$\frac{\frac{2\sqrt{dx+c}\,(Dbdx+3Cbd-3Dad-5Dbc)}{3b^2}-\frac{2\left(A\,d^3-Bc\,d^2+C\,c^2d-Dc^3\right)}{(ad-bc)\sqrt{dx+c}}-\frac{2d^3\left(b^3A-a\,b^2B+C\,a^2b-Da^3\right)\arctan\left(\frac{b\sqrt{dx+c}}{\sqrt{(ad-bc)b}}\right)}{(ad-bc)b^2\sqrt{(ad-bc)b}}}{d^3}$
derivativedivides	$\frac{\frac{2\left(\frac{D(dx+c)^{\frac{3}{2}}b}{3}+dbC\sqrt{dx+c}-Dad\sqrt{dx+c}-2Dcb\sqrt{dx+c}\right)}{b^2}-\frac{2\left(A\,d^3-Bc\,d^2+C\,c^2d-Dc^3\right)}{(ad-bc)\sqrt{dx+c}}-\frac{2d^3\left(b^3A-a\,b^2B+C\,a^2b-Da^3\right)\arctan(}{(ad-bc)b^2\sqrt{(ad-bc)b}}}{d^3}$
default	$\frac{\frac{2\left(\frac{D(dx+c)^{\frac{3}{2}}b}{3}+dbC\sqrt{dx+c}-Dad\sqrt{dx+c}-2Dcb\sqrt{dx+c}\right)}{b^2}-\frac{2\left(A\,d^3-Bc\,d^2+C\,c^2d-Dc^3\right)}{(ad-bc)\sqrt{dx+c}}-\frac{2d^3\left(b^3A-a\,b^2B+C\,a^2b-Da^3\right)\arctan(}{(ad-bc)b^2\sqrt{(ad-bc)b}}}{d^3}$

```
[In] int((D*x^3+C*x^2+B*x+A)/(b*x+a)/(d*x+c)^(3/2),x,method=_RETURNVERBOSE)
```

```
[Out] 2*(1/3*(d*x+c)^(1/2)*(D*b*d*x+3*C*b*d-3*D*a*d-5*D*b*c)/b^2-(A*d^3-B*c*d^2+C
*c^2*d-D*c^3)/(a*d-b*c)/(d*x+c)^(1/2)-d^3*(A*b^3-B*a*b^2+C*a^2*b-D*a^3)/(a*
d-b*c)/b^2/((a*d-b*c)*b)^(1/2)*arctan(b*(d*x+c)^(1/2)/((a*d-b*c)*b)^(1/2)))
/d^3
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 426 vs. $2(172) = 344$.

Time = 0.33 (sec) , antiderivative size = 866, normalized size of antiderivative = 4.49

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{3/2}}\,dx = \left[-\frac{3\left((Da^3-Ca^2b+Bab^2-Ab^3)d^4x+(Da^3c-(Ca^2b-Bab^2+Ab^3)c)d^3\right.}{}\right.$$

$$-\frac{2\left(3\left((Da^3-Ca^2b+Bab^2-Ab^3)d^4x+(Da^3c-(Ca^2b-Bab^2+Ab^3)c)d^3\right)\sqrt{-b^2c+abd}\,\arctan\left(\frac{\sqrt{-b^2c+}}{bdx}\right.\right.}{}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)/(d*x+c)^(3/2),x, algorithm="fricas")
```

```
[Out] [-1/3*(3*((D*a^3 - C*a^2*b + B*a*b^2 - A*b^3)*d^4*x + (D*a^3*c - (C*a^2*b -
 B*a*b^2 + A*b^3)*c)*d^3)*sqrt(b^2*c - a*b*d)*log((b*d*x + 2*b*c - a*d - 2*
sqrt(b^2*c - a*b*d)*sqrt(d*x + c))/(b*x + a)) + 2*(8*D*b^4*c^4 + 3*A*a*b^3*
d^4 + 3*(D*a^3*b*c - (C*a^2*b^2 + B*a*b^3 + A*b^4)*c)*d^3 - (D*a^2*b^2*c^2
 - 3*(3*C*a*b^3 + B*b^4)*c^2)*d^2 - (D*b^4*c^2*d^2 - 2*D*a*b^3*c*d^3 + D*a^2
*b^2*d^4)*x^2 - 2*(5*D*a*b^3*c^3 + 3*C*b^4*c^3)*d + (4*D*b^4*c^3*d + 3*(D*a
^3*b - C*a^2*b^2)*d^4 - 2*(D*a^2*b^2*c - 3*C*a*b^3*c)*d^3 - (5*D*a*b^3*c^2
+ 3*C*b^4*c^2)*d^2)*x)*sqrt(d*x + c))/(b^5*c^3*d^3 - 2*a*b^4*c^2*d^4 + a^2*
b^3*c*d^5 + (b^5*c^2*d^4 - 2*a*b^4*c*d^5 + a^2*b^3*d^6)*x), -2/3*(3*((D*a^3
 - C*a^2*b + B*a*b^2 - A*b^3)*d^4*x + (D*a^3*c - (C*a^2*b - B*a*b^2 + A*b^3
)*c)*d^3)*sqrt(-b^2*c + a*b*d)*arctan(sqrt(-b^2*c + a*b*d)*sqrt(d*x + c)/(b
*d*x + b*c)) + (8*D*b^4*c^4 + 3*A*a*b^3*d^4 + 3*(D*a^3*b*c - (C*a^2*b^2 + B
```

```
*a*b^3 + A*b^4)*c)*d^3 - (D*a^2*b^2*c^2 - 3*(3*C*a*b^3 + B*b^4)*c^2)*d^2 -
(D*b^4*c^2*d^2 - 2*D*a*b^3*c*d^3 + D*a^2*b^2*d^4)*x^2 - 2*(5*D*a*b^3*c^3 +
3*C*b^4*c^3)*d + (4*D*b^4*c^3*d + 3*(D*a^3*b - C*a^2*b^2)*d^4 - 2*(D*a^2*b^
2*c - 3*C*a*b^3*c)*d^3 - (5*D*a*b^3*c^2 + 3*C*b^4*c^2)*d^2)*x)*sqrt(d*x + c
))/(b^5*c^3*d^3 - 2*a*b^4*c^2*d^4 + a^2*b^3*c*d^5 + (b^5*c^2*d^4 - 2*a*b^4*
c*d^5 + a^2*b^3*d^6)*x)]
```

Sympy [A] (verification not implemented)

Time = 8.48 (sec) , antiderivative size = 262, normalized size of antiderivative = 1.36

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{3/2}}\,dx=\begin{cases} \frac{2\left(\frac{D(c+dx)^{\frac{3}{2}}}{3bd^2}+\frac{-Ad^3+Bcd^2-Cc^2d+Dc^3}{d^2\sqrt{c+dx}(ad-bc)}+\frac{\sqrt{c+dx}(Cbd-Dad-2Dbc)}{b^2d^2}+\frac{d\left(-Ab^3+Bab^2-Ca^2b+Da^3\right)\operatorname{atan}\left(\right.}{b^3\sqrt{\frac{ad-bc}{b}}(ad-bc)}\right.}{d} \\ \frac{\frac{Dx^3}{3b}+\frac{x^2(Cb-Da)}{2b^2}+\frac{x\left(Bb^2-Cab+Da^2\right)}{b^3}-\frac{\left(-Ab^3+Bab^2-Ca^2b+Da^3\right)\left(\begin{cases}\frac{x}{a} & \text{for}\ b=0\\ \frac{\log{(a+bx)}}{b} & \text{otherwise}\end{cases}\right)}{b^3}}{c^{\frac{3}{2}}} \end{cases}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(b*x+a)/(d*x+c)**(3/2),x)
```

```
[Out] Piecewise((2*(D*(c + d*x)**(3/2)/(3*b*d**2) + (-A*d**3 + B*c*d**2 - C*c**2*
d + D*c**3)/(d**2*sqrt(c + d*x)*(a*d - b*c)) + sqrt(c + d*x)*(C*b*d - D*a*d
 - 2*D*b*c)/(b**2*d**2) + d*(-A*b**3 + B*a*b**2 - C*a**2*b + D*a**3)*atan(s
qrt(c + d*x)/sqrt((a*d - b*c)/b))/(b**3*sqrt((a*d - b*c)/b)*(a*d - b*c)))/d
, Ne(d, 0)), ((D*x**3/(3*b) + x**2*(C*b - D*a)/(2*b**2) + x*(B*b**2 - C*a*b
 + D*a**2)/b**3 - (-A*b**3 + B*a*b**2 - C*a**2*b + D*a**3)*Piecewise((x/a,
Eq(b, 0)), (log(a + b*x)/b, True))/b**3)/c**(3/2), True))
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{3/2}}\,dx=\text{Exception raised: ValueError}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)/(d*x+c)^(3/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 200, normalized size of antiderivative = 1.04

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{3/2}}\,dx = -\frac{2\left(Da^3-Ca^2b+Bab^2-Ab^3\right)\arctan\left(\frac{\sqrt{dx+c}b}{\sqrt{-b^2c+abd}}\right)}{(b^3c-ab^2d)\sqrt{-b^2c+abd}} - \frac{2\left(Dc^3-Cc^2d+Bcd^2-Ad^3\right)}{(bcd^3-ad^4)\sqrt{dx+c}} + \frac{2\left((dx+c)^{\frac{3}{2}}Db^2d^6-6\sqrt{dx+c}Db^2cd^6-3\sqrt{dx+c}Dabd^7+3\sqrt{dx+c}Cb^2d^7\right)}{3\,b^3d^9}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)/(d*x+c)^(3/2),x, algorithm="giac")
```

```
[Out] -2*(D*a^3 - C*a^2*b + B*a*b^2 - A*b^3)*arctan(sqrt(d*x + c)*b/sqrt(-b^2*c +
 a*b*d))/((b^3*c - a*b^2*d)*sqrt(-b^2*c + a*b*d)) - 2*(D*c^3 - C*c^2*d + B*
c*d^2 - A*d^3)/((b*c*d^3 - a*d^4)*sqrt(d*x + c)) + 2/3*((d*x + c)^(3/2)*D*b
^2*d^6 - 6*sqrt(d*x + c)*D*b^2*c*d^6 - 3*sqrt(d*x + c)*D*a*b*d^7 + 3*sqrt(d
*x + c)*C*b^2*d^7)/(b^3*d^9)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{3/2}}\,dx = \int \frac{A+B\,x+C\,x^2+x^3\,D}{(a+b\,x)\,\left(c+d\,x\right)^{3/2}}\,dx$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)*(c + d*x)^(3/2)),x)
```

```
[Out] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)*(c + d*x)^(3/2)), x)
```

3.15 $\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{3/2}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 253

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{3/2}}\,dx=\frac{ab^2Bd^3-a^2bCd^3+a^3d^3D-b^3(2c^2Cd-2Bcd^2+3Ad^3-2c^3D)}{b^3d^2(bc-ad)^2\sqrt{c+dx}}$$
$$-\frac{A-\frac{a(b^2B-abC+a^2D)}{b^3}}{(bc-ad)(a+bx)\sqrt{c+dx}}+\frac{2D\sqrt{c+dx}}{b^2d^2}$$
$$-\frac{(b^3(2Bc-3Ad)-ab^2(4cC-Bd)-3a^3dD+a^2b(Cd+6cD))\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{5/2}(bc-ad)^{5/2}}$$

```
[Out] -(b^3*(-3*A*d+2*B*c)-a*b^2*(-B*d+4*C*c)-3*a^3*d*D+a^2*b*(C*d+6*D*c))*arctan
h(b^(1/2)*(d*x+c)^(1/2)/(-a*d+b*c)^(1/2))/b^(5/2)/(-a*d+b*c)^(5/2)+(a*b^2*B
*d^3-a^2*b*C*d^3+a^3*d^3*D-b^3*(3*A*d^3-2*B*c*d^2+2*C*c^2*d-2*D*c^3))/b^3/d
^2/(-a*d+b*c)^2/(d*x+c)^(1/2)+(-A+a*(B*b^2-C*a*b+D*a^2)/b^3)/(-a*d+b*c)/(b*
x+a)/(d*x+c)^(1/2)+2*D*(d*x+c)^(1/2)/b^2/d^2
```

Rubi [A] (verified)

Time = 0.39 (sec) , antiderivative size = 253, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {1635, 911, 1275, 214}

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{3/2}}\,dx=-\frac{A-\frac{a(a^2D-abC+b^2B)}{b^3}}{(a+bx)\sqrt{c+dx}(bc-ad)}$$
$$-\frac{\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)(-3a^3dD+a^2b(6cD+Cd)-ab^2(4cC-Bd)+b^3(2Bc-3Ad))}{b^{5/2}(bc-ad)^{5/2}}$$
$$+\frac{a^3d^3D-a^2bCd^3+ab^2Bd^3-(b^3(3Ad^3-2Bcd^2-2c^3D+2c^2Cd))}{b^3d^2\sqrt{c+dx}(bc-ad)^2}+\frac{2D\sqrt{c+dx}}{b^2d^2}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^2*(c + d*x)^(3/2)),x]
```

```
[Out] (a*b^2*B*d^3 - a^2*b*C*d^3 + a^3*d^3*D - b^3*(2*c^2*C*d - 2*B*c*d^2 + 3*A*d
^3 - 2*c^3*D))/(b^3*d^2*(b*c - a*d)^2*Sqrt[c + d*x]) - (A - (a*(b^2*B - a*b
*C + a^2*D))/b^3)/((b*c - a*d)*(a + b*x)*Sqrt[c + d*x]) + (2*D*Sqrt[c + d*x
])/(b^2*d^2) - ((b^3*(2*B*c - 3*A*d) - a*b^2*(4*c*C - B*d) - 3*a^3*d*D + a^
2*b*(C*d + 6*c*D))*ArcTanh[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[b*c - a*d]])/(b^(5/
2)*(b*c - a*d)^(5/2))
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 911

```
Int[((d_.) + (e_.)*(x_))^(m_)*((f_.) + (g_.)*(x_))^(n_)*((a_.) + (b_.)*(x_)
 + (c_.)*(x_)^2)^(p_.), x_Symbol] :> With[{q = Denominator[m]}, Dist[q/e, S
ubst[Int[x^(q*(m + 1) - 1)*((e*f - d*g)/e + g*(x^q/e))^n*((c*d^2 - b*d*e +
a*e^2)/e^2 - (2*c*d - b*e)*(x^q/e^2) + c*(x^(2*q)/e^2))^p, x], x, (d + e*x)
^(1/q)], x]] /; FreeQ[{a, b, c, d, e, f, g}, x] && NeQ[e*f - d*g, 0] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IntegersQ[n, p] && Fra
ctionQ[m]
```

Rule 1275

```
Int[((f_.)*(x_))^(m_.)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (
c_.)*(x_)^4)^(p_.), x_Symbol] :> Int[ExpandIntegrand[(f*x)^m*(d + e*x^2)^q*
(a + b*x^2 + c*x^4)^p, x], x] /; FreeQ[{a, b, c, d, e, f, m, q}, x] && NeQ[
b^2 - 4*a*c, 0] && IGtQ[p, 0] && IGtQ[q, -2]
```

Rule 1635

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol] :
> With[{Qx = PolynomialQuotient[Px, a + b*x, x], R = PolynomialRemainder[Px
, a + b*x, x]}, Simp[R*(a + b*x)^(m + 1)*((c + d*x)^(n + 1)/((m + 1)*(b*c -
 a*d))), x] + Dist[1/((m + 1)*(b*c - a*d)), Int[(a + b*x)^(m + 1)*(c + d*x)
^n*ExpandToSum[(m + 1)*(b*c - a*d)*Qx - d*R*(m + n + 2), x], x], x]] /; Fre
eQ[{a, b, c, d, n}, x] && PolyQ[Px, x] && ILtQ[m, -1] && GtQ[Expon[Px, x],
2]
```

Rubi steps

$$\text{integral} = -\frac{A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}}{(bc-ad)(a+bx)\sqrt{c+dx}} + \frac{\int \frac{-\frac{b^3(2Bc-3Ad)-ab^2(2cC-Bd)+a^3dD-a^2b(Cd-2cD)}{2b^3}-\frac{(bc-ad)(bC-aD)x}{b^2}-\left(c-\frac{ad}{b}\right)Dx^2}{(a+bx)(c+dx)^{3/2}}\,dx}{-bc+ad}$$

$$= -\frac{A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}}{(bc-ad)(a+bx)\sqrt{c+dx)}} - \frac{2\text{Subst}\left(\int \frac{\frac{-c^2\left(c-\frac{ad}{b}\right)D+\frac{cd(bc-ad)(bC-aD)}{b^2}-\frac{d^2\left(b^3(2Bc-3Ad)-ab^2(2cC-Bd)+a^3dD-a^2b(Cd-2cD)\right)}{2b^3}}{d^2}-\frac{\left(-2c\left(c-\frac{ad}{b}\right)D+\frac{d(bc-ad)(bC}{b^2}\right.}{d^2}}{x^2\left(\frac{-bc+ad}{d}+\frac{bx^2}{d}\right)}\right.}{d(bc-ad)}$$

$$= -\frac{A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}}{(bc-ad)(a+bx)\sqrt{c+dx}} - \frac{2\text{Subst}\left(\int\left(-\frac{(bc-ad)D}{b^2d}+\frac{ab^2Bd^3-a^2bCd^3+a^3d^3D-b^3\left(2c^2Cd-2Bcd^2+3Ad^3-2c^3D\right)}{2b^3d(bc-ad)x^2}+\frac{d\left(b^3(2Bc-3Ad)-ab^2(4cC-Bd)-\right.}{2b^2(bc-ad)(bc-}\right.\right.}{d(bc-ad)}$$

$$= \frac{ab^2Bd^3-a^2bCd^3+a^3d^3D-b^3(2c^2Cd-2Bcd^2+3Ad^3-2c^3D)}{b^3d^2(bc-ad)^2\sqrt{c+dx}} - \frac{A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}}{(bc-ad)(a+bx)\sqrt{c+dx}} + \frac{2D\sqrt{c+dx}}{b^2d^2} - \frac{(b^3(2Bc-3Ad)-ab^2(4cC-Bd)-3a^3dD+a^2b(Cd+6cD))\,\text{Subst}\left(\int\frac{1}{bc-ad-bx^2}\,dx,x,\sqrt{c+dx}\right.}{b^2(bc-ad)^2}$$

$$= \frac{ab^2Bd^3-a^2bCd^3+a^3d^3D-b^3(2c^2Cd-2Bcd^2+3Ad^3-2c^3D)}{b^3d^2(bc-ad)^2\sqrt{c+dx}} - \frac{A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}}{(bc-ad)(a+bx)\sqrt{c+dx}} + \frac{2D\sqrt{c+dx}}{b^2d^2} - \frac{(b^3(2Bc-3Ad)-ab^2(4cC-Bd)-3a^3dD+a^2b(Cd+6cD))\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{5/2}(bc-ad)^{5/2}}$$

Mathematica [A] (verified)

Time = 0.65 (sec) , antiderivative size = 259, normalized size of antiderivative = 1.02

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{3/2}}\,dx = \frac{3a^3d^2D(c+dx)+a^2bd(c+dx)(-Cd-4cD+2dDx)+b^3(-Ad^2(c+3dx)+}{b^2} + \frac{(b^3(2Bc-3Ad)+ab^2(-4cC+Bd)-3a^3dD+a^2b(Cd+6cD))\arctan\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{-bc+ad}}\right)}{b^{5/2}(-bc+ad)^{5/2}}$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^2*(c + d*x)^(3/2)),x]
```

```
[Out] (3*a^3*d^2*D*(c + d*x) + a^2*b*d*(c + d*x)*(-(C*d) - 4*c*D + 2*d*D*x) + b^3
*(-(A*d^2*(c + 3*d*x)) + 2*c*x*(-(c*C*d) + B*d^2 + 2*c^2*D + c*d*D*x)) + a*
b^2*(4*c^3*D + d^3*(-2*A + B*x) - 2*c^2*d*(C + D*x) + c*d^2*(3*B - 4*D*x^2)
```

```
))/(b^2*d^2*(b*c - a*d)^2*(a + b*x)*Sqrt[c + d*x]) + ((b^3*(2*B*c - 3*A*d)
+ a*b^2*(-4*c*C + B*d) - 3*a^3*d*D + a^2*b*(C*d + 6*c*D))*ArcTan[(Sqrt[b]*S
qrt[c + d*x])/Sqrt[-(b*c) + a*d]])/(b^(5/2)*(-(b*c) + a*d)^(5/2))
```

Maple [A] (verified)

Time = 1.83 (sec) , antiderivative size = 228, normalized size of antiderivative = 0.90

method	result
derivativedivides	$\frac{\frac{2D\sqrt{dx+c}}{b^2}-\frac{2d^2\left(\frac{\left(\frac{1}{2}Ab^3d-\frac{1}{2}Bab^2d+\frac{1}{2}Ca^2bd-\frac{1}{2}a^3dD\right)\sqrt{dx+c}}{(dx+c)b+ad-bc}+\frac{\left(3Ab^3d-Bab^2d-2Bb^3c-Ca^2bd+4Cab^2c+3a^3dD-6Da^2bc\right)}{2\sqrt{(ad-bc)b}}\right.}{(ad-bc)^2b^2}}{d^2}$
default	$\frac{\frac{2D\sqrt{dx+c}}{b^2}-\frac{2d^2\left(\frac{\left(\frac{1}{2}Ab^3d-\frac{1}{2}Bab^2d+\frac{1}{2}Ca^2bd-\frac{1}{2}a^3dD\right)\sqrt{dx+c}}{(dx+c)b+ad-bc}+\frac{\left(3Ab^3d-Bab^2d-2Bb^3c-Ca^2bd+4Cab^2c+3a^3dD-6Da^2bc\right)}{2\sqrt{(ad-bc)b}}\right.}{(ad-bc)^2b^2}}{d^2}$
pseudoelliptic	$-\frac{3\left(\left(\left(b^3A-\frac{1}{3}ab^2B-\frac{1}{3}Ca^2b+Da^3\right)d-\frac{2bc\left(Bb^2-2Cab+3Da^2\right)}{3}\right)\arctan\left(\frac{b\sqrt{dx+c}}{\sqrt{(ad-bc)b}}\right)(bx+a)d^2\sqrt{dx+c}+\frac{2\sqrt{(ad-bc)b}\left(\right.}{}\right.}{}$

```
[In] int((D*x^3+C*x^2+B*x+A)/(b*x+a)^2/(d*x+c)^(3/2),x,method=_RETURNVERBOSE)
```

```
[Out] 2/d^2*(D/b^2*(d*x+c)^(1/2)-(A*d^3-B*c*d^2+C*c^2*d-D*c^3)/(a*d-b*c)^2/(d*x+c
)^(1/2)-d^2/(a*d-b*c)^2/b^2*((1/2*A*b^3*d-1/2*B*a*b^2*d+1/2*C*a^2*b*d-1/2*a
^3*d*D)*(d*x+c)^(1/2)/((d*x+c)*b+a*d-b*c)+1/2*(3*A*b^3*d-B*a*b^2*d-2*B*b^3*
c-C*a^2*b*d+4*C*a*b^2*c+3*D*a^3*d-6*D*a^2*b*c)/((a*d-b*c)*b)^(1/2)*arctan(b
*(d*x+c)^(1/2)/((a*d-b*c)*b)^(1/2))))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 785 vs. 2(236) = 472.

Time = 0.33 (sec) , antiderivative size = 1583, normalized size of antiderivative = 6.26

$$\int\frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{3/2}}\,dx=\text{Too large to display}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^2/(d*x+c)^(3/2),x, algorithm="fricas"
)
```

```
[Out] [1/2*(((3*D*a^4*c - (C*a^3*b + B*a^2*b^2 - 3*A*a*b^3)*c)*d^3 - 2*(3*D*a^3*b
*c^2 - (2*C*a^2*b^2 - B*a*b^3)*c^2)*d^2 + ((3*D*a^3*b - C*a^2*b^2 - B*a*b^3
 + 3*A*b^4)*d^4 - 2*(3*D*a^2*b^2*c - (2*C*a*b^3 - B*b^4)*c)*d^3)*x^2 + ((3*
```

```
D*a^4 - C*a^3*b - B*a^2*b^2 + 3*A*a*b^3)*d^4 - 3*(D*a^3*b*c - (C*a^2*b^2 -
B*a*b^3 + A*b^4)*c)*d^3 - 2*(3*D*a^2*b^2*c^2 - (2*C*a*b^3 - B*b^4)*c^2)*d^2
)*x)*sqrt(b^2*c - a*b*d)*log((b*d*x + 2*b*c - a*d + 2*sqrt(b^2*c - a*b*d)*s
qrt(d*x + c))/(b*x + a)) + 2*(4*D*a*b^4*c^4 + 2*A*a^2*b^3*d^4 - (3*D*a^4*b*
c - (C*a^3*b^2 - 3*B*a^2*b^3 - A*a*b^4)*c)*d^3 + (7*D*a^3*b^2*c^2 + (C*a^2*
b^3 + 3*B*a*b^4 - A*b^5)*c^2)*d^2 + 2*(D*b^5*c^3*d - 3*D*a*b^4*c^2*d^2 + 3*
D*a^2*b^3*c*d^3 - D*a^3*b^2*d^4)*x^2 - 2*(4*D*a^2*b^3*c^3 + C*a*b^4*c^3)*d
+ (4*D*b^5*c^4 + 2*(C*a*b^4 + B*b^5)*c^2*d^2 - (3*D*a^4*b - C*a^3*b^2 + B*a
^2*b^3 - 3*A*a*b^4)*d^4 + (5*D*a^3*b^2*c - (C*a^2*b^3 + B*a*b^4 + 3*A*b^5)*
c)*d^3 - 2*(3*D*a*b^4*c^3 + C*b^5*c^3)*d)*x)*sqrt(d*x + c))/(a*b^6*c^4*d^2
- 3*a^2*b^5*c^3*d^3 + 3*a^3*b^4*c^2*d^4 - a^4*b^3*c*d^5 + (b^7*c^3*d^3 - 3*
a*b^6*c^2*d^4 + 3*a^2*b^5*c*d^5 - a^3*b^4*d^6)*x^2 + (b^7*c^4*d^2 - 2*a*b^6
*c^3*d^3 + 2*a^3*b^4*c*d^5 - a^4*b^3*d^6)*x), -(((3*D*a^4*c - (C*a^3*b + B*
a^2*b^2 - 3*A*a*b^3)*c)*d^3 - 2*(3*D*a^3*b*c^2 - (2*C*a^2*b^2 - B*a*b^3)*c^
2)*d^2 + ((3*D*a^3*b - C*a^2*b^2 - B*a*b^3 + 3*A*b^4)*d^4 - 2*(3*D*a^2*b^2*
c - (2*C*a*b^3 - B*b^4)*c)*d^3)*x^2 + ((3*D*a^4 - C*a^3*b - B*a^2*b^2 + 3*A
*a*b^3)*d^4 - 3*(D*a^3*b*c - (C*a^2*b^2 - B*a*b^3 + A*b^4)*c)*d^3 - 2*(3*D*
a^2*b^2*c^2 - (2*C*a*b^3 - B*b^4)*c^2)*d^2)*x)*sqrt(-b^2*c + a*b*d)*arctan(
sqrt(-b^2*c + a*b*d)*sqrt(d*x + c)/(b*d*x + b*c)) - (4*D*a*b^4*c^4 + 2*A*a^
2*b^3*d^4 - (3*D*a^4*b*c - (C*a^3*b^2 - 3*B*a^2*b^3 - A*a*b^4)*c)*d^3 + (7*
D*a^3*b^2*c^2 + (C*a^2*b^3 + 3*B*a*b^4 - A*b^5)*c^2)*d^2 + 2*(D*b^5*c^3*d -
 3*D*a*b^4*c^2*d^2 + 3*D*a^2*b^3*c*d^3 - D*a^3*b^2*d^4)*x^2 - 2*(4*D*a^2*b^
3*c^3 + C*a*b^4*c^3)*d + (4*D*b^5*c^4 + 2*(C*a*b^4 + B*b^5)*c^2*d^2 - (3*D*
a^4*b - C*a^3*b^2 + B*a^2*b^3 - 3*A*a*b^4)*d^4 + (5*D*a^3*b^2*c - (C*a^2*b^
3 + B*a*b^4 + 3*A*b^5)*c)*d^3 - 2*(3*D*a*b^4*c^3 + C*b^5*c^3)*d)*x)*sqrt(d*
x + c))/(a*b^6*c^4*d^2 - 3*a^2*b^5*c^3*d^3 + 3*a^3*b^4*c^2*d^4 - a^4*b^3*c*
d^5 + (b^7*c^3*d^3 - 3*a*b^6*c^2*d^4 + 3*a^2*b^5*c*d^5 - a^3*b^4*d^6)*x^2 +
 (b^7*c^4*d^2 - 2*a*b^6*c^3*d^3 + 2*a^3*b^4*c*d^5 - a^4*b^3*d^6)*x)]
```

Sympy [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{3/2}}\,dx=\text{Timed out}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(b*x+a)**2/(d*x+c)**(3/2),x)
[Out] Timed out
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{3/2}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^2/(d*x+c)^(3/2),x, algorithm="maxima"
)
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 388, normalized size of antiderivative = 1.53

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{3/2}}\,dx = \frac{(6\,Da^2bc-4\,Cab^2c+2\,Bb^3c-3\,Da^3d+Ca^2bd+Bab^2d-3\,Ab^3d)\arctan\Bigl(}{(b^4c^2-2\,ab^3cd+a^2b^2d^2)\sqrt{-b^2c+abd}}$$

$$+\frac{2\,(dx+c)Db^3c^3-2\,Db^3c^4-2\,(dx+c)Cb^3c^2d+2\,Dab^2c^3d+2\,Cb^3c^3d+2\,(dx+c)Bb^3cd^2-2\,Cab^2c^2d^2-}{(b^4c^2d^2-2\,ab^3cd^3+a^2b^2d^4}$$

$$+\frac{2\sqrt{dx+c}D}{b^2d^2}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^2/(d*x+c)^(3/2),x, algorithm="giac")
```

```
[Out] (6*D*a^2*b*c - 4*C*a*b^2*c + 2*B*b^3*c - 3*D*a^3*d + C*a^2*b*d + B*a*b^2*d
- 3*A*b^3*d)*arctan(sqrt(d*x + c)*b/sqrt(-b^2*c + a*b*d))/((b^4*c^2 - 2*a*b
^3*c*d + a^2*b^2*d^2)*sqrt(-b^2*c + a*b*d)) + (2*(d*x + c)*D*b^3*c^3 - 2*D*
b^3*c^4 - 2*(d*x + c)*C*b^3*c^2*d + 2*D*a*b^2*c^3*d + 2*C*b^3*c^3*d + 2*(d*
x + c)*B*b^3*c*d^2 - 2*C*a*b^2*c^2*d^2 - 2*B*b^3*c^2*d^2 + (d*x + c)*D*a^3*
d^3 - (d*x + c)*C*a^2*b*d^3 + (d*x + c)*B*a*b^2*d^3 - 3*(d*x + c)*A*b^3*d^3
 + 2*B*a*b^2*c*d^3 + 2*A*b^3*c*d^3 - 2*A*a*b^2*d^4)/((b^4*c^2*d^2 - 2*a*b^3
*c*d^3 + a^2*b^2*d^4)*((d*x + c)^(3/2)*b - sqrt(d*x + c)*b*c + sqrt(d*x + c
)*a*d)) + 2*sqrt(d*x + c)*D/(b^2*d^2)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{3/2}}\,dx = \int \frac{A+B\,x+C\,x^2+x^3\,D}{{\left(a+b\,x\right)}^2\,{\left(c+d\,x\right)}^{3/2}} \,d x$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^2*(c + d*x)^(3/2)),x)
[Out] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^2*(c + d*x)^(3/2)), x)
```

3.16 $\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{3/2}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 350

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{3/2}}\,dx =$$

$$-\frac{ab^2Bd^3-a^2bCd^3+a^3d^3D-b^3(4c^2Cd-4Bcd^2+5Ad^3-4c^3D)}{2b^3d(bc-ad)^3\sqrt{c+dx}}$$

$$-\frac{Ab^3-a(b^2B-abC+a^2D)}{2b^3(bc-ad)(a+bx)^2\sqrt{c+dx}}$$

$$-\frac{(b^3(4Bc-5Ad)-ab^2(8cC-Bd)-7a^3dD+3a^2b(Cd+4cD))\sqrt{c+dx}}{4b^2(bc-ad)^3(a+bx)}$$

$$-\frac{(b^3(8c^2C-12Bcd+15Ad^2)-3a^3d^2D-a^2bd(Cd-12cD)+ab^2(8cCd-3Bd^2-24c^2D))\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c}}{\sqrt{bc-}}\right.}{4b^{5/2}(bc-ad)^{7/2}}$$

```
[Out] -1/4*(b^3*(15*A*d^2-12*B*c*d+8*C*c^2)-3*a^3*d^2*D-a^2*b*d*(C*d-12*D*c)+a*b^
2*(-3*B*d^2+8*C*c*d-24*D*c^2))*arctanh(b^(1/2)*(d*x+c)^(1/2)/(-a*d+b*c)^(1/
2))/b^(5/2)/(-a*d+b*c)^(7/2)+1/2*(-a*b^2*B*d^3+a^2*b*C*d^3-a^3*d^3*D+b^3*(5
*A*d^3-4*B*c*d^2+4*C*c^2*d-4*D*c^3))/b^3/d/(-a*d+b*c)^3/(d*x+c)^(1/2)+1/2*(
-A*b^3+a*(B*b^2-C*a*b+D*a^2))/b^3/(-a*d+b*c)/(b*x+a)^2/(d*x+c)^(1/2)-1/4*(b
^3*(-5*A*d+4*B*c)-a*b^2*(-B*d+8*C*c)-7*a^3*d*D+3*a^2*b*(C*d+4*D*c))*(d*x+c)
^(1/2)/b^2/(-a*d+b*c)^3/(b*x+a)
```

Rubi [A] (verified)

Time = 0.54 (sec) , antiderivative size = 350, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}} = 0.156$, Rules used = {1635, 911, 1273, 464, 214}

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{3/2}}\,dx = -\frac{Ab^3-a(a^2D-abC+b^2B)}{2b^3(a+bx)^2\sqrt{c+dx}(bc-ad)}$$

$$-\frac{\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)(-3a^3d^2D-a^2bd(Cd-12cD)+ab^2(-3Bd^2-24c^2D+8cCd)+b^3(15Ad^2-12Bcd+}{4b^{5/2}(bc-ad)^{7/2}}$$

$$-\frac{a^3d^3D-a^2bCd^3+ab^2Bd^3-(b^3(5Ad^3-4Bcd^2-4c^3D+4c^2Cd))}{2b^3d\sqrt{c+dx}(bc-ad)^3}$$

$$-\frac{\sqrt{c+dx}(-7a^3dD+3a^2b(4cD+Cd)-ab^2(8cC-Bd)+b^3(4Bc-5Ad))}{4b^2(a+bx)(bc-ad)^3}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^3*(c + d*x)^(3/2)),x]
```

```
[Out] -1/2*(a*b^2*B*d^3 - a^2*b*C*d^3 + a^3*d^3*D - b^3*(4*c^2*C*d - 4*B*c*d^2 +
5*A*d^3 - 4*c^3*D))/(b^3*d*(b*c - a*d)^3*Sqrt[c + d*x]) - (A*b^3 - a*(b^2*B
 - a*b*C + a^2*D))/(2*b^3*(b*c - a*d)*(a + b*x)^2*Sqrt[c + d*x]) - ((b^3*(4
*B*c - 5*A*d) - a*b^2*(8*c*C - B*d) - 7*a^3*d*D + 3*a^2*b*(C*d + 4*c*D))*Sq
rt[c + d*x])/(4*b^2*(b*c - a*d)^3*(a + b*x)) - ((b^3*(8*c^2*C - 12*B*c*d +
15*A*d^2) - 3*a^3*d^2*D - a^2*b*d*(C*d - 12*c*D) + a*b^2*(8*c*C*d - 3*B*d^2
 - 24*c^2*D))*ArcTanh[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[b*c - a*d]])/(4*b^(5/2)*
(b*c - a*d)^(7/2))
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 464

```
Int[((e_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_.)*((c_) + (d_.)*(x_)^(n
_)), x_Symbol] :> Simp[c*(e*x)^(m + 1)*((a + b*x^n)^(p + 1)/(a*e*(m + 1))),
 x] + Dist[(a*d*(m + 1) - b*c*(m + n*(p + 1) + 1))/(a*e^n*(m + 1)), Int[(e*
x)^(m + n)*(a + b*x^n)^p, x], x] /; FreeQ[{a, b, c, d, e, p}, x] && NeQ[b*c
 - a*d, 0] && (IntegerQ[n] || GtQ[e, 0]) && ((GtQ[n, 0] && LtQ[m, -1]) || (
LtQ[n, 0] && GtQ[m + n, -1])) &&  !ILtQ[p, -1]
```

Rule 911

```
Int[((d_.) + (e_.)*(x_))^(m_)*((f_.) + (g_.)*(x_))^(n_)*((a_.) + (b_.)*(x_)
 + (c_.)*(x_)^2)^(p_.), x_Symbol] :> With[{q = Denominator[m]}, Dist[q/e, S
ubst[Int[x^(q*(m + 1) - 1)*((e*f - d*g)/e + g*(x^q/e))^n*((c*d^2 - b*d*e +
```

```
a*e^2)/e^2 - (2*c*d - b*e)*(x^q/e^2) + c*(x^(2*q)/e^2))^p, x], x, (d + e*x)
^(1/q)], x]] /; FreeQ[{a, b, c, d, e, f, g}, x] && NeQ[e*f - d*g, 0] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IntegersQ[n, p] && Fra
ctionQ[m]
```

Rule 1273

```
Int[(x_)^(m_)*((d_) + (e_.)*(x_)^2)^(q_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^
4)^(p_.), x_Symbol] :> Simp[(-d)^(m/2 - 1)*(c*d^2 - b*d*e + a*e^2)^p*x*((d
+ e*x^2)^(q + 1)/(2*e^(2*p + m/2)*(q + 1))), x] + Dist[(-d)^(m/2 - 1)/(2*e^
(2*p)*(q + 1)), Int[x^m*(d + e*x^2)^(q + 1)*ExpandToSum[Together[(1/(d + e*
x^2))*(2*(-d)^(-m/2 + 1)*e^(2*p)*(q + 1)*(a + b*x^2 + c*x^4)^p - ((c*d^2 -
b*d*e + a*e^2)^p/(e^(m/2)*x^m))*(d + e*(2*q + 3)*x^2))], x], x], x] /; Free
Q[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && IGtQ[p, 0] && ILtQ[q, -1] &
& ILtQ[m/2, 0]
```

Rule 1635

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol] :
> With[{Qx = PolynomialQuotient[Px, a + b*x, x], R = PolynomialRemainder[Px
, a + b*x, x]}, Simp[R*(a + b*x)^(m + 1)*((c + d*x)^(n + 1)/((m + 1)*(b*c -
 a*d))), x] + Dist[1/((m + 1)*(b*c - a*d)), Int[(a + b*x)^(m + 1)*(c + d*x)
^n*ExpandToSum[(m + 1)*(b*c - a*d)*Qx - d*R*(m + n + 2), x], x], x]] /; Fre
eQ[{a, b, c, d, n}, x] && PolyQ[Px, x] && ILtQ[m, -1] && GtQ[Expon[Px, x],
2]
```

Rubi steps

$$\text{integral} = -\frac{Ab^3-a(b^2B-abC+a^2D)}{2b^3(bc-ad)(a+bx)^2\sqrt{c+dx}} - \frac{\int \frac{-\frac{b^3(4Bc-5Ad)-ab^2(4cC-Bd)+a^3dD-a^2b(Cd-4cD)}{2b^3}-\frac{2(bc-ad)(bC-aD)x}{b^2}-2\left(c-\frac{ad}{b}\right)Dx^2}{(a+bx)^2(c+dx)^{3/2}}\,dx}{2(bc-ad)}$$

$$= -\frac{Ab^3-a(b^2B-abC+a^2D)}{2b^3(bc-ad)(a+bx)^2\sqrt{c+dx}} - \frac{\text{Subst}\left(\int \frac{\frac{-2c^2\left(c-\frac{ad}{b}\right)D+\frac{2cd(bc-ad)(bC-aD)}{b^2}-\frac{d^2\left(b^3(4Bc-5Ad)-ab^2(4cC-Bd)+a^3dD-a^2b(Cd-4cD)\right)}{2b^3}}{d^2}-\frac{\left(-4c\left(c-\frac{ad}{b}\right)D+\frac{2d(bc-ad}{b}\right.}{d^2}}{x^2\left(\frac{-bc+ad}{d}+\frac{bx^2}{d}\right)^2}\right.}{d(bc-ad)}$$

$$= -\frac{Ab^3 - a(b^2B - abC + a^2D)}{2b^3(bc-ad)(a+bx)^2\sqrt{c+dx}}$$
$$-\frac{\left(b^3(4Bc-5Ad) - ab^2(8cC-Bd) - 7a^3dD + 3a^2b(Cd+4cD)\right)\sqrt{c+dx}}{4b^2(bc-ad)^3(a+bx)}$$
$$+\frac{d^3\text{Subst}\left(\int \frac{-\frac{(bc-ad)\left(ab^2Bd^3 - a^2bCd^3 + a^3d^3D - b^3\left(4c^2Cd - 4Bcd^2 + 5Ad^3 - 4c^3D\right)\right)}{bd^5} - \frac{\left(a^3d^3D + 3a^2bd^2(Cd-4cD) - ab^2d\left(8cCd - Bd^2 - 24c\right.\right.}{2d^5}}{x^2\left(\frac{-bc+ad}{d} + \frac{bx^2}{d}\right)}\right.}{2b^2(bc-ad)^3}$$

$$= -\frac{ab^2Bd^3 - a^2bCd^3 + a^3d^3D - b^3(4c^2Cd - 4Bcd^2 + 5Ad^3 - 4c^3D)}{2b^3d(bc-ad)^3\sqrt{c+dx}}$$
$$-\frac{Ab^3 - a(b^2B - abC + a^2D)}{2b^3(bc-ad)(a+bx)^2\sqrt{c+dx}}$$
$$-\frac{\left(b^3(4Bc-5Ad) - ab^2(8cC-Bd) - 7a^3dD + 3a^2b(Cd+4cD)\right)\sqrt{c+dx}}{4b^2(bc-ad)^3(a+bx)}$$
$$+\frac{\left(b^3(8c^2C - 12Bcd + 15Ad^2) - 3a^3d^2D - a^2bd(Cd - 12cD) + ab^2(8cCd - 3Bd^2 - 24c^2D)\right)\text{Subs}}{4b^2d(bc-ad)^3}$$

$$= -\frac{ab^2Bd^3 - a^2bCd^3 + a^3d^3D - b^3(4c^2Cd - 4Bcd^2 + 5Ad^3 - 4c^3D)}{2b^3d(bc-ad)^3\sqrt{c+dx}}$$
$$-\frac{Ab^3 - a(b^2B - abC + a^2D)}{2b^3(bc-ad)(a+bx)^2\sqrt{c+dx}}$$
$$-\frac{\left(b^3(4Bc-5Ad) - ab^2(8cC-Bd) - 7a^3dD + 3a^2b(Cd+4cD)\right)\sqrt{c+dx}}{4b^2(bc-ad)^3(a+bx)}$$
$$-\frac{\left(b^3(8c^2C - 12Bcd + 15Ad^2) - 3a^3d^2D - a^2bd(Cd - 12cD) + ab^2(8cCd - 3Bd^2 - 24c^2D)\right)\tanh}{4b^{5/2}(bc-ad)^{7/2}}$$

Mathematica [A] (verified)

Time = 1.38 (sec) , antiderivative size = 387, normalized size of antiderivative = 1.11

$$\int \frac{A + Bx + Cx^2 + Dx^3}{(a+bx)^3(c+dx)^{3/2}}\,dx = \frac{\sqrt{b}\left(-3a^4d^2D(c+dx) - a^3bd(c+dx)(Cd + 5D(-2c+dx)) + 4b^4cx(2c(-Cd+cD)x + Bd(c+3dx)) + ab^3\left(-8cx(3\right.\right.}{}$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^3*(c + d*x)^(3/2)),x]
```

```
[Out] ((Sqrt[b]*(-3*a^4*d^2*D*(c + d*x) - a^3*b*d*(c + d*x)*(C*d + 5*D*(-2*c + d*
x)) + 4*b^4*c*x*(2*c*(-(C*d) + c*D)*x + B*d*(c + 3*d*x)) + a*b^3*(-8*c*x*(3
*c*C*d - 2*c^2*D + C*d^2*x) + B*d*(2*c^2 + 21*c*d*x + 3*d^2*x^2)) - A*b^2*d
*(8*a^2*d^2 + a*b*d*(9*c + 25*d*x) + b^2*(-2*c^2 + 5*c*d*x + 15*d^2*x^2)) +
 a^2*b^2*(8*c^3*D + d^3*x*(5*B + C*x) - 2*c^2*d*(7*C - 6*D*x) + c*d^2*(13*B
```

```
 - 5*C*x + 12*D*x^2))))/(d*(-(b*c) + a*d)^3*(a + b*x)^2*Sqrt[c + d*x]) - ((
b^3*(8*c^2*C - 12*B*c*d + 15*A*d^2) - 3*a^3*d^2*D + a^2*b*d*(-(C*d) + 12*c*
D) + a*b^2*(8*c*C*d - 3*B*d^2 - 24*c^2*D))*ArcTan[(Sqrt[b]*Sqrt[c + d*x])/S
qrt[-(b*c) + a*d]])/(-(b*c) + a*d)^(7/2))/(4*b^(5/2))
```

Maple [A] (verified)

Time = 1.96 (sec) , antiderivative size = 379, normalized size of antiderivative = 1.08

method	result
pseudoelliptic	$-\frac{15\left(\left(\left(A d^{2}-\frac{4}{5} B c d+\frac{8}{15} C c^{2}\right) b^{3}-\frac{a\left(B d^{2}-\frac{8}{3} C c d+8 D c^{2}\right) b^{2}}{5}-\frac{a^{2} b d(C d-12 D c)}{15}-\frac{a^{3} d^{2} D}{5}\right) \sqrt{d x+c}\,(b x+a)^{2} d \arctan\left(\frac{b \sqrt{d}}{\sqrt{(a d}}\right.\right.}{}$
derivativedivides	$-\frac{2 d\left(\frac{d\left(7 A b^{3} d-3 B a b^{2} d-4 B b^{3} c-C a^{2} b d+8 C a b^{2} c+5 a^{3} d D-12 D a^{2} b c\right)(d x+c)^{\frac{3}{2}}}{8 b}+\frac{d\left(9 A a b^{3} d^{2}-9 A b^{4} c d-5 B a^{2} b^{2} d^{2}+B a b^{3} c d+4 B\right.}{((d x+c) b+a d-b c)^{2}}\right.}{}$
default	$-\frac{2 d\left(\frac{d\left(7 A b^{3} d-3 B a b^{2} d-4 B b^{3} c-C a^{2} b d+8 C a b^{2} c+5 a^{3} d D-12 D a^{2} b c\right)(d x+c)^{\frac{3}{2}}}{8 b}+\frac{d\left(9 A a b^{3} d^{2}-9 A b^{4} c d-5 B a^{2} b^{2} d^{2}+B a b^{3} c d+4 B\right.}{((d x+c) b+a d-b c)^{2}}\right.}{}$

```
[In] int((D*x^3+C*x^2+B*x+A)/(b*x+a)^3/(d*x+c)^(3/2),x,method=_RETURNVERBOSE)
```

```
[Out] -15/4/((a*d-b*c)*b)^(1/2)*(((A*d^2-4/5*B*c*d+8/15*C*c^2)*b^3-1/5*a*(B*d^2-8
/3*C*c*d+8*D*c^2)*b^2-1/15*a^2*b*d*(C*d-12*D*c)-1/5*a^3*d^2*D)*(d*x+c)^(1/2
)*(b*x+a)^2*d*arctan(b*(d*x+c)^(1/2)/((a*d-b*c)*b)^(1/2))+8/15*((a*d-b*c)*b
)^(1/2)*((15/8*A*d^3*x^2+5/8*(-12/5*B*x+A)*x*c*d^2-1/4*c^2*(-4*C*x^2+2*B*x+
A)*d-D*c^3*x^2)*b^4+9/8*a*((-1/3*x^2*B+25/9*A*x)*d^3+c*(8/9*C*x^2-7/3*B*x+A
)*d^2-2/9*c^2*(-12*C*x+B)*d-16/9*D*c^3*x)*b^3+a^2*((-5/8*B*x+A-1/8*C*x^2)*d
^3-13/8*c*(12/13*D*x^2-5/13*C*x+B)*d^2+7/4*c^2*(-6/7*D*x+C)*d-D*c^3)*b^2+1/
8*a^3*(d*x+c)*((5*D*x+C)*d-10*D*c)*d*b+3/8*D*a^4*d^2*(d*x+c)))/(d*x+c)^(1/2
)/(a*d-b*c)^3/(b*x+a)^2/b^2/d
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1290 vs. $2(327) = 654$.

Time $= 0.39$ (sec) , antiderivative size $= 2594$, normalized size of antiderivative $= 7.41$

$$\int \frac{A + Bx + Cx^2 + Dx^3}{(a + bx)^3(c + dx)^{3/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^3/(d*x+c)^(3/2),x, algorithm="fricas"
)
```

```
[Out] [-1/8*(((3*D*a^5*c + (C*a^4*b + 3*B*a^3*b^2 - 15*A*a^2*b^3)*c)*d^3 + ((3*D*
a^3*b^2 + C*a^2*b^3 + 3*B*a*b^4 - 15*A*b^5)*d^4 - 4*(3*D*a^2*b^3*c + (2*C*a
*b^4 - 3*B*b^5)*c)*d^3 + 8*(3*D*a*b^4*c^2 - C*b^5*c^2)*d^2)*x^3 - 4*(3*D*a^
4*b*c^2 + (2*C*a^3*b^2 - 3*B*a^2*b^3)*c^2)*d^2 + (2*(3*D*a^4*b + C*a^3*b^2
+ 3*B*a^2*b^3 - 15*A*a*b^4)*d^4 - 3*(7*D*a^3*b^2*c + (5*C*a^2*b^3 - 9*B*a*b
^4 + 5*A*b^5)*c)*d^3 + 12*(3*D*a^2*b^3*c^2 - (2*C*a*b^4 - B*b^5)*c^2)*d^2 +
 8*(3*D*a*b^4*c^3 - C*b^5*c^3)*d)*x^2 + 8*(3*D*a^3*b^2*c^3 - C*a^2*b^3*c^3)
*d - (24*(C*a^2*b^3 - B*a*b^4)*c^2*d^2 - (3*D*a^5 + C*a^4*b + 3*B*a^3*b^2 -
 15*A*a^2*b^3)*d^4 + 6*(D*a^4*b*c + (C*a^3*b^2 - 3*B*a^2*b^3 + 5*A*a*b^4)*c
)*d^3 - 16*(3*D*a^2*b^3*c^3 - C*a*b^4*c^3)*d)*x)*sqrt(b^2*c - a*b*d)*log((b
*d*x + 2*b*c - a*d - 2*sqrt(b^2*c - a*b*d)*sqrt(d*x + c))/(b*x + a)) + 2*(8
*D*a^2*b^4*c^4 + 8*A*a^3*b^3*d^4 + (3*D*a^5*b*c + (C*a^4*b^2 - 13*B*a^3*b^3
 + A*a^2*b^4)*c)*d^3 - (13*D*a^4*b^2*c^2 - (13*C*a^3*b^3 + 11*B*a^2*b^4 - 1
1*A*a*b^5)*c^2)*d^2 + (8*D*b^6*c^4 + (5*D*a^4*b^2 - C*a^3*b^3 - 3*B*a^2*b^4
 + 15*A*a*b^5)*d^4 - (17*D*a^3*b^3*c - 3*(3*C*a^2*b^4 - 3*B*a*b^5 - 5*A*b^6
)*c)*d^3 + 12*(D*a^2*b^4*c^2 + B*b^6*c^2)*d^2 - 8*(D*a*b^5*c^3 + C*b^6*c^3)
*d)*x^2 + 2*(D*a^3*b^3*c^3 - (7*C*a^2*b^4 - B*a*b^5 - A*b^6)*c^3)*d + (16*D
*a*b^5*c^4 + (3*D*a^5*b + C*a^4*b^2 - 5*B*a^3*b^3 + 25*A*a^2*b^4)*d^4 - 4*(
2*D*a^4*b^2*c - (C*a^3*b^3 - 4*B*a^2*b^4 - 5*A*a*b^5)*c)*d^3 - (7*D*a^3*b^3
*c^2 - (19*C*a^2*b^4 + 17*B*a*b^5 - 5*A*b^6)*c^2)*d^2 - 4*(D*a^2*b^4*c^3 +
(6*C*a*b^5 - B*b^6)*c^3)*d)*x)*sqrt(d*x + c))/(a^2*b^7*c^5*d - 4*a^3*b^6*c^
4*d^2 + 6*a^4*b^5*c^3*d^3 - 4*a^5*b^4*c^2*d^4 + a^6*b^3*c*d^5 + (b^9*c^4*d^
2 - 4*a*b^8*c^3*d^3 + 6*a^2*b^7*c^2*d^4 - 4*a^3*b^6*c*d^5 + a^4*b^5*d^6)*x^
3 + (b^9*c^5*d - 2*a*b^8*c^4*d^2 - 2*a^2*b^7*c^3*d^3 + 8*a^3*b^6*c^2*d^4 -
7*a^4*b^5*c*d^5 + 2*a^5*b^4*d^6)*x^2 + (2*a*b^8*c^5*d - 7*a^2*b^7*c^4*d^2 +
 8*a^3*b^6*c^3*d^3 - 2*a^4*b^5*c^2*d^4 - 2*a^5*b^4*c*d^5 + a^6*b^3*d^6)*x),
 -1/4*(((3*D*a^5*c + (C*a^4*b + 3*B*a^3*b^2 - 15*A*a^2*b^3)*c)*d^3 + ((3*D*
a^3*b^2 + C*a^2*b^3 + 3*B*a*b^4 - 15*A*b^5)*d^4 - 4*(3*D*a^2*b^3*c + (2*C*a
*b^4 - 3*B*b^5)*c)*d^3 + 8*(3*D*a*b^4*c^2 - C*b^5*c^2)*d^2)*x^3 - 4*(3*D*a^
4*b*c^2 + (2*C*a^3*b^2 - 3*B*a^2*b^3)*c^2)*d^2 + (2*(3*D*a^4*b + C*a^3*b^2
+ 3*B*a^2*b^3 - 15*A*a*b^4)*d^4 - 3*(7*D*a^3*b^2*c + (5*C*a^2*b^3 - 9*B*a*b
^4 + 5*A*b^5)*c)*d^3 + 12*(3*D*a^2*b^3*c^2 - (2*C*a*b^4 - B*b^5)*c^2)*d^2 +
 8*(3*D*a*b^4*c^3 - C*b^5*c^3)*d)*x^2 + 8*(3*D*a^3*b^2*c^3 - C*a^2*b^3*c^3)
*d - (24*(C*a^2*b^3 - B*a*b^4)*c^2*d^2 - (3*D*a^5 + C*a^4*b + 3*B*a^3*b^2 -
```

```
 15*A*a^2*b^3)*d^4 + 6*(D*a^4*b*c + (C*a^3*b^2 - 3*B*a^2*b^3 + 5*A*a*b^4)*c
)*d^3 - 16*(3*D*a^2*b^3*c^3 - C*a*b^4*c^3)*d)*x)*sqrt(-b^2*c + a*b*d)*arcta
n(sqrt(-b^2*c + a*b*d)*sqrt(d*x + c)/(b*d*x + b*c)) + (8*D*a^2*b^4*c^4 + 8*
A*a^3*b^3*d^4 + (3*D*a^5*b*c + (C*a^4*b^2 - 13*B*a^3*b^3 + A*a^2*b^4)*c)*d^
3 - (13*D*a^4*b^2*c^2 - (13*C*a^3*b^3 + 11*B*a^2*b^4 - 11*A*a*b^5)*c^2)*d^2
 + (8*D*b^6*c^4 + (5*D*a^4*b^2 - C*a^3*b^3 - 3*B*a^2*b^4 + 15*A*a*b^5)*d^4
- (17*D*a^3*b^3*c - 3*(3*C*a^2*b^4 - 3*B*a*b^5 - 5*A*b^6)*c)*d^3 + 12*(D*a^
2*b^4*c^2 + B*b^6*c^2)*d^2 - 8*(D*a*b^5*c^3 + C*b^6*c^3)*d)*x^2 + 2*(D*a^3*
b^3*c^3 - (7*C*a^2*b^4 - B*a*b^5 - A*b^6)*c^3)*d + (16*D*a*b^5*c^4 + (3*D*a
^5*b + C*a^4*b^2 - 5*B*a^3*b^3 + 25*A*a^2*b^4)*d^4 - 4*(2*D*a^4*b^2*c - (C*
a^3*b^3 - 4*B*a^2*b^4 - 5*A*a*b^5)*c)*d^3 - (7*D*a^3*b^3*c^2 - (19*C*a^2*b^
4 + 17*B*a*b^5 - 5*A*b^6)*c^2)*d^2 - 4*(D*a^2*b^4*c^3 + (6*C*a*b^5 - B*b^6)
*c^3)*d)*x)*sqrt(d*x + c))/(a^2*b^7*c^5*d - 4*a^3*b^6*c^4*d^2 + 6*a^4*b^5*c
^3*d^3 - 4*a^5*b^4*c^2*d^4 + a^6*b^3*c*d^5 + (b^9*c^4*d^2 - 4*a*b^8*c^3*d^3
 + 6*a^2*b^7*c^2*d^4 - 4*a^3*b^6*c*d^5 + a^4*b^5*d^6)*x^3 + (b^9*c^5*d - 2*
a*b^8*c^4*d^2 - 2*a^2*b^7*c^3*d^3 + 8*a^3*b^6*c^2*d^4 - 7*a^4*b^5*c*d^5 + 2
*a^5*b^4*d^6)*x^2 + (2*a*b^8*c^5*d - 7*a^2*b^7*c^4*d^2 + 8*a^3*b^6*c^3*d^3
- 2*a^4*b^5*c^2*d^4 - 2*a^5*b^4*c*d^5 + a^6*b^3*d^6)*x)]
```

Sympy [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{3/2}}\,dx=\text{Timed out}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(b*x+a)**3/(d*x+c)**(3/2),x)
```

```
[Out] Timed out
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{3/2}}\,dx=\text{Exception raised: ValueError}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^3/(d*x+c)^(3/2),x, algorithm="maxima"
)
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 617, normalized size of antiderivative = 1.76

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{3/2}}\,dx =$$

$$-\frac{(24\,Dab^2c^2-8\,Cb^3c^2-12\,Da^2bcd-8\,Cab^2cd+12\,Bb^3cd+3\,Da^3d^2+Ca^2bd^2+3\,Bab^2d^2-15\,Ab^3d^2)\,\mathrm{arc}}{4\,(b^5c^3-3\,ab^4c^2d+3\,a^2b^3cd^2-a^3b^2d^3)\sqrt{-b^2c+abd}}$$
$$-\frac{2\,(Dc^3-Cc^2d+Bcd^2-Ad^3)}{(b^3c^3d-3\,ab^2c^2d^2+3\,a^2bcd^3-a^3d^4)\sqrt{dx+c}}$$
$$-\frac{12\,(dx+c)^{\frac{3}{2}}Da^2b^2cd-8\,(dx+c)^{\frac{3}{2}}Cab^3cd+4\,(dx+c)^{\frac{3}{2}}Bb^4cd-12\,\sqrt{dx+c}Da^2b^2c^2d+8\,\sqrt{dx+c}Cab^3c^2}{}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^3/(d*x+c)^(3/2),x, algorithm="giac")
```

```
[Out] -1/4*(24*D*a*b^2*c^2 - 8*C*b^3*c^2 - 12*D*a^2*b*c*d - 8*C*a*b^2*c*d + 12*B*
b^3*c*d + 3*D*a^3*d^2 + C*a^2*b*d^2 + 3*B*a*b^2*d^2 - 15*A*b^3*d^2)*arctan(
sqrt(d*x + c)*b/sqrt(-b^2*c + a*b*d))/((b^5*c^3 - 3*a*b^4*c^2*d + 3*a^2*b^3
*c*d^2 - a^3*b^2*d^3)*sqrt(-b^2*c + a*b*d)) - 2*(D*c^3 - C*c^2*d + B*c*d^2
- A*d^3)/((b^3*c^3*d - 3*a*b^2*c^2*d^2 + 3*a^2*b*c*d^3 - a^3*d^4)*sqrt(d*x
+ c)) - 1/4*(12*(d*x + c)^(3/2)*D*a^2*b^2*c*d - 8*(d*x + c)^(3/2)*C*a*b^3*c
*d + 4*(d*x + c)^(3/2)*B*b^4*c*d - 12*sqrt(d*x + c)*D*a^2*b^2*c^2*d + 8*sqr
t(d*x + c)*C*a*b^3*c^2*d - 4*sqrt(d*x + c)*B*b^4*c^2*d - 5*(d*x + c)^(3/2)*
D*a^3*b*d^2 + (d*x + c)^(3/2)*C*a^2*b^2*d^2 + 3*(d*x + c)^(3/2)*B*a*b^3*d^2
 - 7*(d*x + c)^(3/2)*A*b^4*d^2 + 15*sqrt(d*x + c)*D*a^3*b*c*d^2 - 7*sqrt(d*
x + c)*C*a^2*b^2*c*d^2 - sqrt(d*x + c)*B*a*b^3*c*d^2 + 9*sqrt(d*x + c)*A*b^
4*c*d^2 - 3*sqrt(d*x + c)*D*a^4*d^3 - sqrt(d*x + c)*C*a^3*b*d^3 + 5*sqrt(d*
x + c)*B*a^2*b^2*d^3 - 9*sqrt(d*x + c)*A*a*b^3*d^3)/((b^5*c^3 - 3*a*b^4*c^2
*d + 3*a^2*b^3*c*d^2 - a^3*b^2*d^3)*((d*x + c)*b - b*c + a*d)^2)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{3/2}}\,dx = \int \frac{A+B\,x+C\,x^2+x^3\,D}{{\left(a+b\,x\right)}^3\,{\left(c+d\,x\right)}^{3/2}} \,d x$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^3*(c + d*x)^(3/2)),x)
```

```
[Out] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^3*(c + d*x)^(3/2)), x)
```

3.17 $\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4(c+dx)^{3/2}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 463

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4(c+dx)^{3/2}}\,dx=\frac{ab^2Bd^3-a^2bCd^3+a^3d^3D-b^3(6c^2Cd-6Bcd^2+7Ad^3-6c^3D)}{3b^3(bc-ad)^4\sqrt{c+dx}}$$
$$-\frac{Ab^3-a(b^2B-abC+a^2D)}{3b^3(bc-ad)(a+bx)^3\sqrt{c+dx}}$$
$$-\frac{(b^3(6Bc-7Ad)-ab^2(12cC-Bd)-11a^3dD+a^2b(5Cd+18cD))\sqrt{c+dx}}{12b^2(bc-ad)^3(a+bx)^2}$$
$$-\frac{(b^3(24c^2C-42Bcd+49Ad^2)+5a^3d^2D-a^2bd(11Cd-18cD)+ab^2(36cCd-7Bd^2-72c^2D))\sqrt{c+dx}}{24b^2(bc-ad)^4(a+bx)}$$
$$-\frac{(a^3d^3D+a^2bd^2(Cd-6cD)-ab^2d(12cCd-5Bd^2-24c^2D)-b^3(24c^2Cd-30Bcd^2+35Ad^3-16c^3D))}{8b^{5/2}(bc-ad)^{9/2}}$$

```
[Out] -1/8*(a^3*d^3*D+a^2*b*d^2*(C*d-6*D*c)-a*b^2*d*(-5*B*d^2+12*C*c*d-24*D*c^2)-
b^3*(35*A*d^3-30*B*c*d^2+24*C*c^2*d-16*D*c^3))*arctanh(b^(1/2)*(d*x+c)^(1/2
)/(-a*d+b*c)^(1/2))/b^(5/2)/(-a*d+b*c)^(9/2)+1/3*(a*b^2*B*d^3-a^2*b*C*d^3+a
^3*d^3*D-b^3*(7*A*d^3-6*B*c*d^2+6*C*c^2*d-6*D*c^3))/b^3/(-a*d+b*c)^4/(d*x+c
)^(1/2)+1/3*(-A*b^3+a*(B*b^2-C*a*b+D*a^2))/b^3/(-a*d+b*c)/(b*x+a)^3/(d*x+c)
^(1/2)-1/12*(b^3*(-7*A*d+6*B*c)-a*b^2*(-B*d+12*C*c)-11*a^3*d*D+a^2*b*(5*C*d
+18*D*c))*(d*x+c)^(1/2)/b^2/(-a*d+b*c)^3/(b*x+a)^2-1/24*(b^3*(49*A*d^2-42*B
*c*d+24*C*c^2)+5*a^3*d^2*D-a^2*b*d*(11*C*d-18*D*c)+a*b^2*(-7*B*d^2+36*C*c*d
-72*D*c^2))*(d*x+c)^(1/2)/b^2/(-a*d+b*c)^4/(b*x+a)
```

Rubi [A] (verified)

Time = 0.78 (sec) , antiderivative size = 463, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.188$, Rules used = {1635, 911, 1273, 467, 464, 214}

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4(c+dx)^{3/2}}\,dx = -\frac{Ab^3-a(a^2D-abC+b^2B)}{3b^3(a+bx)^3\sqrt{c+dx}(bc-ad)}$$
$$-\frac{\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)(a^3d^3D+a^2bd^2(Cd-6cD)-ab^2d(-5Bd^2-24c^2D+12cCd)-(b^3(35Ad^3-30Bcd^2 -}{8b^{5/2}(bc-ad)^{9/2}}$$
$$-\frac{\sqrt{c+dx}(5a^3d^2D-a^2bd(11Cd-18cD)+ab^2(-7Bd^2-72c^2D+36cCd)+b^3(49Ad^2-42Bcd+24c^2C))}{24b^2(a+bx)(bc-ad)^4}$$
$$+\frac{a^3d^3D-a^2bCd^3+ab^2Bd^3-(b^3(7Ad^3-6Bcd^2-6c^3D+6c^2Cd))}{3b^3\sqrt{c+dx}(bc-ad)^4}$$
$$-\frac{\sqrt{c+dx}(-11a^3dD+a^2b(18cD+5Cd)-ab^2(12cC-Bd)+b^3(6Bc-7Ad))}{12b^2(a+bx)^2(bc-ad)^3}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^4*(c + d*x)^(3/2)),x]
```

```
[Out] (a*b^2*B*d^3 - a^2*b*C*d^3 + a^3*d^3*D - b^3*(6*c^2*C*d - 6*B*c*d^2 + 7*A*d
^3 - 6*c^3*D))/(3*b^3*(b*c - a*d)^4*Sqrt[c + d*x]) - (A*b^3 - a*(b^2*B - a*
b*C + a^2*D))/(3*b^3*(b*c - a*d)*(a + b*x)^3*Sqrt[c + d*x]) - ((b^3*(6*B*c
- 7*A*d) - a*b^2*(12*c*C - B*d) - 11*a^3*d*D + a^2*b*(5*C*d + 18*c*D))*Sqrt
[c + d*x])/(12*b^2*(b*c - a*d)^3*(a + b*x)^2) - ((b^3*(24*c^2*C - 42*B*c*d
+ 49*A*d^2) + 5*a^3*d^2*D - a^2*b*d*(11*C*d - 18*c*D) + a*b^2*(36*c*C*d - 7
*B*d^2 - 72*c^2*D))*Sqrt[c + d*x])/(24*b^2*(b*c - a*d)^4*(a + b*x)) - ((a^3
*d^3*D + a^2*b*d^2*(C*d - 6*c*D) - a*b^2*d*(12*c*C*d - 5*B*d^2 - 24*c^2*D)
- b^3*(24*c^2*C*d - 30*B*c*d^2 + 35*A*d^3 - 16*c^3*D))*ArcTanh[(Sqrt[b]*Sqr
t[c + d*x])/Sqrt[b*c - a*d]])/(8*b^(5/2)*(b*c - a*d)^(9/2))
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 464

```
Int[((e_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_.)*((c_) + (d_.)*(x_)^(n
_)), x_Symbol] :> Simp[c*(e*x)^(m + 1)*((a + b*x^n)^(p + 1)/(a*e*(m + 1))),
 x] + Dist[(a*d*(m + 1) - b*c*(m + n*(p + 1) + 1))/(a*e^n*(m + 1)), Int[(e*
x)^(m + n)*(a + b*x^n)^p, x], x] /; FreeQ[{a, b, c, d, e, p}, x] && NeQ[b*c
 - a*d, 0] && (IntegerQ[n] || GtQ[e, 0]) && ((GtQ[n, 0] && LtQ[m, -1]) || (
LtQ[n, 0] && GtQ[m + n, -1])) &&  !ILtQ[p, -1]
```

Rule 467

```
Int[(x_)^(m_)*((a_) + (b_.)*(x_)^2)^(p_)*((c_) + (d_.)*(x_)^2), x_Symbol] :
> Simp[(-a)^(m/2 - 1)*(b*c - a*d)*x*((a + b*x^2)^(p + 1)/(2*b^(m/2 + 1)*(p
+ 1))), x] + Dist[1/(2*b^(m/2 + 1)*(p + 1)), Int[x^m*(a + b*x^2)^(p + 1)*Ex
pandToSum[2*b*(p + 1)*Together[(b^(m/2)*(c + d*x^2) - (-a)^(m/2 - 1)*(b*c -
 a*d)*x^(-m + 2))/(a + b*x^2)] - ((-a)^(m/2 - 1)*(b*c - a*d))/x^m, x], x],
x] /; FreeQ[{a, b, c, d}, x] && NeQ[b*c - a*d, 0] && LtQ[p, -1] && ILtQ[m/2
, 0] && (IntegerQ[p] || EqQ[m + 2*p + 1, 0])
```

Rule 911

```
Int[((d_.) + (e_.)*(x_))^(m_)*((f_.) + (g_.)*(x_))^(n_)*((a_.) + (b_.)*(x_)
 + (c_.)*(x_)^2)^(p_.), x_Symbol] :> With[{q = Denominator[m]}, Dist[q/e, S
ubst[Int[x^(q*(m + 1) - 1)*((e*f - d*g)/e + g*(x^q/e))^n*((c*d^2 - b*d*e +
a*e^2)/e^2 - (2*c*d - b*e)*(x^q/e^2) + c*(x^(2*q)/e^2))^p, x], x, (d + e*x)
^(1/q)], x]] /; FreeQ[{a, b, c, d, e, f, g}, x] && NeQ[e*f - d*g, 0] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IntegersQ[n, p] && Fra
ctionQ[m]
```

Rule 1273

```
Int[(x_)^(m_)*((d_) + (e_.)*(x_)^2)^(q_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^
4)^(p_.), x_Symbol] :> Simp[(-d)^(m/2 - 1)*(c*d^2 - b*d*e + a*e^2)^p*x*((d
+ e*x^2)^(q + 1)/(2*e^(2*p + m/2)*(q + 1))), x] + Dist[(-d)^(m/2 - 1)/(2*e^
(2*p)*(q + 1)), Int[x^m*(d + e*x^2)^(q + 1)*ExpandToSum[Together[(1/(d + e*
x^2))*(2*(-d)^(-m/2 + 1)*e^(2*p)*(q + 1)*(a + b*x^2 + c*x^4)^p - ((c*d^2 -
b*d*e + a*e^2)^p/(e^(m/2)*x^m))*(d + e*(2*q + 3)*x^2))], x], x], x] /; Free
Q[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && IGtQ[p, 0] && ILtQ[q, -1] &
& ILtQ[m/2, 0]
```

Rule 1635

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol] :
> With[{Qx = PolynomialQuotient[Px, a + b*x, x], R = PolynomialRemainder[Px
, a + b*x, x]}, Simp[R*(a + b*x)^(m + 1)*((c + d*x)^(n + 1)/((m + 1)*(b*c -
 a*d))), x] + Dist[1/((m + 1)*(b*c - a*d)), Int[(a + b*x)^(m + 1)*(c + d*x)
^n*ExpandToSum[(m + 1)*(b*c - a*d)*Qx - d*R*(m + n + 2), x], x], x]] /; Fre
eQ[{a, b, c, d, n}, x] && PolyQ[Px, x] && ILtQ[m, -1] && GtQ[Expon[Px, x],
2]
```

Rubi steps

$$\begin{aligned}\text{integral} &= -\frac{Ab^3-a(b^2B-abC+a^2D)}{3b^3(bc-ad)(a+bx)^3\sqrt{c+dx}} \\ &\quad -\frac{\int \frac{-\frac{b^3(6Bc-7Ad)-ab^2(6cC-Bd)+a^3dD-a^2b(Cd-6cD)}{2b^3}-\frac{3(bc-ad)(bC-aD)x}{b^2}-3\left(c-\frac{ad}{b}\right)Dx^2}{(a+bx)^3(c+dx)^{3/2}}\,dx}{3(bc-ad)}\end{aligned}$$

$$= -\frac{Ab^3 - a(b^2B - abC + a^2D)}{3b^3(bc-ad)(a+bx)^3\sqrt{c+dx}}$$

$$-\frac{2\text{Subst}\Bigg(\int \frac{\frac{-3c^2\left(c-\frac{ad}{b}\right)D+\frac{3cd(bc-ad)(bC-aD)}{b^2}-\frac{d^2\left(b^3(6Bc-7Ad)-ab^2(6cC-Bd)+a^3dD-a^2b(Cd-6cD)\right)}{2b^3}}{d^2}-\frac{\left(-6c\left(c-\frac{ad}{b}\right)D+\frac{3d(bc-ad}{b}}{d^2}}{x^2\left(\frac{-bc+ad}{d}+\frac{bx^2}{d}\right)^3}}{3d(bc-ad)}$$

$$= -\frac{Ab^3 - a(b^2B - abC + a^2D)}{3b^3(bc-ad)(a+bx)^3\sqrt{c+dx}}$$

$$-\frac{\left(b^3(6Bc-7Ad) - ab^2(12cC-Bd) - 11a^3dD + a^2b(5Cd+18cD)\right)\sqrt{c+dx}}{12b^2(bc-ad)^3(a+bx)^2}$$

$$+\frac{d^3\text{Subst}\Bigg(\int \frac{-\frac{2(bc-ad)\left(ab^2Bd^3-a^2bCd^3+a^3d^3D-b^3\left(6c^2Cd-6Bcd^2+7Ad^3-6c^3D\right)\right)}{bd^5}+\frac{3\left(3a^3d^3D-a^2bd^2(5Cd-6cD)+ab^2d\left(12cCd-Bd^2-\right.\right.}{2d^5}}{x^2\left(\frac{-bc+ad}{d}+\frac{bx^2}{d}\right)^2}}{6b^2(bc-ad)^3}$$

$$= -\frac{Ab^3 - a(b^2B - abC + a^2D)}{3b^3(bc-ad)(a+bx)^3\sqrt{c+dx}}$$

$$-\frac{\left(b^3(6Bc-7Ad) - ab^2(12cC-Bd) - 11a^3dD + a^2b(5Cd+18cD)\right)\sqrt{c+dx}}{12b^2(bc-ad)^3(a+bx)^2}$$

$$-\frac{\left(b^3(24c^2C - 42Bcd + 49Ad^2) + 5a^3d^2D - a^2bd(11Cd-18cD) + ab^2(36cCd - 7Bd^2 - 72c^2D)\right)}{24b^2(bc-ad)^4(a+bx)}$$

$$-\frac{d^3\text{Subst}\Bigg(\int \frac{-\frac{4\left(ab^2Bd^3-a^2bCd^3+a^3d^3D-b^3\left(6c^2Cd-6Bcd^2+7Ad^3-6c^3D\right)\right)}{bd^4}+\frac{\left(b^3\left(24c^2C-42Bcd+49Ad^2\right)+5a^3d^2D-a^2bd(11Cd-18cD)\right.}{2d^3(bc-ad)}}{x^2\left(\frac{-bc+ad}{d}+\frac{bx^2}{d}\right)}}{12b^2(bc-ad)^3}$$

$$= \frac{ab^2Bd^3 - a^2bCd^3 + a^3d^3D - b^3(6c^2Cd - 6Bcd^2 + 7Ad^3 - 6c^3D)}{3b^3(bc-ad)^4\sqrt{c+dx}}$$

$$-\frac{Ab^3 - a(b^2B - abC + a^2D)}{3b^3(bc-ad)(a+bx)^3\sqrt{c+dx}}$$

$$-\frac{\left(b^3(6Bc-7Ad) - ab^2(12cC-Bd) - 11a^3dD + a^2b(5Cd+18cD)\right)\sqrt{c+dx}}{12b^2(bc-ad)^3(a+bx)^2}$$

$$-\frac{\left(b^3(24c^2C - 42Bcd + 49Ad^2) + 5a^3d^2D - a^2bd(11Cd-18cD) + ab^2(36cCd - 7Bd^2 - 72c^2D)\right)}{24b^2(bc-ad)^4(a+bx)}$$

$$+\frac{\left(a^3d^3D + a^2bd^2(Cd-6cD) - ab^2d(12cCd - 5Bd^2 - 24c^2D) - b^3(24c^2Cd - 30Bcd^2 + 35Ad^3 -\right.}{8b^2d(bc-ad)^4}$$

$$
\begin{aligned}
&= \frac{ab^2Bd^3 - a^2bCd^3 + a^3d^3D - b^3(6c^2Cd - 6Bcd^2 + 7Ad^3 - 6c^3D)}{3b^3(bc-ad)^4\sqrt{c+dx}} \\
&- \frac{Ab^3 - a(b^2B - abC + a^2D)}{3b^3(bc-ad)(a+bx)^3\sqrt{c+dx}} \\
&- \frac{(b^3(6Bc - 7Ad) - ab^2(12cC - Bd) - 11a^3dD + a^2b(5Cd + 18cD))\sqrt{c+dx}}{12b^2(bc-ad)^3(a+bx)^2} \\
&- \frac{(b^3(24c^2C - 42Bcd + 49Ad^2) + 5a^3d^2D - a^2bd(11Cd - 18cD) + ab^2(36cCd - 7Bd^2 - 72c^2D)}{24b^2(bc-ad)^4(a+bx)} \\
&- \frac{(a^3d^3D + a^2bd^2(Cd - 6cD) - ab^2d(12cCd - 5Bd^2 - 24c^2D) - b^3(24c^2Cd - 30Bcd^2 + 35Ad^3 -}{8b^{5/2}(bc-ad)^{9/2}}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 2.31 (sec) , antiderivative size = 561, normalized size of antiderivative = 1.21

$$
\int \frac{A + Bx + Cx^2 + Dx^3}{(a+bx)^4(c+dx)^{3/2}}\,dx = \frac{-3a^5d^2D(c+dx) + a^4bd(c+dx)(-3Cd + 16cD - 8dDx) + 6b^5cx(4cx(-cC}{}
$$

$$
+ \frac{(a^3d^3D + a^2bd^2(Cd - 6cD) + ab^2d(-12cCd + 5Bd^2 + 24c^2D) + b^3(-24c^2Cd + 30Bcd^2 - 35Ad^3 + 16c^3}{8b^{5/2}(-bc+ad)^{9/2}}
$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^4*(c + d*x)^(3/2)),x]
```

```
[Out] (-3*a^5*d^2*D*(c + d*x) + a^4*b*d*(c + d*x)*(-3*C*d + 16*c*D - 8*d*D*x) + 6
*b^5*c*x*(4*c*x*(-(c*C) - 3*C*d*x + 2*c*D*x) + B*(-2*c^2 + 5*c*d*x + 15*d^2
*x^2)) - A*b^2*(48*a^3*d^3 + 3*a^2*b*d^2*(29*c + 77*d*x) + 2*a*b^2*d*(-19*c
^2 + 49*c*d*x + 140*d^2*x^2) + b^3*(8*c^3 - 14*c^2*d*x + 35*c*d^2*x^2 + 105
*d^3*x^3)) + a*b^4*(B*(-4*c^3 + 82*c^2*d*x + 245*c*d^2*x^2 + 15*d^3*x^3) -
12*c*x*(3*C*d^2*x^2 + 2*c^2*(C - 9*D*x) + c*d*x*(17*C - 6*D*x))) + a^3*b^2*
(92*c^3*D + c^2*(-94*C*d + 58*d*D*x) + d^3*x*(33*B + x*(8*C + 3*D*x)) + c*d
^2*(81*B + x*(-38*C + 17*D*x))) + a^2*b^3*(d^3*x^2*(40*B + 3*C*x) + c^3*(-8
*C + 252*D*x) + 2*c^2*d*(14*B + 5*x*(-25*C + 9*D*x)) + c*d^2*x*(212*B - x*(
95*C + 18*D*x))))/(24*b^2*(b*c - a*d)^4*(a + b*x)^3*Sqrt[c + d*x]) + ((a^3*
d^3*D + a^2*b*d^2*(C*d - 6*c*D) + a*b^2*d*(-12*c*C*d + 5*B*d^2 + 24*c^2*D)
+ b^3*(-24*c^2*C*d + 30*B*c*d^2 - 35*A*d^3 + 16*c^3*D))*ArcTan[(Sqrt[b]*Sqr
t[c + d*x])/Sqrt[-(b*c) + a*d]])/(8*b^(5/2)*(-(b*c) + a*d)^(9/2))
```

Maple [A] (verified)

Time = 1.95 (sec) , antiderivative size = 511, normalized size of antiderivative = 1.10

method	result
pseudoelliptic	$-\frac{35\left(\left(\left(A\,d^{3}-\frac{6}{7}Bc\,d^{2}+\frac{24}{35}C\,c^{2}d-\frac{16}{35}Dc^{3}\right)b^{3}-\frac{a\left(B\,d^{2}-\frac{12}{5}Ccd+\frac{24}{5}Dc^{2}\right)d\,b^{2}}{7}-\frac{a^{2}b\,d^{2}\left(Cd-6Dc\right)}{35}-\frac{a^{3}d^{3}D}{35}\right)\sqrt{dx+c}\,\left(bx+a\right)^{3}\right.}{}$
derivativedivides	$-\frac{2\left(A\,d^{3}-Bc\,d^{2}+C\,c^{2}d-Dc^{3}\right)}{\left(ad-bc\right)^{4}\sqrt{dx+c}}-\frac{2\left(\frac{\left(\frac{19}{16}A\,b^{3}d^{3}-\frac{7}{8}B\,b^{3}c\,d^{2}+\frac{1}{2}C\,b^{3}c^{2}d-\frac{5}{16}Ba\,b^{2}d^{3}-\frac{1}{16}a^{2}bC\,d^{3}+\frac{3}{4}Ca\,b^{2}c\,d^{2}-\frac{1}{16}a^{3}d^{3}D+\frac{3}{8}\right.}{}\right.}{}$
default	$-\frac{2\left(A\,d^{3}-Bc\,d^{2}+C\,c^{2}d-Dc^{3}\right)}{\left(ad-bc\right)^{4}\sqrt{dx+c}}-\frac{2\left(\frac{\left(\frac{19}{16}A\,b^{3}d^{3}-\frac{7}{8}B\,b^{3}c\,d^{2}+\frac{1}{2}C\,b^{3}c^{2}d-\frac{5}{16}Ba\,b^{2}d^{3}-\frac{1}{16}a^{2}bC\,d^{3}+\frac{3}{4}Ca\,b^{2}c\,d^{2}-\frac{1}{16}a^{3}d^{3}D+\frac{3}{8}\right.}{}\right.}{}$

```
[In] int((D*x^3+C*x^2+B*x+A)/(b*x+a)^4/(d*x+c)^(3/2),x,method=_RETURNVERBOSE)
```

```
[Out] -35/8/((a*d-b*c)*b)^(1/2)/(d*x+c)^(1/2)*(((A*d^3-6/7*B*c*d^2+24/35*C*c^2*d-
16/35*D*c^3)*b^3-1/7*a*(B*d^2-12/5*C*c*d+24/5*D*c^2)*d*b^2-1/35*a^2*b*d^2*(
C*d-6*D*c)-1/35*a^3*d^3*D)*(d*x+c)^(1/2)*(b*x+a)^3*arctan(b*(d*x+c)^(1/2)/(
(a*d-b*c)*b)^(1/2))+16/35*((a*d-b*c)*b)^(1/2)*((35/16*A*d^3*x^3+35/48*x^2*c
*(-18/7*B*x+A)*d^2-7/24*x*c^2*(-36/7*C*x^2+15/7*B*x+A)*d+1/6*c^3*(-6*D*x^3+
3*C*x^2+3/2*B*x+A))*b^5-19/24*a*(-140/19*x^2*(-3/56*B*x+A)*d^3-49/19*x*(18/
49*C*x^2-5/2*B*x+A)*c*d^2+c^2*(36/19*D*x^3-102/19*C*x^2+41/19*B*x+A)*d-2/19
*c^3*(-54*D*x^2+6*C*x+B))*b^4+29/16*a^2*(77/29*x*(-1/77*C*x^2-40/231*B*x+A)
*d^3+c*(6/29*D*x^3+95/87*C*x^2-212/87*B*x+A)*d^2-28/87*(45/14*D*x^2-125/14*
C*x+B)*c^2*d+8/87*(-63/2*D*x+C)*c^3)*b^3+((A-1/16*D*x^3-1/6*C*x^2-11/16*B*x
)*d^3-27/16*(17/81*D*x^2-38/81*C*x+B)*c*d^2+47/24*(-29/47*D*x+C)*c^2*d-23/1
2*D*c^3)*a^3*b^2+1/16*a^4*(d*x+c)*((8/3*D*x+C)*d-16/3*D*c)*d*b+1/16*D*a^5*d
^2*(d*x+c)))/(b*x+a)^3/(a*d-b*c)^4/b^2
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1910 vs. 2(435) = 870.

Time = 0.56 (sec) , antiderivative size = 3834, normalized size of antiderivative = 8.28

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4(c+dx)^{3/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^4/(d*x+c)^(3/2),x, algorithm="fricas"
)
```

```
[Out] [1/48*(3*(16*D*a^3*b^3*c^4 + (16*D*b^6*c^3*d + (D*a^3*b^3 + C*a^2*b^4 + 5*B
*a*b^5 - 35*A*b^6)*d^4 - 6*(D*a^2*b^4*c + (2*C*a*b^5 - 5*B*b^6)*c)*d^3 + 24
*(D*a*b^5*c^2 - C*b^6*c^2)*d^2)*x^4 + (D*a^6*c + (C*a^5*b + 5*B*a^4*b^2 - 3
5*A*a^3*b^3)*c)*d^3 + (16*D*b^6*c^4 + 3*(D*a^4*b^2 + C*a^3*b^3 + 5*B*a^2*b^
4 - 35*A*a*b^5)*d^4 - (17*D*a^3*b^3*c + 5*(7*C*a^2*b^4 - 19*B*a*b^5 + 7*A*b
^6)*c)*d^3 + 6*(11*D*a^2*b^4*c^2 - (14*C*a*b^5 - 5*B*b^6)*c^2)*d^2 + 24*(3*
D*a*b^5*c^3 - C*b^6*c^3)*d)*x^3 - 6*(D*a^5*b*c^2 + (2*C*a^4*b^2 - 5*B*a^3*b
^3)*c^2)*d^2 + 3*(16*D*a*b^5*c^4 + (D*a^5*b + C*a^4*b^2 + 5*B*a^3*b^3 - 35*
A*a^2*b^4)*d^4 - (5*D*a^4*b^2*c + (11*C*a^3*b^3 - 35*B*a^2*b^4 + 35*A*a*b^5
)*c)*d^3 + 6*(3*D*a^3*b^3*c^2 - (6*C*a^2*b^4 - 5*B*a*b^5)*c^2)*d^2 + 8*(5*D
*a^2*b^4*c^3 - 3*C*a*b^5*c^3)*d)*x^2 + 24*(D*a^4*b^2*c^3 - C*a^3*b^3*c^3)*d
 + (48*D*a^2*b^4*c^4 + (D*a^6 + C*a^5*b + 5*B*a^4*b^2 - 35*A*a^3*b^3)*d^4 -
 3*(D*a^5*b*c + (3*C*a^4*b^2 - 15*B*a^3*b^3 + 35*A*a^2*b^4)*c)*d^3 + 6*(D*a
^4*b^2*c^2 - 5*(2*C*a^3*b^3 - 3*B*a^2*b^4)*c^2)*d^2 + 8*(11*D*a^3*b^3*c^3 -
 9*C*a^2*b^4*c^3)*d)*x)*sqrt(b^2*c - a*b*d)*log((b*d*x + 2*b*c - a*d - 2*sq
rt(b^2*c - a*b*d)*sqrt(d*x + c))/(b*x + a)) + 2*(92*D*a^3*b^4*c^4 + 48*A*a^
4*b^3*d^4 - 4*(2*C*a^2*b^5 + B*a*b^6 + 2*A*b^7)*c^4 + 3*(D*a^6*b*c + (C*a^5
*b^2 - 27*B*a^4*b^3 + 13*A*a^3*b^4)*c)*d^3 + 3*(16*D*b^7*c^4 - (D*a^4*b^3 +
 C*a^3*b^4 + 5*B*a^2*b^5 - 35*A*a*b^6)*d^4 + (7*D*a^3*b^4*c + (13*C*a^2*b^5
 - 25*B*a*b^6 - 35*A*b^7)*c)*d^3 - 6*(5*D*a^2*b^5*c^2 - (2*C*a*b^6 + 5*B*b^
7)*c^2)*d^2 + 8*(D*a*b^6*c^3 - 3*C*b^7*c^3)*d)*x^3 - (19*D*a^5*b^2*c^2 - (9
1*C*a^4*b^3 + 53*B*a^3*b^4 - 125*A*a^2*b^5)*c^2)*d^2 + (216*D*a*b^6*c^4 - 2
4*C*b^7*c^4 + 8*(D*a^5*b^2 - C*a^4*b^3 - 5*B*a^3*b^4 + 35*A*a^2*b^5)*d^4 -
(25*D*a^4*b^3*c - (103*C*a^3*b^4 - 205*B*a^2*b^5 - 245*A*a*b^6)*c)*d^3 - (7
3*D*a^3*b^4*c^2 - (109*C*a^2*b^5 + 215*B*a*b^6 - 35*A*b^7)*c^2)*d^2 - 6*(21
*D*a^2*b^5*c^3 + 5*(6*C*a*b^6 - B*b^7)*c^3)*d)*x^2 - 2*(38*D*a^4*b^3*c^3 +
(43*C*a^3*b^4 - 16*B*a^2*b^5 - 23*A*a*b^6)*c^3)*d + (252*D*a^2*b^5*c^4 - 12
*(2*C*a*b^6 + B*b^7)*c^4 + 3*(D*a^6*b + C*a^5*b^2 - 11*B*a^4*b^3 + 77*A*a^3
*b^4)*d^4 - (11*D*a^5*b^2*c - (35*C*a^4*b^3 - 179*B*a^3*b^4 - 133*A*a^2*b^5
)*c)*d^3 - 2*(25*D*a^4*b^3*c^2 - (106*C*a^3*b^4 + 65*B*a^2*b^5 - 56*A*a*b^6
)*c^2)*d^2 - 2*(97*D*a^3*b^4*c^3 + (113*C*a^2*b^5 - 47*B*a*b^6 - 7*A*b^7)*c
^3)*d)*x)*sqrt(d*x + c))/(a^3*b^8*c^6 - 5*a^4*b^7*c^5*d + 10*a^5*b^6*c^4*d^
2 - 10*a^6*b^5*c^3*d^3 + 5*a^7*b^4*c^2*d^4 - a^8*b^3*c*d^5 + (b^11*c^5*d -
5*a*b^10*c^4*d^2 + 10*a^2*b^9*c^3*d^3 - 10*a^3*b^8*c^2*d^4 + 5*a^4*b^7*c*d^
```

```
5 - a^5*b^6*d^6)*x^4 + (b^11*c^6 - 2*a*b^10*c^5*d - 5*a^2*b^9*c^4*d^2 + 20*
a^3*b^8*c^3*d^3 - 25*a^4*b^7*c^2*d^4 + 14*a^5*b^6*c*d^5 - 3*a^6*b^5*d^6)*x^
3 + 3*(a*b^10*c^6 - 4*a^2*b^9*c^5*d + 5*a^3*b^8*c^4*d^2 - 5*a^5*b^6*c^2*d^4
 + 4*a^6*b^5*c*d^5 - a^7*b^4*d^6)*x^2 + (3*a^2*b^9*c^6 - 14*a^3*b^8*c^5*d +
 25*a^4*b^7*c^4*d^2 - 20*a^5*b^6*c^3*d^3 + 5*a^6*b^5*c^2*d^4 + 2*a^7*b^4*c*
d^5 - a^8*b^3*d^6)*x), 1/24*(3*(16*D*a^3*b^3*c^4 + (16*D*b^6*c^3*d + (D*a^3
*b^3 + C*a^2*b^4 + 5*B*a*b^5 - 35*A*b^6)*d^4 - 6*(D*a^2*b^4*c + (2*C*a*b^5
- 5*B*b^6)*c)*d^3 + 24*(D*a*b^5*c^2 - C*b^6*c^2)*d^2)*x^4 + (D*a^6*c + (C*a
^5*b + 5*B*a^4*b^2 - 35*A*a^3*b^3)*c)*d^3 + (16*D*b^6*c^4 + 3*(D*a^4*b^2 +
C*a^3*b^3 + 5*B*a^2*b^4 - 35*A*a*b^5)*d^4 - (17*D*a^3*b^3*c + 5*(7*C*a^2*b^
4 - 19*B*a*b^5 + 7*A*b^6)*c)*d^3 + 6*(11*D*a^2*b^4*c^2 - (14*C*a*b^5 - 5*B*
b^6)*c^2)*d^2 + 24*(3*D*a*b^5*c^3 - C*b^6*c^3)*d)*x^3 - 6*(D*a^5*b*c^2 + (2
*C*a^4*b^2 - 5*B*a^3*b^3)*c^2)*d^2 + 3*(16*D*a*b^5*c^4 + (D*a^5*b + C*a^4*b
^2 + 5*B*a^3*b^3 - 35*A*a^2*b^4)*d^4 - (5*D*a^4*b^2*c + (11*C*a^3*b^3 - 35*
B*a^2*b^4 + 35*A*a*b^5)*c)*d^3 + 6*(3*D*a^3*b^3*c^2 - (6*C*a^2*b^4 - 5*B*a*
b^5)*c^2)*d^2 + 8*(5*D*a^2*b^4*c^3 - 3*C*a*b^5*c^3)*d)*x^2 + 24*(D*a^4*b^2*
c^3 - C*a^3*b^3*c^3)*d + (48*D*a^2*b^4*c^4 + (D*a^6 + C*a^5*b + 5*B*a^4*b^2
 - 35*A*a^3*b^3)*d^4 - 3*(D*a^5*b*c + (3*C*a^4*b^2 - 15*B*a^3*b^3 + 35*A*a^
2*b^4)*c)*d^3 + 6*(D*a^4*b^2*c^2 - 5*(2*C*a^3*b^3 - 3*B*a^2*b^4)*c^2)*d^2 +
 8*(11*D*a^3*b^3*c^3 - 9*C*a^2*b^4*c^3)*d)*x)*sqrt(-b^2*c + a*b*d)*arctan(s
qrt(-b^2*c + a*b*d)*sqrt(d*x + c)/(b*d*x + b*c)) + (92*D*a^3*b^4*c^4 + 48*A
*a^4*b^3*d^4 - 4*(2*C*a^2*b^5 + B*a*b^6 + 2*A*b^7)*c^4 + 3*(D*a^6*b*c + (C*
a^5*b^2 - 27*B*a^4*b^3 + 13*A*a^3*b^4)*c)*d^3 + 3*(16*D*b^7*c^4 - (D*a^4*b^
3 + C*a^3*b^4 + 5*B*a^2*b^5 - 35*A*a*b^6)*d^4 + (7*D*a^3*b^4*c + (13*C*a^2*
b^5 - 25*B*a*b^6 - 35*A*b^7)*c)*d^3 - 6*(5*D*a^2*b^5*c^2 - (2*C*a*b^6 + 5*B
*b^7)*c^2)*d^2 + 8*(D*a*b^6*c^3 - 3*C*b^7*c^3)*d)*x^3 - (19*D*a^5*b^2*c^2 -
 (91*C*a^4*b^3 + 53*B*a^3*b^4 - 125*A*a^2*b^5)*c^2)*d^2 + (216*D*a*b^6*c^4
- 24*C*b^7*c^4 + 8*(D*a^5*b^2 - C*a^4*b^3 - 5*B*a^3*b^4 + 35*A*a^2*b^5)*d^4
 - (25*D*a^4*b^3*c - (103*C*a^3*b^4 - 205*B*a^2*b^5 - 245*A*a*b^6)*c)*d^3 -
 (73*D*a^3*b^4*c^2 - (109*C*a^2*b^5 + 215*B*a*b^6 - 35*A*b^7)*c^2)*d^2 - 6*
(21*D*a^2*b^5*c^3 + 5*(6*C*a*b^6 - B*b^7)*c^3)*d)*x^2 - 2*(38*D*a^4*b^3*c^3
 + (43*C*a^3*b^4 - 16*B*a^2*b^5 - 23*A*a*b^6)*c^3)*d + (252*D*a^2*b^5*c^4 -
 12*(2*C*a*b^6 + B*b^7)*c^4 + 3*(D*a^6*b + C*a^5*b^2 - 11*B*a^4*b^3 + 77*A*
a^3*b^4)*d^4 - (11*D*a^5*b^2*c - (35*C*a^4*b^3 - 179*B*a^3*b^4 - 133*A*a^2*
b^5)*c)*d^3 - 2*(25*D*a^4*b^3*c^2 - (106*C*a^3*b^4 + 65*B*a^2*b^5 - 56*A*a*
b^6)*c^2)*d^2 - 2*(97*D*a^3*b^4*c^3 + (113*C*a^2*b^5 - 47*B*a*b^6 - 7*A*b^7
)*c^3)*d)*x)*sqrt(d*x + c))/(a^3*b^8*c^6 - 5*a^4*b^7*c^5*d + 10*a^5*b^6*c^4
*d^2 - 10*a^6*b^5*c^3*d^3 + 5*a^7*b^4*c^2*d^4 - a^8*b^3*c*d^5 + (b^11*c^5*d
 - 5*a*b^10*c^4*d^2 + 10*a^2*b^9*c^3*d^3 - 10*a^3*b^8*c^2*d^4 + 5*a^4*b^7*c
*d^5 - a^5*b^6*d^6)*x^4 + (b^11*c^6 - 2*a*b^10*c^5*d - 5*a^2*b^9*c^4*d^2 +
20*a^3*b^8*c^3*d^3 - 25*a^4*b^7*c^2*d^4 + 14*a^5*b^6*c*d^5 - 3*a^6*b^5*d^6)
*x^3 + 3*(a*b^10*c^6 - 4*a^2*b^9*c^5*d + 5*a^3*b^8*c^4*d^2 - 5*a^5*b^6*c^2*
d^4 + 4*a^6*b^5*c*d^5 - a^7*b^4*d^6)*x^2 + (3*a^2*b^9*c^6 - 14*a^3*b^8*c^5*
d + 25*a^4*b^7*c^4*d^2 - 20*a^5*b^6*c^3*d^3 + 5*a^6*b^5*c^2*d^4 + 2*a^7*b^4
*c*d^5 - a^8*b^3*d^6)*x)]
```

Sympy [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4(c+dx)^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(b*x+a)**4/(d*x+c)**(3/2),x)
```

```
[Out] Timed out
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4(c+dx)^{3/2}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^4/(d*x+c)^(3/2),x, algorithm="maxima"
)
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the ’assume’ command before evaluation *may* h
elp (example of legal syntax is ’assume(a*d-b*c>0)’, see ‘assume?‘ for more
 detail
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1085 vs. $2(435) = 870$.

Time = 0.32 (sec) , antiderivative size = 1085, normalized size of antiderivative = 2.34

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4(c+dx)^{3/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^4/(d*x+c)^(3/2),x, algorithm="giac")
```

```
[Out] 1/8*(16*D*b^3*c^3 + 24*D*a*b^2*c^2*d - 24*C*b^3*c^2*d - 6*D*a^2*b*c*d^2 - 1
2*C*a*b^2*c*d^2 + 30*B*b^3*c*d^2 + D*a^3*d^3 + C*a^2*b*d^3 + 5*B*a*b^2*d^3
- 35*A*b^3*d^3)*arctan(sqrt(d*x + c)*b/sqrt(-b^2*c + a*b*d))/((b^6*c^4 - 4*
a*b^5*c^3*d + 6*a^2*b^4*c^2*d^2 - 4*a^3*b^3*c*d^3 + a^4*b^2*d^4)*sqrt(-b^2*
c + a*b*d)) + 2*(D*c^3 - C*c^2*d + B*c*d^2 - A*d^3)/((b^4*c^4 - 4*a*b^3*c^3
*d + 6*a^2*b^2*c^2*d^2 - 4*a^3*b*c*d^3 + a^4*d^4)*sqrt(d*x + c)) + 1/24*(72
*(d*x + c)^(5/2)*D*a*b^4*c^2*d - 24*(d*x + c)^(5/2)*C*b^5*c^2*d - 144*(d*x
+ c)^(3/2)*D*a*b^4*c^3*d + 48*(d*x + c)^(3/2)*C*b^5*c^3*d + 72*sqrt(d*x + c
)*D*a*b^4*c^4*d - 24*sqrt(d*x + c)*C*b^5*c^4*d - 18*(d*x + c)^(5/2)*D*a^2*b
^3*c*d^2 - 36*(d*x + c)^(5/2)*C*a*b^4*c*d^2 + 42*(d*x + c)^(5/2)*B*b^5*c*d^
```

```
2 + 144*(d*x + c)^(3/2)*D*a^2*b^3*c^2*d^2 + 48*(d*x + c)^(3/2)*C*a*b^4*c^2*
d^2 - 96*(d*x + c)^(3/2)*B*b^5*c^2*d^2 - 126*sqrt(d*x + c)*D*a^2*b^3*c^3*d^
2 - 12*sqrt(d*x + c)*C*a*b^4*c^3*d^2 + 54*sqrt(d*x + c)*B*b^5*c^3*d^2 + 3*(
d*x + c)^(5/2)*D*a^3*b^2*d^3 + 3*(d*x + c)^(5/2)*C*a^2*b^3*d^3 + 15*(d*x +
c)^(5/2)*B*a*b^4*d^3 - 57*(d*x + c)^(5/2)*A*b^5*d^3 + 8*(d*x + c)^(3/2)*D*a
^3*b^2*c*d^3 - 104*(d*x + c)^(3/2)*C*a^2*b^3*c*d^3 + 56*(d*x + c)^(3/2)*B*a
*b^4*c*d^3 + 136*(d*x + c)^(3/2)*A*b^5*c*d^3 + 33*sqrt(d*x + c)*D*a^3*b^2*c
^2*d^3 + 93*sqrt(d*x + c)*C*a^2*b^3*c^2*d^3 - 75*sqrt(d*x + c)*B*a*b^4*c^2*
d^3 - 87*sqrt(d*x + c)*A*b^5*c^2*d^3 - 8*(d*x + c)^(3/2)*D*a^4*b*d^4 + 8*(d
*x + c)^(3/2)*C*a^3*b^2*d^4 + 40*(d*x + c)^(3/2)*B*a^2*b^3*d^4 - 136*(d*x +
 c)^(3/2)*A*a*b^4*d^4 + 24*sqrt(d*x + c)*D*a^4*b*c*d^4 - 54*sqrt(d*x + c)*C
*a^3*b^2*c*d^4 - 12*sqrt(d*x + c)*B*a^2*b^3*c*d^4 + 174*sqrt(d*x + c)*A*a*b
^4*c*d^4 - 3*sqrt(d*x + c)*D*a^5*d^5 - 3*sqrt(d*x + c)*C*a^4*b*d^5 + 33*sqr
t(d*x + c)*B*a^3*b^2*d^5 - 87*sqrt(d*x + c)*A*a^2*b^3*d^5)/((b^6*c^4 - 4*a*
b^5*c^3*d + 6*a^2*b^4*c^2*d^2 - 4*a^3*b^3*c*d^3 + a^4*b^2*d^4)*((d*x + c)*b
 - b*c + a*d)^3)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^4(c+dx)^{3/2}}\,dx=\int \frac{A+B\,x+C\,x^2+x^3\,D}{{\left(a+b\,x\right)}^4\,{\left(c+d\,x\right)}^{3/2}}\,dx$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^4*(c + d*x)^(3/2)),x)
```

```
[Out] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^4*(c + d*x)^(3/2)), x)
```

3.18 $\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 434

$$\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx=\frac{2(bc-ad)^3\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{3d^7(c+dx)^{3/2}}$$
$$+\frac{2(bc-ad)^2\left(ad\left(2cCd-Bd^2-3c^2D\right)-b\left(5c^2Cd-4Bcd^2+3Ad^3-6c^3D\right)\right)}{d^7\sqrt{c+dx}}$$
$$-\frac{2(bc-ad)\left(a^2d^2(Cd-3cD)-abd\left(8cCd-3Bd^2-15c^2D\right)+b^2\left(10c^2Cd-6Bcd^2+3Ad^3-15c^3D\right)\right)\sqrt{c}}{d^7}$$
$$+\frac{2\left(a^3d^3D+3a^2bd^2(Cd-4cD)-3ab^2d\left(4cCd-Bd^2-10c^2D\right)+b^3\left(10c^2Cd-4Bcd^2+Ad^3-20c^3D\right)\right)(c}{3d^7}$$
$$+\frac{2b\left(3a^2d^2D+3abd(Cd-5cD)-b^2\left(5cCd-Bd^2-15c^2D\right)\right)(c+dx)^{5/2}}{5d^7}$$
$$+\frac{2b^2(bCd-6bcD+3adD)(c+dx)^{7/2}}{7d^7}+\frac{2b^3D(c+dx)^{9/2}}{9d^7}$$

```
[Out] 2/3*(-a*d+b*c)^3*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)/d^7/(d*x+c)^(3/2)+2/3*(a^3*d
^3*D+3*a^2*b*d^2*(C*d-4*D*c)-3*a*b^2*d*(-B*d^2+4*C*c*d-10*D*c^2)+b^3*(A*d^3
-4*B*c*d^2+10*C*c^2*d-20*D*c^3))*(d*x+c)^(3/2)/d^7+2/5*b*(3*a^2*d^2*D+3*a*b
*d*(C*d-5*D*c)-b^2*(-B*d^2+5*C*c*d-15*D*c^2))*(d*x+c)^(5/2)/d^7+2/7*b^2*(C*
b*d+3*D*a*d-6*D*b*c)*(d*x+c)^(7/2)/d^7+2/9*b^3*D*(d*x+c)^(9/2)/d^7+2*(-a*d+
b*c)^2*(a*d*(-B*d^2+2*C*c*d-3*D*c^2)-b*(3*A*d^3-4*B*c*d^2+5*C*c^2*d-6*D*c^3
))/d^7/(d*x+c)^(1/2)-2*(-a*d+b*c)*(a^2*d^2*(C*d-3*D*c)-a*b*d*(-3*B*d^2+8*C*
c*d-15*D*c^2)+b^2*(3*A*d^3-6*B*c*d^2+10*C*c^2*d-15*D*c^3))*(d*x+c)^(1/2)/d^
7
```

Rubi [A] (verified)

Time = 0.24 (sec) , antiderivative size = 434, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}} = 0.031$, Rules used = {1634}

$$\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx =$$

$$-\frac{2\sqrt{c+dx}(bc-ad)\left(a^2d^2(Cd-3cD)-abd(-3Bd^2-15c^2D+8cCd)+b^2(3Ad^3-6Bcd^2-15c^3D+10c^2}{d^7}$$

$$+\frac{2b(c+dx)^{5/2}\left(3a^2d^2D+3abd(Cd-5cD)-\left(b^2(-Bd^2-15c^2D+5cCd)\right)\right)}{5d^7}$$

$$+\frac{2(c+dx)^{3/2}\left(a^3d^3D+3a^2bd^2(Cd-4cD)-3ab^2d(-Bd^2-10c^2D+4cCd)+b^3(Ad^3-4Bcd^2-20c^3D+1\right.}{3d^7}$$

$$+\frac{2(bc-ad)^2\left(ad(-Bd^2-3c^2D+2cCd)-b(3Ad^3-4Bcd^2-6c^3D+5c^2Cd)\right)}{d^7\sqrt{c+dx}}$$

$$+\frac{2(bc-ad)^3\left(Ad^3-Bcd^2+c^3(-D)+c^2Cd\right)}{3d^7(c+dx)^{3/2}}$$

$$+\frac{2b^2(c+dx)^{7/2}(3adD-6bcD+bCd)}{7d^7}+\frac{2b^3D(c+dx)^{9/2}}{9d^7}$$

```
[In] Int[((a + b*x)^3*(A + B*x + C*x^2 + D*x^3))/(c + d*x)^(5/2),x]
```

```
[Out] (2*(b*c - a*d)^3*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D))/(3*d^7*(c + d*x)^(3/2
)) + (2*(b*c - a*d)^2*(a*d*(2*c*C*d - B*d^2 - 3*c^2*D) - b*(5*c^2*C*d - 4*B
*c*d^2 + 3*A*d^3 - 6*c^3*D)))/(d^7*Sqrt[c + d*x]) - (2*(b*c - a*d)*(a^2*d^2
*(C*d - 3*c*D) - a*b*d*(8*c*C*d - 3*B*d^2 - 15*c^2*D) + b^2*(10*c^2*C*d - 6
*B*c*d^2 + 3*A*d^3 - 15*c^3*D))*Sqrt[c + d*x])/d^7 + (2*(a^3*d^3*D + 3*a^2*
b*d^2*(C*d - 4*c*D) - 3*a*b^2*d*(4*c*C*d - B*d^2 - 10*c^2*D) + b^3*(10*c^2*
C*d - 4*B*c*d^2 + A*d^3 - 20*c^3*D))*(c + d*x)^(3/2))/(3*d^7) + (2*b*(3*a^2
*d^2*D + 3*a*b*d*(C*d - 5*c*D) - b^2*(5*c*C*d - B*d^2 - 15*c^2*D))*(c + d*x
)^(5/2))/(5*d^7) + (2*b^2*(b*C*d - 6*b*c*D + 3*a*d*D)*(c + d*x)^(7/2))/(7*d
^7) + (2*b^3*D*(c + d*x)^(9/2))/(9*d^7)
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \left(\frac{(-bc+ad)^3\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{d^6(c+dx)^{5/2}}\right.\\
&\quad+\frac{(bc-ad)^2\left(-ad(2cCd-Bd^2-3c^2D)+b(5c^2Cd-4Bcd^2+3Ad^3-6c^3D)\right)}{d^6(c+dx)^{3/2}}\\
&\quad+\frac{(bc-ad)\left(-a^2d^2(Cd-3cD)+abd(8cCd-3Bd^2-15c^2D)-b^2(10c^2Cd-6Bcd^2+3Ad^3-15c^3D)\right)}{d^6\sqrt{c+dx}}\\
&\quad+\frac{(a^3d^3D+3a^2bd^2(Cd-4cD)-3ab^2d(4cCd-Bd^2-10c^2D)+b^3(10c^2Cd-4Bcd^2+Ad^3-20c^3D))}{d^6}\\
&\quad+\frac{b(3a^2d^2D+3abd(Cd-5cD)-b^2(5cCd-Bd^2-15c^2D))(c+dx)^{3/2}}{d^6}\\
&\quad\left.+\frac{b^2(bCd-6bcD+3adD)(c+dx)^{5/2}}{d^6}+\frac{b^3D(c+dx)^{7/2}}{d^6}\right)dx\\
&= \frac{2(bc-ad)^3\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{3d^7(c+dx)^{3/2}}\\
&\quad+\frac{2(bc-ad)^2\left(ad(2cCd-Bd^2-3c^2D)-b(5c^2Cd-4Bcd^2+3Ad^3-6c^3D)\right)}{d^7\sqrt{c+dx}}\\
&\quad-\frac{2(bc-ad)\left(a^2d^2(Cd-3cD)-abd(8cCd-3Bd^2-15c^2D)+b^2(10c^2Cd-6Bcd^2+3Ad^3-15\right.}{d^7}\\
&\quad+\frac{2(a^3d^3D+3a^2bd^2(Cd-4cD)-3ab^2d(4cCd-Bd^2-10c^2D)+b^3(10c^2Cd-4Bcd^2+Ad^3-2}{3d^7}\\
&\quad+\frac{2b(3a^2d^2D+3abd(Cd-5cD)-b^2(5cCd-Bd^2-15c^2D))(c+dx)^{5/2}}{5d^7}\\
&\quad+\frac{2b^2(bCd-6bcD+3adD)(c+dx)^{7/2}}{7d^7}+\frac{2b^3D(c+dx)^{9/2}}{9d^7}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.53 (sec) , antiderivative size = 485, normalized size of antiderivative = 1.12

$$
\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \frac{2(-105a^3d^3(16c^3D-8c^2d(C-3Dx)+2cd^2(B+3x(-2C+D}{}
$$

```
[In] Integrate[((a + b*x)^3*(A + B*x + C*x^2 + D*x^3))/(c + d*x)^(5/2),x]
```

```
[Out] (2*(-105*a^3*d^3*(16*c^3*D - 8*c^2*d*(C - 3*D*x) + 2*c*d^2*(B + 3*x*(-2*C +
 D*x)) + d^3*(A + 3*B*x - x^2*(3*C + D*x))) + 63*a^2*b*d^2*(128*c^4*D + c^3
*(-80*C*d + 192*d*D*x) + 8*c^2*d^2*(5*B + 3*x*(-5*C + 2*D*x)) + d^4*x*(-15*
A + x*(15*B + 5*C*x + 3*D*x^2)) - 2*c*d^3*(5*A + x*(-30*B + 15*C*x + 4*D*x^
2))) + b^3*(5120*c^6*D - 3840*c^5*d*(C - 2*D*x) + 384*c^4*d^2*(7*B + 5*x*(-
3*C + D*x)) + 24*c^2*d^4*x*(-105*A + x*(42*B + 5*x*(2*C + D*x))) - 6*c*d^5*
x^2*(105*A + x*(28*B + 5*x*(3*C + 2*D*x))) - 16*c^3*d^3*(105*A + 2*x*(-126*
```

```
B + 5*x*(9*C + 2*D*x))) + d^6*x^3*(105*A + x*(63*B + 5*x*(9*C + 7*D*x)))) +
 9*a*b^2*d*(-1280*c^5*D + 128*c^4*d*(7*C - 15*D*x) - 16*c^3*d^2*(35*B + 6*x
*(-14*C + 5*D*x)) + d^5*x^2*(105*A + x*(35*B + 3*x*(7*C + 5*D*x))) + 8*c^2*
d^3*(35*A + x*(-105*B + 2*x*(21*C + 5*D*x))) - 2*c*d^4*x*(-210*A + x*(105*B
 + x*(28*C + 15*D*x))))))/(315*d^7*(c + d*x)^(3/2))
```

Maple [A] (verified)

Time = 1.77 (sec) , antiderivative size = 439, normalized size of antiderivative = 1.01

method	result
pseudoelliptic	$-\frac{2\left(\left(-x^{3}\left(\frac{1}{3}Dx^{3}+\frac{3}{7}C\,x^{2}+\frac{3}{5}Bx+A\right)b^{3}-9a\left(A+\frac{1}{7}Dx^{3}+\frac{1}{5}C\,x^{2}+\frac{1}{3}Bx\right)x^{2}b^{2}+9a^{2}x\left(-\frac{1}{5}Dx^{3}-\frac{1}{3}C\,x^{2}-Bx+A\right)b+a^{3}\left(-Dx\right.\right.\right.}{}$
gosper	$-\frac{2\left(-35Db^{3}x^{6}d^{6}-45C\,b^{3}d^{6}x^{5}-135Da\,b^{2}d^{6}x^{5}+60Db^{3}c\,d^{5}x^{5}-63B\,b^{3}d^{6}x^{4}-189Ca\,b^{2}d^{6}x^{4}+90C\,b^{3}c\,d^{5}x^{4}-189Da^{2}b\,d^{6}x\right.}{}$
trager	$-\frac{2\left(-35Db^{3}x^{6}d^{6}-45C\,b^{3}d^{6}x^{5}-135Da\,b^{2}d^{6}x^{5}+60Db^{3}c\,d^{5}x^{5}-63B\,b^{3}d^{6}x^{4}-189Ca\,b^{2}d^{6}x^{4}+90C\,b^{3}c\,d^{5}x^{4}-189Da^{2}b\,d^{6}x\right.}{}$
derivativedivides	$\frac{\frac{2Db^{3}(dx+c)^{\frac{9}{2}}}{9}-\frac{2\left(a^{3}A\,d^{6}-3A\,a^{2}bc\,d^{5}+3Aa\,b^{2}c^{2}d^{4}-A\,b^{3}c^{3}d^{3}-B\,a^{3}c\,d^{5}+3B\,a^{2}b\,c^{2}d^{4}-3Ba\,b^{2}c^{3}d^{3}+B\,b^{3}c^{4}d^{2}+C\,a^{3}c^{2}d^{4}-3C\,a^{2}b\,c^{3}\right.}{3(dx+c)^{\frac{3}{2}}}}{}$
default	$\frac{\frac{2Db^{3}(dx+c)^{\frac{9}{2}}}{9}-\frac{2\left(a^{3}A\,d^{6}-3A\,a^{2}bc\,d^{5}+3Aa\,b^{2}c^{2}d^{4}-A\,b^{3}c^{3}d^{3}-B\,a^{3}c\,d^{5}+3B\,a^{2}b\,c^{2}d^{4}-3Ba\,b^{2}c^{3}d^{3}+B\,b^{3}c^{4}d^{2}+C\,a^{3}c^{2}d^{4}-3C\,a^{2}b\,c^{3}\right.}{3(dx+c)^{\frac{3}{2}}}}{}$

```
[In] int((b*x+a)^3*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x,method=_RETURNVERBOSE)
```

```
[Out] -2/3/(d*x+c)^(3/2)*((-x^3*(1/3*D*x^3+3/7*C*x^2+3/5*B*x+A)*b^3-9*a*(A+1/7*D*
x^3+1/5*C*x^2+1/3*B*x)*x^2*b^2+9*a^2*x*(-1/5*D*x^3-1/3*C*x^2-B*x+A)*b+a^3*(
-D*x^3-3*C*x^2+3*B*x+A))*d^6+6*(x^2*(2/21*D*x^3+1/7*C*x^2+4/15*B*x+A)*b^3-6
*a*x*(-1/14*D*x^3-2/15*C*x^2-1/2*B*x+A)*b^2+a^2*(4/5*D*x^3+3*C*x^2-6*B*x+A)
*b+1/3*a^3*(3*D*x^2-6*C*x+B))*c*d^5-24*c^2*(-(-1/21*D*x^3-2/21*C*x^2-2/5*B*
x+A)*x*b^3+a*(2/7*D*x^3+6/5*C*x^2-3*B*x+A)*b^2+a^2*(6/5*D*x^2-3*C*x+B)*b+1/
3*a^3*(-3*D*x+C))*d^4+16*((4/21*D*x^3+6/7*C*x^2-12/5*B*x+A)*b^3+3*a*(6/7*D*
x^2-12/5*C*x+B)*b^2+3*(-12/5*D*x+C)*a^2*b+D*a^3)*c^3*d^3-128/5*b*c^4*((5/7*
D*x^2-15/7*C*x+B)*b^2+3*a*(-15/7*D*x+C)*b+3*D*a^2)*d^2+256/7*((-2*D*x+C)*b+
3*D*a)*b^2*c^5*d-1024/21*D*b^3*c^6)/d^7
```

Fricas [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 689, normalized size of antiderivative = 1.59

$$\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \frac{2\left(35\,Db^3d^6x^6+5120\,Db^3c^6-105\,Aa^3d^6+840\left(Ca^3+3\,Ba^2b+\right.\right.}{}$$

```
[In] integrate((b*x+a)^3*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x, algorithm="fricas"
)
```

```
[Out] 2/315*(35*D*b^3*d^6*x^6 + 5120*D*b^3*c^6 - 105*A*a^3*d^6 + 840*(C*a^3 + 3*B
*a^2*b + 3*A*a*b^2)*c^2*d^4 - 210*(B*a^3 + 3*A*a^2*b)*c*d^5 - 15*(4*D*b^3*c
*d^5 - 3*(3*D*a*b^2 + C*b^3)*d^6)*x^5 + 3*(40*D*b^3*c^2*d^4 + 21*(3*D*a^2*b
 + 3*C*a*b^2 + B*b^3)*d^6 - 30*(3*D*a*b^2*c + C*b^3*c)*d^5)*x^4 - 1680*(D*a
^3*c^3 + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^3)*d^3 - (320*D*b^3*c^3*d^3 - 10
5*(D*a^3 + 3*C*a^2*b + 3*B*a*b^2 + A*b^3)*d^6 + 168*(3*D*a^2*b*c + (3*C*a*b
^2 + B*b^3)*c)*d^5 - 240*(3*D*a*b^2*c^2 + C*b^3*c^2)*d^4)*x^3 + 2688*(3*D*a
^2*b*c^4 + (3*C*a*b^2 + B*b^3)*c^4)*d^2 + 3*(640*D*b^3*c^4*d^2 + 105*(C*a^3
 + 3*B*a^2*b + 3*A*a*b^2)*d^6 - 210*(D*a^3*c + (3*C*a^2*b + 3*B*a*b^2 + A*b
^3)*c)*d^5 + 336*(3*D*a^2*b*c^2 + (3*C*a*b^2 + B*b^3)*c^2)*d^4 - 480*(3*D*a
*b^2*c^3 + C*b^3*c^3)*d^3)*x^2 - 3840*(3*D*a*b^2*c^5 + C*b^3*c^5)*d + 3*(25
60*D*b^3*c^5*d + 420*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c*d^5 - 105*(B*a^3 + 3
*A*a^2*b)*d^6 - 840*(D*a^3*c^2 + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^2)*d^4 +
 1344*(3*D*a^2*b*c^3 + (3*C*a*b^2 + B*b^3)*c^3)*d^3 - 1920*(3*D*a*b^2*c^4 +
 C*b^3*c^4)*d^2)*x)*sqrt(d*x + c)/(d^9*x^2 + 2*c*d^8*x + c^2*d^7)
```

Sympy [A] (verification not implemented)

Time = 73.78 (sec) , antiderivative size = 729, normalized size of antiderivative = 1.68

$$\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \begin{cases} \frac{2\left(\frac{Db^3(c+dx)^{\frac{9}{2}}}{9d^6}+\frac{(c+dx)^{\frac{7}{2}}\left(Cb^3d+3Dab^2d-6Db^3c\right)}{7d^6}+\frac{(c+dx)^{\frac{5}{2}}\left(Bb^3d^2+3Cab^2d^2-5Cb^3cd+\right.}{5d^6}\right.}{} \\ \frac{Aa^3x+\frac{Db^3x^7}{7}+\frac{x^6\left(Cb^3+3Dab^2\right)}{6}+\frac{x^5\left(Bb^3+3Cab^2+3Da^2b\right)}{5}+\frac{x^4\left(Ab^3+3Bab^2+3Ca^2b+Da\right.}{4}}{c^{\frac{5}{2}}} \end{cases}$$

```
[In] integrate((b*x+a)**3*(D*x**3+C*x**2+B*x+A)/(d*x+c)**(5/2),x)
```

```
[Out] Piecewise((2*(D*b**3*(c + d*x)**(9/2)/(9*d**6) + (c + d*x)**(7/2)*(C*b**3*d
 + 3*D*a*b**2*d - 6*D*b**3*c)/(7*d**6) + (c + d*x)**(5/2)*(B*b**3*d**2 + 3*
C*a*b**2*d**2 - 5*C*b**3*c*d + 3*D*a**2*b*d**2 - 15*D*a*b**2*c*d + 15*D*b**
3*c**2)/(5*d**6) + (c + d*x)**(3/2)*(A*b**3*d**3 + 3*B*a*b**2*d**3 - 4*B*b*
*3*c*d**2 + 3*C*a**2*b*d**3 - 12*C*a*b**2*c*d**2 + 10*C*b**3*c**2*d + D*a**
```

```
3*d**3 - 12*D*a**2*b*c*d**2 + 30*D*a*b**2*c**2*d - 20*D*b**3*c**3)/(3*d**6)
 + sqrt(c + d*x)*(3*A*a*b**2*d**4 - 3*A*b**3*c*d**3 + 3*B*a**2*b*d**4 - 9*B
*a*b**2*c*d**3 + 6*B*b**3*c**2*d**2 + C*a**3*d**4 - 9*C*a**2*b*c*d**3 + 18*
C*a*b**2*c**2*d**2 - 10*C*b**3*c**3*d - 3*D*a**3*c*d**3 + 18*D*a**2*b*c**2*
d**2 - 30*D*a*b**2*c**3*d + 15*D*b**3*c**4)/d**6 - (a*d - b*c)**2*(3*A*b*d*
*3 + B*a*d**3 - 4*B*b*c*d**2 - 2*C*a*c*d**2 + 5*C*b*c**2*d + 3*D*a*c**2*d -
 6*D*b*c**3)/(d**6*sqrt(c + d*x)) + (a*d - b*c)**3*(-A*d**3 + B*c*d**2 - C*
c**2*d + D*c**3)/(3*d**6*(c + d*x)**(3/2)))/d, Ne(d, 0)), ((A*a**3*x + D*b*
*3*x**7/7 + x**6*(C*b**3 + 3*D*a*b**2)/6 + x**5*(B*b**3 + 3*C*a*b**2 + 3*D*
a**2*b)/5 + x**4*(A*b**3 + 3*B*a*b**2 + 3*C*a**2*b + D*a**3)/4 + x**3*(3*A*
a*b**2 + 3*B*a**2*b + C*a**3)/3 + x**2*(3*A*a**2*b + B*a**3)/2)/c**(5/2), T
rue))
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 627, normalized size of antiderivative = 1.44

$$\int \frac{(a+bx)^3\,(A+Bx+Cx^2+Dx^3)}{(c+dx)^{5/2}}\,dx = \frac{2\left(\frac{35\,(dx+c)^{\frac{9}{2}}Db^3-45\,(6\,Db^3c-(3\,Dab^2+Cb^3)\,d)\,(dx+c)^{\frac{7}{2}}+63\,(15\,Db^3c^2-5\,(3\,Dab^2+C}{}\right.}{}$$

```
[In] integrate((b*x+a)^3*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x, algorithm="maxima"
)
```

```
[Out] 2/315*((35*(d*x + c)^(9/2)*D*b^3 - 45*(6*D*b^3*c - (3*D*a*b^2 + C*b^3)*d)*(
d*x + c)^(7/2) + 63*(15*D*b^3*c^2 - 5*(3*D*a*b^2 + C*b^3)*c*d + (3*D*a^2*b
+ 3*C*a*b^2 + B*b^3)*d^2)*(d*x + c)^(5/2) - 105*(20*D*b^3*c^3 - 10*(3*D*a*b
^2 + C*b^3)*c^2*d + 4*(3*D*a^2*b + 3*C*a*b^2 + B*b^3)*c*d^2 - (D*a^3 + 3*C*
a^2*b + 3*B*a*b^2 + A*b^3)*d^3)*(d*x + c)^(3/2) + 315*(15*D*b^3*c^4 - 10*(3
*D*a*b^2 + C*b^3)*c^3*d + 6*(3*D*a^2*b + 3*C*a*b^2 + B*b^3)*c^2*d^2 - 3*(D*
a^3 + 3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c*d^3 + (C*a^3 + 3*B*a^2*b + 3*A*a*b^2
)*d^4)*sqrt(d*x + c))/d^6 - 105*(D*b^3*c^6 + A*a^3*d^6 - (3*D*a*b^2 + C*b^3
)*c^5*d + (3*D*a^2*b + 3*C*a*b^2 + B*b^3)*c^4*d^2 - (D*a^3 + 3*C*a^2*b + 3*
B*a*b^2 + A*b^3)*c^3*d^3 + (C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^2*d^4 - (B*a^3
 + 3*A*a^2*b)*c*d^5 - 3*(6*D*b^3*c^5 - 5*(3*D*a*b^2 + C*b^3)*c^4*d + 4*(3*D
*a^2*b + 3*C*a*b^2 + B*b^3)*c^3*d^2 - 3*(D*a^3 + 3*C*a^2*b + 3*B*a*b^2 + A*
b^3)*c^2*d^3 + 2*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c*d^4 - (B*a^3 + 3*A*a^2*b
)*d^5)*(d*x + c))/((d*x + c)^(3/2)*d^6))/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1030 vs. 2(412) = 824.

Time = 0.32 (sec) , antiderivative size = 1030, normalized size of antiderivative = 2.37

$$\int \frac{(a+bx)^3 (A+Bx+Cx^2+Dx^3)}{(c+dx)^{5/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((b*x+a)^3*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x, algorithm="giac")
```

```
[Out] 2/3*(18*(d*x + c)*D*b^3*c^5 - D*b^3*c^6 - 45*(d*x + c)*D*a*b^2*c^4*d - 15*(
d*x + c)*C*b^3*c^4*d + 3*D*a*b^2*c^5*d + C*b^3*c^5*d + 36*(d*x + c)*D*a^2*b
*c^3*d^2 + 36*(d*x + c)*C*a*b^2*c^3*d^2 + 12*(d*x + c)*B*b^3*c^3*d^2 - 3*D*
a^2*b*c^4*d^2 - 3*C*a*b^2*c^4*d^2 - B*b^3*c^4*d^2 - 9*(d*x + c)*D*a^3*c^2*d
^3 - 27*(d*x + c)*C*a^2*b*c^2*d^3 - 27*(d*x + c)*B*a*b^2*c^2*d^3 - 9*(d*x +
 c)*A*b^3*c^2*d^3 + D*a^3*c^3*d^3 + 3*C*a^2*b*c^3*d^3 + 3*B*a*b^2*c^3*d^3 +
 A*b^3*c^3*d^3 + 6*(d*x + c)*C*a^3*c*d^4 + 18*(d*x + c)*B*a^2*b*c*d^4 + 18*
(d*x + c)*A*a*b^2*c*d^4 - C*a^3*c^2*d^4 - 3*B*a^2*b*c^2*d^4 - 3*A*a*b^2*c^2
*d^4 - 3*(d*x + c)*B*a^3*d^5 - 9*(d*x + c)*A*a^2*b*d^5 + B*a^3*c*d^5 + 3*A*
a^2*b*c*d^5 - A*a^3*d^6)/((d*x + c)^(3/2)*d^7) + 2/315*(35*(d*x + c)^(9/2)*
D*b^3*d^56 - 270*(d*x + c)^(7/2)*D*b^3*c*d^56 + 945*(d*x + c)^(5/2)*D*b^3*c
^2*d^56 - 2100*(d*x + c)^(3/2)*D*b^3*c^3*d^56 + 4725*sqrt(d*x + c)*D*b^3*c^
4*d^56 + 135*(d*x + c)^(7/2)*D*a*b^2*d^57 + 45*(d*x + c)^(7/2)*C*b^3*d^57 -
 945*(d*x + c)^(5/2)*D*a*b^2*c*d^57 - 315*(d*x + c)^(5/2)*C*b^3*c*d^57 + 31
50*(d*x + c)^(3/2)*D*a*b^2*c^2*d^57 + 1050*(d*x + c)^(3/2)*C*b^3*c^2*d^57 -
 9450*sqrt(d*x + c)*D*a*b^2*c^3*d^57 - 3150*sqrt(d*x + c)*C*b^3*c^3*d^57 +
189*(d*x + c)^(5/2)*D*a^2*b*d^58 + 189*(d*x + c)^(5/2)*C*a*b^2*d^58 + 63*(d
*x + c)^(5/2)*B*b^3*d^58 - 1260*(d*x + c)^(3/2)*D*a^2*b*c*d^58 - 1260*(d*x
+ c)^(3/2)*C*a*b^2*c*d^58 - 420*(d*x + c)^(3/2)*B*b^3*c*d^58 + 5670*sqrt(d*
x + c)*D*a^2*b*c^2*d^58 + 5670*sqrt(d*x + c)*C*a*b^2*c^2*d^58 + 1890*sqrt(d
*x + c)*B*b^3*c^2*d^58 + 105*(d*x + c)^(3/2)*D*a^3*d^59 + 315*(d*x + c)^(3/
2)*C*a^2*b*d^59 + 315*(d*x + c)^(3/2)*B*a*b^2*d^59 + 105*(d*x + c)^(3/2)*A*
b^3*d^59 - 945*sqrt(d*x + c)*D*a^3*c*d^59 - 2835*sqrt(d*x + c)*C*a^2*b*c*d^
59 - 2835*sqrt(d*x + c)*B*a*b^2*c*d^59 - 945*sqrt(d*x + c)*A*b^3*c*d^59 + 3
15*sqrt(d*x + c)*C*a^3*d^60 + 945*sqrt(d*x + c)*B*a^2*b*d^60 + 945*sqrt(d*x
 + c)*A*a*b^2*d^60)/d^63
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+bx)^3\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx=\int \frac{{\left(a+b\,x\right)}^3\,\left(A+B\,x+C\,x^2+x^3\,D\right)}{{\left(c+d\,x\right)}^{5/2}}\,dx$$

```
[In] int(((a + b*x)^3*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(5/2),x)
[Out] int(((a + b*x)^3*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(5/2), x)
```

3.19 $\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 322

$$
\begin{aligned}
\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx &= -\frac{2(bc-ad)^2\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{3d^6(c+dx)^{3/2}} \\
&- \frac{2(bc-ad)\left(ad(2cCd-Bd^2-3c^2D)-b(4c^2Cd-3Bcd^2+2Ad^3-5c^3D)\right)}{d^6\sqrt{c+dx}} \\
&+ \frac{2(a^2d^2(Cd-3cD)-2abd(3cCd-Bd^2-6c^2D)+b^2(6c^2Cd-3Bcd^2+Ad^3-10c^3D))\sqrt{c+dx}}{d^6} \\
&+ \frac{2(a^2d^2D+2abd(Cd-4cD)-b^2(4cCd-Bd^2-10c^2D))\,(c+dx)^{3/2}}{3d^6} \\
&+ \frac{2b(bCd-5bcD+2adD)(c+dx)^{5/2}}{5d^6} + \frac{2b^2D(c+dx)^{7/2}}{7d^6}
\end{aligned}
$$

```
[Out] -2/3*(-a*d+b*c)^2*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)/d^6/(d*x+c)^(3/2)+2/3*(a^2*
d^2*D+2*a*b*d*(C*d-4*D*c)-b^2*(-B*d^2+4*C*c*d-10*D*c^2))*(d*x+c)^(3/2)/d^6+
2/5*b*(C*b*d+2*D*a*d-5*D*b*c)*(d*x+c)^(5/2)/d^6+2/7*b^2*D*(d*x+c)^(7/2)/d^6
-2*(-a*d+b*c)*(a*d*(-B*d^2+2*C*c*d-3*D*c^2)-b*(2*A*d^3-3*B*c*d^2+4*C*c^2*d-
5*D*c^3))/d^6/(d*x+c)^(1/2)+2*(a^2*d^2*(C*d-3*D*c)-2*a*b*d*(-B*d^2+3*C*c*d-
6*D*c^2)+b^2*(A*d^3-3*B*c*d^2+6*C*c^2*d-10*D*c^3))*(d*x+c)^(1/2)/d^6
```

Rubi [A] (verified)

Time = 0.16 (sec) , antiderivative size = 322, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.031, Rules used = {1634}

$$\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \frac{2\sqrt{c+dx}(a^2d^2(Cd-3cD)-2abd(-Bd^2-6c^2D+3cCd)+b^2(A}{d^6}$$

$$+\frac{2(c+dx)^{3/2}\left(a^2d^2D+2abd(Cd-4cD)-(b^2(-Bd^2-10c^2D+4cCd))\right)}{3d^6}$$

$$-\frac{2(bc-ad)\left(ad(-Bd^2-3c^2D+2cCd)-b(2Ad^3-3Bcd^2-5c^3D+4c^2Cd)\right)}{d^6\sqrt{c+dx}}$$

$$-\frac{2(bc-ad)^2\left(Ad^3-Bcd^2+c^3(-D)+c^2Cd\right)}{3d^6(c+dx)^{3/2}}$$

$$+\frac{2b(c+dx)^{5/2}(2adD-5bcD+bCd)}{5d^6}+\frac{2b^2D(c+dx)^{7/2}}{7d^6}$$

```
[In] Int[((a + b*x)^2*(A + B*x + C*x^2 + D*x^3))/(c + d*x)^(5/2),x]
```

```
[Out] (-2*(b*c - a*d)^2*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D))/(3*d^6*(c + d*x)^(3/
2)) - (2*(b*c - a*d)*(a*d*(2*c*C*d - B*d^2 - 3*c^2*D) - b*(4*c^2*C*d - 3*B*
c*d^2 + 2*A*d^3 - 5*c^3*D)))/(d^6*Sqrt[c + d*x]) + (2*(a^2*d^2*(C*d - 3*c*D
) - 2*a*b*d*(3*c*C*d - B*d^2 - 6*c^2*D) + b^2*(6*c^2*C*d - 3*B*c*d^2 + A*d^
3 - 10*c^3*D))*Sqrt[c + d*x])/d^6 + (2*(a^2*d^2*D + 2*a*b*d*(C*d - 4*c*D) -
 b^2*(4*c*C*d - B*d^2 - 10*c^2*D))*(c + d*x)^(3/2))/(3*d^6) + (2*b*(b*C*d -
 5*b*c*D + 2*a*d*D)*(c + d*x)^(5/2))/(5*d^6) + (2*b^2*D*(c + d*x)^(7/2))/(7
*d^6)
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

integral

$$= \int \left(\frac{(-bc+ad)^2\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{d^5(c+dx)^{5/2}} + \frac{(bc-ad)\left(ad(2cCd-Bd^2-3c^2D)-b(4c^2Cd-3Bcd^2+2Ad^3-5c^3D)\right)}{d^5(c+dx)^{3/2}} + \frac{a^2d^2(Cd-3cD)-2abd(3cCd-Bd^2-6c^2D)+b^2(6c^2Cd-3Bcd^2+Ad^3-10c^3D)}{d^5\sqrt{c+dx}} + \frac{\left(a^2d^2D+2abd(Cd-4cD)-b^2(4cCd-Bd^2-10c^2D)\right)\sqrt{c+dx}}{d^5} + \frac{b(bCd-5bcD+2adD)(c+dx)^{3/2}}{d^5} + \frac{b^2D(c+dx)^{5/2}}{d^5} \right) dx$$

$$= -\frac{2(bc-ad)^2\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{3d^6(c+dx)^{3/2}} - \frac{2(bc-ad)\left(ad(2cCd-Bd^2-3c^2D)-b(4c^2Cd-3Bcd^2+2Ad^3-5c^3D)\right)}{d^6\sqrt{c+dx}} + \frac{2\left(a^2d^2(Cd-3cD)-2abd(3cCd-Bd^2-6c^2D)+b^2(6c^2Cd-3Bcd^2+Ad^3-10c^3D)\right)\sqrt{c+dx}}{d^6} + \frac{2\left(a^2d^2D+2abd(Cd-4cD)-b^2(4cCd-Bd^2-10c^2D)\right)(c+dx)^{3/2}}{3d^6} + \frac{2b(bCd-5bcD+2adD)(c+dx)^{5/2}}{5d^6} + \frac{2b^2D(c+dx)^{7/2}}{7d^6}$$

Mathematica [A] (verified)

Time = 0.36 (sec) , antiderivative size = 317, normalized size of antiderivative = 0.98

$$\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \frac{-70a^2d^2(16c^3D-8c^2d(C-3Dx)+2cd^2(B+3x(-2C+Dx))}{}$$

```
[In] Integrate[((a + b*x)^2*(A + B*x + C*x^2 + D*x^3))/(c + d*x)^(5/2),x]
```

```
[Out] (-70*a^2*d^2*(16*c^3*D - 8*c^2*d*(C - 3*D*x) + 2*c*d^2*(B + 3*x*(-2*C + D*x
)) + d^3*(A + 3*B*x - x^2*(3*C + D*x))) + 28*a*b*d*(128*c^4*D + c^3*(-80*C*
d + 192*d*D*x) + 8*c^2*d^2*(5*B + 3*x*(-5*C + 2*D*x)) + d^4*x*(-15*A + x*(1
5*B + 5*C*x + 3*D*x^2)) - 2*c*d^3*(5*A + x*(-30*B + 15*C*x + 4*D*x^2))) + 2
*b^2*(-1280*c^5*D + 128*c^4*d*(7*C - 15*D*x) - 16*c^3*d^2*(35*B + 6*x*(-14*
C + 5*D*x)) + d^5*x^2*(105*A + x*(35*B + 3*x*(7*C + 5*D*x))) + 8*c^2*d^3*(3
5*A + x*(-105*B + 2*x*(21*C + 5*D*x))) - 2*c*d^4*x*(-210*A + x*(105*B + x*(
28*C + 15*D*x)))))/(105*d^6*(c + d*x)^(3/2))
```

Maple [A] (verified)

Time = 1.73 (sec) , antiderivative size = 288, normalized size of antiderivative = 0.89

method	result
pseudoelliptic	$-\frac{2\left(\left(\left(-\frac{3}{7}Dx^5-\frac{3}{5}Cx^4-3Ax^2-x^3B\right)b^2+6ax\left(-\frac{1}{5}Dx^3-\frac{1}{3}Cx^2-Bx+A\right)b+a^2\left(-Dx^3-3Cx^2+3Bx+A\right)\right)d^5+4\left(-3x(-\right.\right.}{}$
gosper	$-\frac{2(-15Db^2x^5d^5-21Cb^2d^5x^4-42Dabd^5x^4+30Db^2cd^4x^4-35Bb^2d^5x^3-70Cabd^5x^3+56Cb^2cd^4x^3-35Da^2d^5x^3+112}{}$
trager	$-\frac{2(-15Db^2x^5d^5-21Cb^2d^5x^4-42Dabd^5x^4+30Db^2cd^4x^4-35Bb^2d^5x^3-70Cabd^5x^3+56Cb^2cd^4x^3-35Da^2d^5x^3+112}{}$
derivativedivides	$\frac{\frac{2Db^2(dx+c)^{\frac{7}{2}}}{7}+\frac{2Cb^2d(dx+c)^{\frac{5}{2}}}{5}+\frac{4Dabd(dx+c)^{\frac{5}{2}}}{5}-2Db^2c(dx+c)^{\frac{5}{2}}+\frac{2Bb^2d^2(dx+c)^{\frac{3}{2}}}{3}+\frac{4Cabd^2(dx+c)^{\frac{3}{2}}}{3}-\frac{8Cb^2cd(dx+c)^{\frac{3}{2}}}{3}+}{}$
default	$\frac{\frac{2Db^2(dx+c)^{\frac{7}{2}}}{7}+\frac{2Cb^2d(dx+c)^{\frac{5}{2}}}{5}+\frac{4Dabd(dx+c)^{\frac{5}{2}}}{5}-2Db^2c(dx+c)^{\frac{5}{2}}+\frac{2Bb^2d^2(dx+c)^{\frac{3}{2}}}{3}+\frac{4Cabd^2(dx+c)^{\frac{3}{2}}}{3}-\frac{8Cb^2cd(dx+c)^{\frac{3}{2}}}{3}+}{}$

```
[In] int((b*x+a)^2*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x,method=_RETURNVERBOSE)
```

```
[Out] -2/3/(d*x+c)^(3/2)*(((-3/7*D*x^5-3/5*C*x^4-3*A*x^2-x^3*B)*b^2+6*a*x*(-1/5*D
*x^3-1/3*C*x^2-B*x+A)*b+a^2*(-D*x^3-3*C*x^2+3*B*x+A))*d^5+4*(-3*x*(-1/14*D*
x^3-2/15*C*x^2-1/2*B*x+A)*b^2+a*(4/5*D*x^3+3*C*x^2-6*B*x+A)*b+1/2*a^2*(3*D*
x^2-6*C*x+B))*c*d^4-8*((2/7*D*x^3+6/5*C*x^2-3*B*x+A)*b^2+2*a*(6/5*D*x^2-3*C
*x+B)*b+a^2*(-3*D*x+C))*c^2*d^3+16*c^3*((6/7*D*x^2-12/5*C*x+B)*b^2+2*(-12/5
*D*x+C)*a*b+D*a^2)*d^2-128/5*((-15/7*D*x+C)*b+2*D*a)*b*c^4*d+256/7*D*b^2*c^
5)/d^6
```

Fricas [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 431, normalized size of antiderivative = 1.34

$$\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \frac{2\,(15\,Db^2d^5x^5-1280\,Db^2c^5-35\,Aa^2d^5+280\,(Ca^2+2\,Bab+Ab}{}$$

```
[In] integrate((b*x+a)^2*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x, algorithm="fricas"
)
```

```
[Out] 2/105*(15*D*b^2*d^5*x^5 - 1280*D*b^2*c^5 - 35*A*a^2*d^5 + 280*(C*a^2 + 2*B*
a*b + A*b^2)*c^2*d^3 - 70*(B*a^2 + 2*A*a*b)*c*d^4 - 3*(10*D*b^2*c*d^4 - 7*(
2*D*a*b + C*b^2)*d^5)*x^4 + (80*D*b^2*c^2*d^3 + 35*(D*a^2 + 2*C*a*b + B*b^2
)*d^5 - 56*(2*D*a*b*c + C*b^2*c)*d^4)*x^3 - 560*(D*a^2*c^3 + (2*C*a*b + B*b
^2)*c^3)*d^2 - 3*(160*D*b^2*c^3*d^2 - 35*(C*a^2 + 2*B*a*b + A*b^2)*d^5 + 70
*(D*a^2*c + (2*C*a*b + B*b^2)*c)*d^4 - 112*(2*D*a*b*c^2 + C*b^2*c^2)*d^3)*x
```

```
^2 + 896*(2*D*a*b*c^4 + C*b^2*c^4)*d - 3*(640*D*b^2*c^4*d - 140*(C*a^2 + 2*
B*a*b + A*b^2)*c*d^4 + 35*(B*a^2 + 2*A*a*b)*d^5 + 280*(D*a^2*c^2 + (2*C*a*b
 + B*b^2)*c^2)*d^3 - 448*(2*D*a*b*c^3 + C*b^2*c^3)*d^2)*x)*sqrt(d*x + c)/(d
^8*x^2 + 2*c*d^7*x + c^2*d^6)
```

Sympy [A] (verification not implemented)

Time = 29.35 (sec) , antiderivative size = 476, normalized size of antiderivative = 1.48

$$\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \begin{cases} \frac{2\left(\frac{Db^2(c+dx)^{\frac{7}{2}}}{7d^5}+\frac{(c+dx)^{\frac{5}{2}}\left(Cb^2d+2Dabd-5Db^2c\right)}{5d^5}+\frac{(c+dx)^{\frac{3}{2}}\left(Bb^2d^2+2Cabd^2-4Cb^2cd+Dc\right.}{3d^5}\right.}{} \\ \frac{Aa^2x+\frac{Db^2x^6}{6}+\frac{x^5\left(Cb^2+2Dab\right)}{5}+\frac{x^4\left(Bb^2+2Cab+Da^2\right)}{4}+\frac{x^3\left(Ab^2+2Bab+Ca^2\right)}{3}+\frac{x^2\cdot\left(2Aa\right.}{2}}{c^{\frac{5}{2}}} \end{cases}$$

```
[In] integrate((b*x+a)**2*(D*x**3+C*x**2+B*x+A)/(d*x+c)**(5/2),x)
```

```
[Out] Piecewise((2*(D*b**2*(c + d*x)**(7/2)/(7*d**5) + (c + d*x)**(5/2)*(C*b**2*d
 + 2*D*a*b*d - 5*D*b**2*c)/(5*d**5) + (c + d*x)**(3/2)*(B*b**2*d**2 + 2*C*a
*b*d**2 - 4*C*b**2*c*d + D*a**2*d**2 - 8*D*a*b*c*d + 10*D*b**2*c**2)/(3*d**
5) + sqrt(c + d*x)*(A*b**2*d**3 + 2*B*a*b*d**3 - 3*B*b**2*c*d**2 + C*a**2*d
**3 - 6*C*a*b*c*d**2 + 6*C*b**2*c**2*d - 3*D*a**2*c*d**2 + 12*D*a*b*c**2*d
- 10*D*b**2*c**3)/d**5 - (a*d - b*c)*(2*A*b*d**3 + B*a*d**3 - 3*B*b*c*d**2
- 2*C*a*c*d**2 + 4*C*b*c**2*d + 3*D*a*c**2*d - 5*D*b*c**3)/(d**5*sqrt(c + d
*x)) + (a*d - b*c)**2*(-A*d**3 + B*c*d**2 - C*c**2*d + D*c**3)/(3*d**5*(c +
 d*x)**(3/2)))/d, Ne(d, 0)), ((A*a**2*x + D*b**2*x**6/6 + x**5*(C*b**2 + 2*
D*a*b)/5 + x**4*(B*b**2 + 2*C*a*b + D*a**2)/4 + x**3*(A*b**2 + 2*B*a*b + C*
a**2)/3 + x**2*(2*A*a*b + B*a**2)/2)/c**(5/2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.21 (sec) , antiderivative size = 393, normalized size of antiderivative = 1.22

$$\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \frac{2\left(\frac{15\,(dx+c)^{\frac{7}{2}}\,Db^2-21\,\left(5\,Db^2c-\left(2\,Dab+Cb^2\right)d\right)(dx+c)^{\frac{5}{2}}+35\,\left(10\,Db^2c^2-4\,\left(2\,Dab+C\right.\right.}{}\right.}{}$$

```
[In] integrate((b*x+a)^2*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x, algorithm="maxima"
)
```

```
[Out] 2/105*((15*(d*x + c)^(7/2)*D*b^2 - 21*(5*D*b^2*c - (2*D*a*b + C*b^2)*d)*(d*
x + c)^(5/2) + 35*(10*D*b^2*c^2 - 4*(2*D*a*b + C*b^2)*c*d + (D*a^2 + 2*C*a*
b + B*b^2)*d^2)*(d*x + c)^(3/2) - 105*(10*D*b^2*c^3 - 6*(2*D*a*b + C*b^2)*c
```

```
^2*d + 3*(D*a^2 + 2*C*a*b + B*b^2)*c*d^2 - (C*a^2 + 2*B*a*b + A*b^2)*d^3)*s
qrt(d*x + c))/d^5 + 35*(D*b^2*c^5 - A*a^2*d^5 - (2*D*a*b + C*b^2)*c^4*d + (
D*a^2 + 2*C*a*b + B*b^2)*c^3*d^2 - (C*a^2 + 2*B*a*b + A*b^2)*c^2*d^3 + (B*a
^2 + 2*A*a*b)*c*d^4 - 3*(5*D*b^2*c^4 - 4*(2*D*a*b + C*b^2)*c^3*d + 3*(D*a^2
 + 2*C*a*b + B*b^2)*c^2*d^2 - 2*(C*a^2 + 2*B*a*b + A*b^2)*c*d^3 + (B*a^2 +
2*A*a*b)*d^4)*(d*x + c))/((d*x + c)^(3/2)*d^5))/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 622 vs. 2(302) = 604.

Time = 0.30 (sec) , antiderivative size = 622, normalized size of antiderivative = 1.93

$$\int \frac{(a+bx)^2\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx =$$

$$-\frac{2\left(15\,(dx+c)Db^2c^4-Db^2c^5-24\,(dx+c)Dabc^3d-12\,(dx+c)Cb^2c^3d+2\,Dabc^4d+Cb^2c^4d+9\,(dx+c)\right.}{}$$

$$+\frac{2\left(15\,(dx+c)^{\frac{7}{2}}Db^2d^{36}-105\,(dx+c)^{\frac{5}{2}}Db^2cd^{36}+350\,(dx+c)^{\frac{3}{2}}Db^2c^2d^{36}-1050\,\sqrt{dx+c}Db^2c^3d^{36}+42\,(dx\right.}{}$$

```
[In] integrate((b*x+a)^2*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x, algorithm="giac")
```

```
[Out] -2/3*(15*(d*x + c)*D*b^2*c^4 - D*b^2*c^5 - 24*(d*x + c)*D*a*b*c^3*d - 12*(d
*x + c)*C*b^2*c^3*d + 2*D*a*b*c^4*d + C*b^2*c^4*d + 9*(d*x + c)*D*a^2*c^2*d
^2 + 18*(d*x + c)*C*a*b*c^2*d^2 + 9*(d*x + c)*B*b^2*c^2*d^2 - D*a^2*c^3*d^2
 - 2*C*a*b*c^3*d^2 - B*b^2*c^3*d^2 - 6*(d*x + c)*C*a^2*c*d^3 - 12*(d*x + c)
*B*a*b*c*d^3 - 6*(d*x + c)*A*b^2*c*d^3 + C*a^2*c^2*d^3 + 2*B*a*b*c^2*d^3 +
A*b^2*c^2*d^3 + 3*(d*x + c)*B*a^2*d^4 + 6*(d*x + c)*A*a*b*d^4 - B*a^2*c*d^4
 - 2*A*a*b*c*d^4 + A*a^2*d^5)/((d*x + c)^(3/2)*d^6) + 2/105*(15*(d*x + c)^(
7/2)*D*b^2*d^36 - 105*(d*x + c)^(5/2)*D*b^2*c*d^36 + 350*(d*x + c)^(3/2)*D*
b^2*c^2*d^36 - 1050*sqrt(d*x + c)*D*b^2*c^3*d^36 + 42*(d*x + c)^(5/2)*D*a*b
*d^37 + 21*(d*x + c)^(5/2)*C*b^2*d^37 - 280*(d*x + c)^(3/2)*D*a*b*c*d^37 -
140*(d*x + c)^(3/2)*C*b^2*c*d^37 + 1260*sqrt(d*x + c)*D*a*b*c^2*d^37 + 630*
sqrt(d*x + c)*C*b^2*c^2*d^37 + 35*(d*x + c)^(3/2)*D*a^2*d^38 + 70*(d*x + c)
^(3/2)*C*a*b*d^38 + 35*(d*x + c)^(3/2)*B*b^2*d^38 - 315*sqrt(d*x + c)*D*a^2
*c*d^38 - 630*sqrt(d*x + c)*C*a*b*c*d^38 - 315*sqrt(d*x + c)*B*b^2*c*d^38 +
 105*sqrt(d*x + c)*C*a^2*d^39 + 210*sqrt(d*x + c)*B*a*b*d^39 + 105*sqrt(d*x
 + c)*A*b^2*d^39)/d^42
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+bx)^2\,(A+Bx+Cx^2+Dx^3)}{(c+dx)^{5/2}}\,dx=\int \frac{{\left(a+b\,x\right)}^2\,\left(A+B\,x+C\,x^2+x^3\,D\right)}{{\left(c+d\,x\right)}^{5/2}}\,dx$$

```
[In] int(((a + b*x)^2*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(5/2),x)
[Out] int(((a + b*x)^2*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(5/2), x)
```

3.20 $\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx$

Optimal result

Integrand size = 30, antiderivative size = 210

$$
\begin{aligned}
\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx &= \frac{2(bc-ad)\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{3d^5(c+dx)^{3/2}} \\
&+ \frac{2(ad(2cCd-Bd^2-3c^2D)-b(3c^2Cd-2Bcd^2+Ad^3-4c^3D))}{d^5\sqrt{c+dx}} \\
&+ \frac{2(ad(Cd-3cD)-b(3cCd-Bd^2-6c^2D))\sqrt{c+dx}}{d^5} \\
&+ \frac{2(bCd-4bcD+adD)(c+dx)^{3/2}}{3d^5} + \frac{2bD(c+dx)^{5/2}}{5d^5}
\end{aligned}
$$

```
[Out] 2/3*(-a*d+b*c)*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)/d^5/(d*x+c)^(3/2)+2/3*(C*b*d+D
*a*d-4*D*b*c)*(d*x+c)^(3/2)/d^5+2/5*b*D*(d*x+c)^(5/2)/d^5+2*(a*d*(-B*d^2+2*
C*c*d-3*D*c^2)-b*(A*d^3-2*B*c*d^2+3*C*c^2*d-4*D*c^3))/d^5/(d*x+c)^(1/2)+2*(
a*d*(C*d-3*D*c)-b*(-B*d^2+3*C*c*d-6*D*c^2))*(d*x+c)^(1/2)/d^5
```

Rubi [A] (verified)

Time = 0.12 (sec) , antiderivative size = 210, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.033, Rules used

$= \{1634\}$

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \frac{2(ad(-Bd^2-3c^2D+2cCd)-b(Ad^3-2Bcd^2-4c^3D+3c^2Cd))}{d^5\sqrt{c+dx}}$$
$$+\frac{2(bc-ad)\left(Ad^3-Bcd^2+c^3(-D)+c^2Cd\right)}{3d^5(c+dx)^{3/2}}$$
$$+\frac{2\sqrt{c+dx}(ad(Cd-3cD)-b(-Bd^2-6c^2D+3cCd))}{d^5}$$
$$+\frac{2(c+dx)^{3/2}(adD-4bcD+bCd)}{3d^5}+\frac{2bD(c+dx)^{5/2}}{5d^5}$$

```
[In] Int[((a + b*x)*(A + B*x + C*x^2 + D*x^3))/(c + d*x)^(5/2),x]
```

```
[Out] (2*(b*c - a*d)*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D))/(3*d^5*(c + d*x)^(3/2))
 + (2*(a*d*(2*c*C*d - B*d^2 - 3*c^2*D) - b*(3*c^2*C*d - 2*B*c*d^2 + A*d^3 -
 4*c^3*D)))/(d^5*Sqrt[c + d*x]) + (2*(a*d*(C*d - 3*c*D) - b*(3*c*C*d - B*d^
2 - 6*c^2*D))*Sqrt[c + d*x])/d^5 + (2*(b*C*d - 4*b*c*D + a*d*D)*(c + d*x)^(
3/2))/(3*d^5) + (2*b*D*(c + d*x)^(5/2))/(5*d^5)
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \left(\frac{(-bc+ad)\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{d^4(c+dx)^{5/2}} + \frac{-ad(2cCd-Bd^2-3c^2D)+b(3c^2Cd-2Bcd^2+Ad^3-4c^3D)}{d^4(c+dx)^{3/2}} + \frac{ad(Cd-3cD)-b(3cCd-Bd^2-6c^2D)}{d^4\sqrt{c+dx}} + \frac{(bCd-4bcD+adD)\sqrt{c+dx}}{d^4} + \frac{bD(c+dx)^{3/2}}{d^4}\right) dx \\
&= \frac{2(bc-ad)\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{3d^5(c+dx)^{3/2}} + \frac{2(ad(2cCd-Bd^2-3c^2D)-b(3c^2Cd-2Bcd^2+Ad^3-4c^3D))}{d^5\sqrt{c+dx}} \\
&\quad + \frac{2(ad(Cd-3cD)-b(3cCd-Bd^2-6c^2D))\sqrt{c+dx}}{d^5} + \frac{2(bCd-4bcD+adD)(c+dx)^{3/2}}{3d^5} + \frac{2bD(c+dx)^{5/2}}{5d^5}
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.17 (sec) , antiderivative size = 177, normalized size of antiderivative = 0.84

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \frac{2(-5ad(16c^3D-8c^2d(C-3Dx)+2cd^2(B+3x(-2C+Dx))+d}{}$$

```
[In] Integrate[((a + b*x)*(A + B*x + C*x^2 + D*x^3))/(c + d*x)^(5/2),x]
```

```
[Out] (2*(-5*a*d*(16*c^3*D - 8*c^2*d*(C - 3*D*x) + 2*c*d^2*(B + 3*x*(-2*C + D*x))
 + d^3*(A + 3*B*x - x^2*(3*C + D*x))) + b*(128*c^4*D + c^3*(-80*C*d + 192*d
*D*x) + 8*c^2*d^2*(5*B + 3*x*(-5*C + 2*D*x)) + d^4*x*(-15*A + x*(15*B + 5*C
*x + 3*D*x^2)) - 2*c*d^3*(5*A + x*(-30*B + 15*C*x + 4*D*x^2)))))/(15*d^5*(c
 + d*x)^(3/2))
```

Maple [A] (verified)

Time = 1.68 (sec) , antiderivative size = 168, normalized size of antiderivative = 0.80

method	result
pseudoelliptic	$-\frac{2\left(\left(-\frac{3Db\,x^4}{5}+(-Cb-Da)x^3+(-3Bb-3Ca)x^2+(3Ab+3Ba)x+Aa\right)d^4+2\left(\frac{4Db\,x^3}{5}+(3Cb+3Da)x^2+(-6Bb-6Ca)x+A\right.}{3(dx+c)^{\frac{3}{2}}\,d^5}$
gosper	$-\frac{2(-3Db\,x^4d^4-5Cb\,d^4x^3-5Da\,d^4x^3+8Dbc\,d^3x^3-15Bb\,d^4x^2-15Ca\,d^4x^2+30Cbc\,d^3x^2+30Dac\,d^3x^2-48Db\,c^2d^2x^2+15}{}$
trager	$-\frac{2(-3Db\,x^4d^4-5Cb\,d^4x^3-5Da\,d^4x^3+8Dbc\,d^3x^3-15Bb\,d^4x^2-15Ca\,d^4x^2+30Cbc\,d^3x^2+30Dac\,d^3x^2-48Db\,c^2d^2x^2+15}{}$
derivativedivides	$\frac{\frac{2Db(dx+c)^{\frac{5}{2}}}{5}+\frac{2Cbd(dx+c)^{\frac{3}{2}}}{3}+\frac{2Dad(dx+c)^{\frac{3}{2}}}{3}-\frac{8Dbc(dx+c)^{\frac{3}{2}}}{3}+2Bb\,d^2\sqrt{dx+c}+2Ca\,d^2\sqrt{dx+c}-6Cbcd\sqrt{dx+c}-6Dacd\sqrt{dx+}}{}$
default	$\frac{\frac{2Db(dx+c)^{\frac{5}{2}}}{5}+\frac{2Cbd(dx+c)^{\frac{3}{2}}}{3}+\frac{2Dad(dx+c)^{\frac{3}{2}}}{3}-\frac{8Dbc(dx+c)^{\frac{3}{2}}}{3}+2Bb\,d^2\sqrt{dx+c}+2Ca\,d^2\sqrt{dx+c}-6Cbcd\sqrt{dx+c}-6Dacd\sqrt{dx+}}{}$

```
[In] int((b*x+a)*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x,method=_RETURNVERBOSE)
```

```
[Out] -2/3/(d*x+c)^(3/2)*((-3/5*D*b*x^4+(-C*b-D*a)*x^3+(-3*B*b-3*C*a)*x^2+(3*A*b+
3*B*a)*x+A*a)*d^4+2*(4/5*D*b*x^3+(3*C*b+3*D*a)*x^2+(-6*B*b-6*C*a)*x+A*b+B*a
)*c*d^3-8*(6/5*D*b*x^2+(-3*C*b-3*D*a)*x+B*b+C*a)*c^2*d^2+16*(-12/5*D*b*x+C*
b+D*a)*c^3*d-128/5*D*b*c^4)/d^5
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 225, normalized size of antiderivative = 1.07

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \frac{2\,(3\,Dbd^4x^4+128\,Dbc^4-5\,Aad^4+40\,(Ca+Bb)c^2d^2-10\,(Ba+}{}$$

```
[In] integrate((b*x+a)*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x, algorithm="fricas")
```

```
[Out] 2/15*(3*D*b*d^4*x^4 + 128*D*b*c^4 - 5*A*a*d^4 + 40*(C*a + B*b)*c^2*d^2 - 10
*(B*a + A*b)*c*d^3 - (8*D*b*c*d^3 - 5*(D*a + C*b)*d^4)*x^3 + 3*(16*D*b*c^2*
d^2 + 5*(C*a + B*b)*d^4 - 10*(D*a*c + C*b*c)*d^3)*x^2 - 80*(D*a*c^3 + C*b*c
^3)*d + 3*(64*D*b*c^3*d + 20*(C*a + B*b)*c*d^3 - 5*(B*a + A*b)*d^4 - 40*(D*
a*c^2 + C*b*c^2)*d^2)*x)*sqrt(d*x + c)/(d^7*x^2 + 2*c*d^6*x + c^2*d^5)
```

Sympy [A] (verification not implemented)

Time = 8.27 (sec) , antiderivative size = 282, normalized size of antiderivative = 1.34

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \begin{cases} \frac{2\left(\frac{Db(c+dx)^{\frac{5}{2}}}{5d^4}+\frac{(c+dx)^{\frac{3}{2}}(Cbd+Dad-4Dbc)}{3d^4}+\frac{\sqrt{c+dx}\left(Bbd^2+Cad^2-3Cbcd-3Dacd+6Dbc^2\right)}{d^4}-\right.}{} \\ \frac{Aax+\frac{Dbx^5}{5}+\frac{x^4(Cb+Da)}{4}+\frac{x^3(Bb+Ca)}{3}+\frac{x^2(Ab+Ba)}{2}}{c^{\frac{5}{2}}} \end{cases}$$

```
[In] integrate((b*x+a)*(D*x**3+C*x**2+B*x+A)/(d*x+c)**(5/2),x)
```

```
[Out] Piecewise((2*(D*b*(c + d*x)**(5/2)/(5*d**4) + (c + d*x)**(3/2)*(C*b*d + D*a
*d - 4*D*b*c)/(3*d**4) + sqrt(c + d*x)*(B*b*d**2 + C*a*d**2 - 3*C*b*c*d - 3
*D*a*c*d + 6*D*b*c**2)/d**4 - (A*b*d**3 + B*a*d**3 - 2*B*b*c*d**2 - 2*C*a*c
*d**2 + 3*C*b*c**2*d + 3*D*a*c**2*d - 4*D*b*c**3)/(d**4*sqrt(c + d*x)) + (a
*d - b*c)*(-A*d**3 + B*c*d**2 - C*c**2*d + D*c**3)/(3*d**4*(c + d*x)**(3/2)
))/d, Ne(d, 0)), ((A*a*x + D*b*x**5/5 + x**4*(C*b + D*a)/4 + x**3*(B*b + C*
a)/3 + x**2*(A*b + B*a)/2)/c**(5/2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 204, normalized size of antiderivative = 0.97

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \frac{2\left(\frac{3\,(dx+c)^{\frac{5}{2}}\,Db-5\,(4\,Dbc-(Da+Cb)d)(dx+c)^{\frac{3}{2}}+15\,\left(6\,Dbc^2-3\,(Da+Cb)cd+(Ca+Bb)\right.}{d^4}\right.}{}$$

```
[In] integrate((b*x+a)*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x, algorithm="maxima")
```

```
[Out] 2/15*((3*(d*x + c)^(5/2)*D*b - 5*(4*D*b*c - (D*a + C*b)*d)*(d*x + c)^(3/2)
+ 15*(6*D*b*c^2 - 3*(D*a + C*b)*c*d + (C*a + B*b)*d^2)*sqrt(d*x + c))/d^4 -
 5*(D*b*c^4 + A*a*d^4 - (D*a + C*b)*c^3*d + (C*a + B*b)*c^2*d^2 - (B*a + A*
b)*c*d^3 - 3*(4*D*b*c^3 - 3*(D*a + C*b)*c^2*d + 2*(C*a + B*b)*c*d^2 - (B*a
+ A*b)*d^3)*(d*x + c))/((d*x + c)^(3/2)*d^4))/d
```

Giac [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 302, normalized size of antiderivative = 1.44

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \frac{2\left(12\,(dx+c)Dbc^3 - Dbc^4 - 9\,(dx+c)Dac^2d - 9\,(dx+c)Cbc^2d +\right.}{}$$

$$+\frac{2\left(3\,(dx+c)^{\frac{5}{2}}Dbd^{20} - 20\,(dx+c)^{\frac{3}{2}}Dbcd^{20} + 90\,\sqrt{dx+c}Dbc^2d^{20} + 5\,(dx+c)^{\frac{3}{2}}Dad^{21} + 5\,(dx+c)^{\frac{3}{2}}Cbd^{21} -\right.}{15\,d^{25}}$$

```
[In] integrate((b*x+a)*(D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x, algorithm="giac")
```

```
[Out] 2/3*(12*(d*x + c)*D*b*c^3 - D*b*c^4 - 9*(d*x + c)*D*a*c^2*d - 9*(d*x + c)*C
*b*c^2*d + D*a*c^3*d + C*b*c^3*d + 6*(d*x + c)*C*a*c*d^2 + 6*(d*x + c)*B*b*
c*d^2 - C*a*c^2*d^2 - B*b*c^2*d^2 - 3*(d*x + c)*B*a*d^3 - 3*(d*x + c)*A*b*d
^3 + B*a*c*d^3 + A*b*c*d^3 - A*a*d^4)/((d*x + c)^(3/2)*d^5) + 2/15*(3*(d*x
+ c)^(5/2)*D*b*d^20 - 20*(d*x + c)^(3/2)*D*b*c*d^20 + 90*sqrt(d*x + c)*D*b*
c^2*d^20 + 5*(d*x + c)^(3/2)*D*a*d^21 + 5*(d*x + c)^(3/2)*C*b*d^21 - 45*sqr
t(d*x + c)*D*a*c*d^21 - 45*sqrt(d*x + c)*C*b*c*d^21 + 15*sqrt(d*x + c)*C*a*
d^22 + 15*sqrt(d*x + c)*B*b*d^22)/d^25
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+bx)\left(A+Bx+Cx^2+Dx^3\right)}{(c+dx)^{5/2}}\,dx = \int \frac{(a+b\,x)\,\left(A+B\,x+C\,x^2+x^3\,D\right)}{{(c+d\,x)}^{5/2}} \,d x$$

```
[In] int(((a + b*x)*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(5/2),x)
```

```
[Out] int(((a + b*x)*(A + B*x + C*x^2 + x^3*D))/(c + d*x)^(5/2), x)
```

3.21 $\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{5/2}}\,dx$

Optimal result

Integrand size = 25, antiderivative size = 113

$$\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{5/2}}\,dx = -\frac{2(c^2Cd-Bcd^2+Ad^3-c^3D)}{3d^4(c+dx)^{3/2}} + \frac{2(2cCd-Bd^2-3c^2D)}{d^4\sqrt{c+dx}} + \frac{2(Cd-3cD)\sqrt{c+dx}}{d^4} + \frac{2D(c+dx)^{3/2}}{3d^4}$$

```
[Out] -2/3*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)/d^4/(d*x+c)^(3/2)+2/3*D*(d*x+c)^(3/2)/d^
4+2*(-B*d^2+2*C*c*d-3*D*c^2)/d^4/(d*x+c)^(1/2)+2*(C*d-3*D*c)*(d*x+c)^(1/2)/
d^4
```

Rubi [A] (verified)

Time = 0.04 (sec) , antiderivative size = 113, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.040, Rules used = {1864}

$$\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{5/2}}\,dx = -\frac{2(Ad^3-Bcd^2+c^3(-D)+c^2Cd)}{3d^4(c+dx)^{3/2}} + \frac{2(-Bd^2-3c^2D+2cCd)}{d^4\sqrt{c+dx}} + \frac{2\sqrt{c+dx}(Cd-3cD)}{d^4} + \frac{2D(c+dx)^{3/2}}{3d^4}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/(c + d*x)^(5/2),x]
```

```
[Out] (-2*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D))/(3*d^4*(c + d*x)^(3/2)) + (2*(2*c*
C*d - B*d^2 - 3*c^2*D))/(d^4*Sqrt[c + d*x]) + (2*(C*d - 3*c*D)*Sqrt[c + d*x
])/d^4 + (2*D*(c + d*x)^(3/2))/(3*d^4)
```

Rule 1864

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_.))^(p_.), x_Symbol] :> Int[ExpandIntegrand
[Pq*(a + b*x^n)^p, x], x] /; FreeQ[{a, b, n}, x] && PolyQ[Pq, x] && (IGtQ[p
, 0] || EqQ[n, 1])
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \left(\frac{c^2Cd - Bcd^2 + Ad^3 - c^3D}{d^3(c+dx)^{5/2}} + \frac{-2cCd + Bd^2 + 3c^2D}{d^3(c+dx)^{3/2}} + \frac{Cd - 3cD}{d^3\sqrt{c+dx}} + \frac{D\sqrt{c+dx}}{d^3} \right) dx \\
&= -\frac{2(c^2Cd - Bcd^2 + Ad^3 - c^3D)}{3d^4(c+dx)^{3/2}} + \frac{2(2cCd - Bd^2 - 3c^2D)}{d^4\sqrt{c+dx}} + \frac{2(Cd - 3cD)\sqrt{c+dx}}{d^4} + \frac{2D(c+dx)^{3/2}}{3d^4}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.07 (sec) , antiderivative size = 75, normalized size of antiderivative = 0.66

$$
\int \frac{A + Bx + Cx^2 + Dx^3}{(c+dx)^{5/2}}\, dx = -\frac{2(16c^3D - 8c^2d(C - 3Dx) + 2cd^2(B + 3x(-2C + Dx)) + d^3(A + 3Bx - x^2(3C + Dx)))}{3d^4(c+dx)^{3/2}}
$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/(c + d*x)^(5/2),x]
[Out] (-2*(16*c^3*D - 8*c^2*d*(C - 3*D*x) + 2*c*d^2*(B + 3*x*(-2*C + D*x)) + d^3*
(A + 3*B*x - x^2*(3*C + D*x))))/(3*d^4*(c + d*x)^(3/2))
```

Maple [A] (verified)

Time = 1.64 (sec) , antiderivative size = 72, normalized size of antiderivative = 0.64

method	result	size
pseudoelliptic	$-\frac{2\left(\left(-D x^{3}-3 C\,x^{2}+3 B x +A \right) d^{3}+2 c \left(3 D \,x^{2}-6 C x +B \right) d^{2}-8 c^{2} \left(-3 D x +C \right) d +16 D \,c^{3}\right)}{3 \left(d x +c \right)^{\frac{3}{2}} d^{4}}$	72
gosper	$-\frac{2 \left(-D \,x^{3} d^{3}-3 C \,d^{3} x^{2}+6 D c \,d^{2} x^{2}+3 B \,d^{3} x -12 C c \,d^{2} x +24 D \,c^{2} d x +A \,d^{3}+2 B c \,d^{2}-8 C \,c^{2} d +16 D \,c^{3}\right)}{3 \left(d x +c \right)^{\frac{3}{2}} d^{4}}$	90
trager	$-\frac{2 \left(-D \,x^{3} d^{3}-3 C \,d^{3} x^{2}+6 D c \,d^{2} x^{2}+3 B \,d^{3} x -12 C c \,d^{2} x +24 D \,c^{2} d x +A \,d^{3}+2 B c \,d^{2}-8 C \,c^{2} d +16 D \,c^{3}\right)}{3 \left(d x +c \right)^{\frac{3}{2}} d^{4}}$	90
derivativedivides	$\frac{\frac{2 D \left(d x +c \right)^{\frac{3}{2}}}{3}+2 d C \sqrt{d x +c}-6 D c \sqrt{d x +c}-\frac{2 \left(B \,d^{2}-2 C c d +3 D \,c^{2}\right)}{\sqrt{d x +c}}-\frac{2 \left(A \,d^{3}-B c \,d^{2}+C \,c^{2} d -D \,c^{3}\right)}{3 \left(d x +c \right)^{\frac{3}{2}}}}{d^{4}}$	98
default	$\frac{\frac{2 D \left(d x +c \right)^{\frac{3}{2}}}{3}+2 d C \sqrt{d x +c}-6 D c \sqrt{d x +c}-\frac{2 \left(B \,d^{2}-2 C c d +3 D \,c^{2}\right)}{\sqrt{d x +c}}-\frac{2 \left(A \,d^{3}-B c \,d^{2}+C \,c^{2} d -D \,c^{3}\right)}{3 \left(d x +c \right)^{\frac{3}{2}}}}{d^{4}}$	98

```
[In] int((D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x,method=_RETURNVERBOSE)
```

```
[Out] -2/3*((-D*x^3-3*C*x^2+3*B*x+A)*d^3+2*c*(3*D*x^2-6*C*x+B)*d^2-8*c^2*(-3*D*x+
C)*d+16*D*c^3)/(d*x+c)^(3/2)/d^4
```

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 110, normalized size of antiderivative = 0.97

$$\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{5/2}}\,dx = \frac{2\,(Dd^3x^3 - 16\,Dc^3 + 8\,Cc^2d - 2\,Bcd^2 - Ad^3 - 3\,(2\,Dcd^2 - Cd^3)x^2 - 3\,(8\,D}{3\,(d^6x^2 + 2\,cd^5x + c^2d^4)}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x, algorithm="fricas")
```

```
[Out] 2/3*(D*d^3*x^3 - 16*D*c^3 + 8*C*c^2*d - 2*B*c*d^2 - A*d^3 - 3*(2*D*c*d^2 -
C*d^3)*x^2 - 3*(8*D*c^2*d - 4*C*c*d^2 + B*d^3)*x)*sqrt(d*x + c)/(d^6*x^2 +
2*c*d^5*x + c^2*d^4)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 425 vs. 2(119) = 238.

Time = 0.36 (sec) , antiderivative size = 425, normalized size of antiderivative = 3.76

$$\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{5/2}}\,dx = \begin{cases} -\frac{2Ad^3}{3cd^4\sqrt{c+dx}+3d^5x\sqrt{c+dx}} - \frac{4Bcd^2}{3cd^4\sqrt{c+dx}+3d^5x\sqrt{c+dx}} - \frac{6Bd^3x}{3cd^4\sqrt{c+dx}+3d^5x\sqrt{c+dx}} + \\ \frac{Ax+\frac{Bx^2}{2}+\frac{Cx^3}{3}+\frac{Dx^4}{4}}{c^{\frac{5}{2}}} \end{cases}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(d*x+c)**(5/2),x)
```

```
[Out] Piecewise((-2*A*d**3/(3*c*d**4*sqrt(c + d*x) + 3*d**5*x*sqrt(c + d*x)) - 4*
B*c*d**2/(3*c*d**4*sqrt(c + d*x) + 3*d**5*x*sqrt(c + d*x)) - 6*B*d**3*x/(3*
c*d**4*sqrt(c + d*x) + 3*d**5*x*sqrt(c + d*x)) + 16*C*c**2*d/(3*c*d**4*sqrt
(c + d*x) + 3*d**5*x*sqrt(c + d*x)) + 24*C*c*d**2*x/(3*c*d**4*sqrt(c + d*x)
 + 3*d**5*x*sqrt(c + d*x)) + 6*C*d**3*x**2/(3*c*d**4*sqrt(c + d*x) + 3*d**5
*x*sqrt(c + d*x)) - 32*D*c**3/(3*c*d**4*sqrt(c + d*x) + 3*d**5*x*sqrt(c + d
*x)) - 48*D*c**2*d*x/(3*c*d**4*sqrt(c + d*x) + 3*d**5*x*sqrt(c + d*x)) - 12
*D*c*d**2*x**2/(3*c*d**4*sqrt(c + d*x) + 3*d**5*x*sqrt(c + d*x)) + 2*D*d**3
*x**3/(3*c*d**4*sqrt(c + d*x) + 3*d**5*x*sqrt(c + d*x)), Ne(d, 0)), ((A*x +
 B*x**2/2 + C*x**3/3 + D*x**4/4)/c**(5/2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 98, normalized size of antiderivative = 0.87

$$\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{5/2}}\,dx = \frac{2\left(\frac{(dx+c)^{\frac{3}{2}}D-3\,(3\,Dc-Cd)\sqrt{dx+c}}{d^3}+\frac{Dc^3-Cc^2d+Bcd^2-Ad^3-3\,(3\,Dc^2-2\,Ccd+Bd^2)(dx+c)}{(dx+c)^{\frac{3}{2}}d^3}\right)}{3\,d}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x, algorithm="maxima")
```

```
[Out] 2/3*(((d*x + c)^(3/2)*D - 3*(3*D*c - C*d)*sqrt(d*x + c))/d^3 + (D*c^3 - C*c
^2*d + B*c*d^2 - A*d^3 - 3*(3*D*c^2 - 2*C*c*d + B*d^2)*(d*x + c))/((d*x + c
)^(3/2)*d^3))/d
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 115, normalized size of antiderivative = 1.02

$$\begin{aligned}\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{5/2}}\,dx =& \\ &-\frac{2\,(9\,(dx+c)Dc^2-Dc^3-6\,(dx+c)Ccd+Cc^2d+3\,(dx+c)Bd^2-Bcd^2+Ad^3)}{3\,(dx+c)^{\frac{3}{2}}d^4} \\ &+\frac{2\left((dx+c)^{\frac{3}{2}}Dd^8-9\,\sqrt{dx+c}Dcd^8+3\,\sqrt{dx+c}Cd^9\right)}{3\,d^{12}}\end{aligned}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(d*x+c)^(5/2),x, algorithm="giac")
```

```
[Out] -2/3*(9*(d*x + c)*D*c^2 - D*c^3 - 6*(d*x + c)*C*c*d + C*c^2*d + 3*(d*x + c)
*B*d^2 - B*c*d^2 + A*d^3)/((d*x + c)^(3/2)*d^4) + 2/3*((d*x + c)^(3/2)*D*d^
8 - 9*sqrt(d*x + c)*D*c*d^8 + 3*sqrt(d*x + c)*C*d^9)/d^12
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(c+dx)^{5/2}}\,dx=\int \frac{A+B\,x+C\,x^2+x^3\,D}{{\left(c+d\,x\right)}^{5/2}} \,d x$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/(c + d*x)^(5/2),x)
[Out] int((A + B*x + C*x^2 + x^3*D)/(c + d*x)^(5/2), x)
```

3.22 $\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{5/2}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 210

$$\begin{aligned}\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{5/2}}\,dx &= \frac{2(c^2Cd-Bcd^2+Ad^3-c^3D)}{3d^3(bc-ad)(c+dx)^{3/2}} \\ &+ \frac{2(ad(2cCd-Bd^2-3c^2D)-b(c^2Cd-Ad^3-2c^3D))}{d^3(bc-ad)^2\sqrt{c+dx}} \\ &+ \frac{2D\sqrt{c+dx}}{bd^3} - \frac{2(Ab^3-a(b^2B-abC+a^2D))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{3/2}(bc-ad)^{5/2}}\end{aligned}$$

```
[Out] 2/3*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)/d^3/(-a*d+b*c)/(d*x+c)^(3/2)-2*(A*b^3-a*(
B*b^2-C*a*b+D*a^2))*arctanh(b^(1/2)*(d*x+c)^(1/2)/(-a*d+b*c)^(1/2))/b^(3/2)
/(-a*d+b*c)^(5/2)+2*(a*d*(-B*d^2+2*C*c*d-3*D*c^2)-b*(-A*d^3+C*c^2*d-2*D*c^3
))/d^3/(-a*d+b*c)^2/(d*x+c)^(1/2)+2*D*(d*x+c)^(1/2)/b/d^3
```

Rubi [A] (verified)

Time = 0.21 (sec) , antiderivative size = 210, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.094, Rules used = {1633, 65, 214}

$$\begin{aligned}\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{5/2}}\,dx &= -\frac{2(Ab^3-a(a^2D-abC+b^2B))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{3/2}(bc-ad)^{5/2}} \\ &+ \frac{2(ad(-Bd^2-3c^2D+2cCd)-b(-Ad^3-2c^3D+c^2Cd))}{d^3\sqrt{c+dx}(bc-ad)^2} \\ &+ \frac{2(Ad^3-Bcd^2+c^3(-D)+c^2Cd)}{3d^3(c+dx)^{3/2}(bc-ad)} + \frac{2D\sqrt{c+dx}}{bd^3}\end{aligned}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/((a + b*x)*(c + d*x)^(5/2)),x]
```

```
[Out] (2*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D))/(3*d^3*(b*c - a*d)*(c + d*x)^(3/2))
 + (2*(a*d*(2*c*C*d - B*d^2 - 3*c^2*D) - b*(c^2*C*d - A*d^3 - 2*c^3*D)))/(d
^3*(b*c - a*d)^2*Sqrt[c + d*x]) + (2*D*Sqrt[c + d*x])/(b*d^3) - (2*(A*b^3 -
 a*(b^2*B - a*b*C + a^2*D))*ArcTanh[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[b*c - a*d]
])/(b^(3/2)*(b*c - a*d)^(5/2))
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 1633

```
Int[((Px_)*((c_.) + (d_.)*(x_))^(n_.))/((a_.) + (b_.)*(x_)), x_Symbol] :> I
nt[ExpandIntegrand[1/Sqrt[c + d*x], Px*((c + d*x)^(n + 1/2)/(a + b*x)), x],
 x] /; FreeQ[{a, b, c, d, n}, x] && PolyQ[Px, x] && ILtQ[n + 1/2, 0] && GtQ
[Expon[Px, x], 2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \left(\frac{c^2Cd - Bcd^2 + Ad^3 - c^3D}{d^2(-bc+ad)(c+dx)^{5/2}} + \frac{-ad(2cCd - Bd^2 - 3c^2D) + b(c^2Cd - Ad^3 - 2c^3D)}{d^2(bc-ad)^2(c+dx)^{3/2}} + \frac{D}{bd^2\sqrt{c+dx}} + \frac{Ab^3 - a(b^2B - abC + a^2D)}{b(bc-ad)^2(a+bx)\sqrt{c+dx}}\right) dx \\
&= \frac{2(c^2Cd - Bcd^2 + Ad^3 - c^3D)}{3d^3(bc-ad)(c+dx)^{3/2}} + \frac{2(ad(2cCd - Bd^2 - 3c^2D) - b(c^2Cd - Ad^3 - 2c^3D))}{d^3(bc-ad)^2\sqrt{c+dx}} + \frac{2D\sqrt{c+dx}}{bd^3} + \frac{(Ab^3 - a(b^2B - abC + a^2D))\int \frac{1}{(a+bx)\sqrt{c+dx}}\,dx}{b(bc-ad)^2}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{2(c^2Cd - Bcd^2 + Ad^3 - c^3D)}{3d^3(bc - ad)(c + dx)^{3/2}} + \frac{2(ad(2cCd - Bd^2 - 3c^2D) - b(c^2Cd - Ad^3 - 2c^3D))}{d^3(bc - ad)^2\sqrt{c + dx}} \\
&\quad + \frac{2D\sqrt{c + dx}}{bd^3} + \frac{(2(Ab^3 - a(b^2B - abC + a^2D)))\,\text{Subst}\left(\int \frac{1}{a - \frac{bc}{d} + \frac{bx^2}{d}}\,dx, x, \sqrt{c + dx}\right)}{bd(bc - ad)^2} \\
&= \frac{2(c^2Cd - Bcd^2 + Ad^3 - c^3D)}{3d^3(bc - ad)(c + dx)^{3/2}} \\
&\quad + \frac{2(ad(2cCd - Bd^2 - 3c^2D) - b(c^2Cd - Ad^3 - 2c^3D))}{d^3(bc - ad)^2\sqrt{c + dx}} \\
&\quad + \frac{2D\sqrt{c + dx}}{bd^3} - \frac{2(Ab^3 - a(b^2B - abC + a^2D))\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{3/2}(bc - ad)^{5/2}}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.47 (sec) , antiderivative size = 230, normalized size of antiderivative = 1.10

$$
\begin{aligned}
&\int \frac{A + Bx + Cx^2 + Dx^3}{(a + bx)(c + dx)^{5/2}}\,dx = \frac{6a^2d^2D(c + dx)^2 - 2abd(14c^3D + d^3(A + 3Bx) + c^2(-5Cd + 21dDx) + 2cd^2}{3b} \\
&+ \frac{2(Ab^3 - a(b^2B - abC + a^2D))\arctan\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{-bc+ad}}\right)}{b^{3/2}(-bc + ad)^{5/2}}
\end{aligned}
$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/((a + b*x)*(c + d*x)^(5/2)),x]
```

```
[Out] (6*a^2*d^2*D*(c + d*x)^2 - 2*a*b*d*(14*c^3*D + d^3*(A + 3*B*x) + c^2*(-5*C*
d + 21*d*D*x) + 2*c*d^2*(B - 3*C*x + 3*D*x^2)) + 2*b^2*(4*A*c*d^3 + 8*c^4*D
 + 3*A*d^4*x - 2*c^3*d*(C - 6*D*x) - c^2*d^2*(B + 3*x*(C - D*x))))/(3*b*d^3
*(b*c - a*d)^2*(c + d*x)^(3/2)) + (2*(A*b^3 - a*(b^2*B - a*b*C + a^2*D))*Ar
cTan[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[-(b*c) + a*d]])/(b^(3/2)*(-(b*c) + a*d)^(
5/2))
```

Maple [A] (verified)

Time = 1.82 (sec) , antiderivative size = 202, normalized size of antiderivative = 0.96

method	result
derivativedivides	$\frac{\frac{2D\sqrt{dx+c}}{b}+\frac{2\left(b^{3}A-a\,b^{2}B+C\,a^{2}b-Da^{3}\right)d^{3}\arctan\left(\frac{b\sqrt{dx+c}}{\sqrt{(ad-bc)b}}\right)}{(ad-bc)^{2}b\sqrt{(ad-bc)b}}-\frac{2\left(A\,d^{3}-Bc\,d^{2}+C\,c^{2}d-Dc^{3}\right)}{3(ad-bc)(dx+c)^{\frac{3}{2}}}-\frac{2\left(-Ab\,d^{3}+Ba\,d^{3}-2Cac\,d^{2}+C\right.}{(ad-bc)^{2}}}{d^{3}}$
default	$\frac{\frac{2D\sqrt{dx+c}}{b}+\frac{2\left(b^{3}A-a\,b^{2}B+C\,a^{2}b-Da^{3}\right)d^{3}\arctan\left(\frac{b\sqrt{dx+c}}{\sqrt{(ad-bc)b}}\right)}{(ad-bc)^{2}b\sqrt{(ad-bc)b}}-\frac{2\left(A\,d^{3}-Bc\,d^{2}+C\,c^{2}d-Dc^{3}\right)}{3(ad-bc)(dx+c)^{\frac{3}{2}}}-\frac{2\left(-Ab\,d^{3}+Ba\,d^{3}-2Cac\,d^{2}+C\right.}{(ad-bc)^{2}}}{d^{3}}$
pseudoelliptic	$\frac{\frac{2D\sqrt{dx+c}}{b}+\frac{2\left(b^{3}A-a\,b^{2}B+C\,a^{2}b-Da^{3}\right)d^{3}\arctan\left(\frac{b\sqrt{dx+c}}{\sqrt{(ad-bc)b}}\right)}{(ad-bc)^{2}b\sqrt{(ad-bc)b}}-\frac{2\left(A\,d^{3}-Bc\,d^{2}+C\,c^{2}d-Dc^{3}\right)}{3(ad-bc)(dx+c)^{\frac{3}{2}}}+\frac{2\left(Ab\,d^{3}-Ba\,d^{3}+2Cac\,d^{2}-Cb\,c\right.}{(ad-bc)^{2}}}{d^{3}}$

```
[In] int((D*x^3+C*x^2+B*x+A)/(b*x+a)/(d*x+c)^(5/2),x,method=_RETURNVERBOSE)
```

```
[Out] 2/d^3*(D/b*(d*x+c)^(1/2)-1/3*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)/(a*d-b*c)/(d*x+c
)^(3/2)-1/(a*d-b*c)^2*(-A*b*d^3+B*a*d^3-2*C*a*c*d^2+C*b*c^2*d+3*D*a*c^2*d-2
*D*b*c^3)/(d*x+c)^(1/2)+1/(a*d-b*c)^2*(A*b^3-B*a*b^2+C*a^2*b-D*a^3)/b*d^3/(
(a*d-b*c)*b)^(1/2)*arctan(b*(d*x+c)^(1/2)/((a*d-b*c)*b)^(1/2)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 637 vs. $2(191) = 382$.

Time = 0.30 (sec) , antiderivative size = 1287, normalized size of antiderivative = 6.13

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{5/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)/(d*x+c)^(5/2),x, algorithm="fricas")
```

```
[Out] [1/3*(3*((D*a^3 - C*a^2*b + B*a*b^2 - A*b^3)*d^5*x^2 + 2*(D*a^3*c - (C*a^2*
b - B*a*b^2 + A*b^3)*c)*d^4*x + (D*a^3*c^2 - (C*a^2*b - B*a*b^2 + A*b^3)*c^
2)*d^3)*sqrt(b^2*c - a*b*d)*log((b*d*x + 2*b*c - a*d + 2*sqrt(b^2*c - a*b*d
)*sqrt(d*x + c))/(b*x + a)) + 2*(8*D*b^4*c^5 + A*a^2*b^2*d^5 + (2*B*a^2*b^2
 - 5*A*a*b^3)*c*d^4 - (3*D*a^3*b*c^2 + (5*C*a^2*b^2 + B*a*b^3 - 4*A*b^4)*c^
2)*d^3 + (17*D*a^2*b^2*c^3 + (7*C*a*b^3 - B*b^4)*c^3)*d^2 + 3*(D*b^4*c^3*d^
2 - 3*D*a*b^3*c^2*d^3 + 3*D*a^2*b^2*c*d^4 - D*a^3*b*d^5)*x^2 - 2*(11*D*a*b^
3*c^4 + C*b^4*c^4)*d + 3*(4*D*b^4*c^4*d + (B*a^2*b^2 - A*a*b^3)*d^5 - (2*D*
a^3*b*c + (2*C*a^2*b^2 + B*a*b^3 - A*b^4)*c)*d^4 + 3*(3*D*a^2*b^2*c^2 + C*a
*b^3*c^2)*d^3 - (11*D*a*b^3*c^3 + C*b^4*c^3)*d^2)*x)*sqrt(d*x + c))/(b^5*c^
5*d^3 - 3*a*b^4*c^4*d^4 + 3*a^2*b^3*c^3*d^5 - a^3*b^2*c^2*d^6 + (b^5*c^3*d^
5 - 3*a*b^4*c^2*d^6 + 3*a^2*b^3*c*d^7 - a^3*b^2*d^8)*x^2 + 2*(b^5*c^4*d^4 -
 3*a*b^4*c^3*d^5 + 3*a^2*b^3*c^2*d^6 - a^3*b^2*c*d^7)*x), -2/3*(3*((D*a^3 -
 C*a^2*b + B*a*b^2 - A*b^3)*d^5*x^2 + 2*(D*a^3*c - (C*a^2*b - B*a*b^2 + A*b
^3)*c)*d^4*x + (D*a^3*c^2 - (C*a^2*b - B*a*b^2 + A*b^3)*c^2)*d^3)*sqrt(-b^2
*c + a*b*d)*arctan(sqrt(-b^2*c + a*b*d)*sqrt(d*x + c)/(b*d*x + b*c)) - (8*D
*b^4*c^5 + A*a^2*b^2*d^5 + (2*B*a^2*b^2 - 5*A*a*b^3)*c*d^4 - (3*D*a^3*b*c^2
```

```
 + (5*C*a^2*b^2 + B*a*b^3 - 4*A*b^4)*c^2)*d^3 + (17*D*a^2*b^2*c^3 + (7*C*a*
b^3 - B*b^4)*c^3)*d^2 + 3*(D*b^4*c^3*d^2 - 3*D*a*b^3*c^2*d^3 + 3*D*a^2*b^2*
c*d^4 - D*a^3*b*d^5)*x^2 - 2*(11*D*a*b^3*c^4 + C*b^4*c^4)*d + 3*(4*D*b^4*c^
4*d + (B*a^2*b^2 - A*a*b^3)*d^5 - (2*D*a^3*b*c + (2*C*a^2*b^2 + B*a*b^3 - A
*b^4)*c)*d^4 + 3*(3*D*a^2*b^2*c^2 + C*a*b^3*c^2)*d^3 - (11*D*a*b^3*c^3 + C*
b^4*c^3)*d^2)*x)*sqrt(d*x + c))/(b^5*c^5*d^3 - 3*a*b^4*c^4*d^4 + 3*a^2*b^3*
c^3*d^5 - a^3*b^2*c^2*d^6 + (b^5*c^3*d^5 - 3*a*b^4*c^2*d^6 + 3*a^2*b^3*c*d^
7 - a^3*b^2*d^8)*x^2 + 2*(b^5*c^4*d^4 - 3*a*b^4*c^3*d^5 + 3*a^2*b^3*c^2*d^6
 - a^3*b^2*c*d^7)*x)]
```

Sympy [A] (verification not implemented)

Time = 10.43 (sec) , antiderivative size = 306, normalized size of antiderivative = 1.46

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{5/2}}\,dx = \begin{cases} \frac{2\left(\frac{D\sqrt{c+dx}}{bd^{2}} - \frac{-Abd^{3}+Bad^{3}-2Cacd^{2}+Cbc^{2}d+3Dac^{2}d-2Dbc^{3}}{d^{2}\sqrt{c+dx}(ad-bc)^{2}} + \frac{-Ad^{3}+Bcd^{2}-Cc^{2}d+Dc^{3}}{3d^{2}(c+dx)^{\frac{3}{2}}(ad-bc)} - \frac{d\left(-Ab^{3}+Bab^{2}-C\right.}{b^{2}\sqrt{\quad}}\right.}{d} \\ \frac{\frac{Dx^{3}}{3b} + \frac{x^{2}(Cb-Da)}{2b^{2}} + \frac{x\left(Bb^{2}-Cab+Da^{2}\right)}{b^{3}} - \frac{\left(-Ab^{3}+Bab^{2}-Ca^{2}b+Da^{3}\right)\left(\begin{cases} \frac{x}{a} & \text{for}\: b = 0 \\ \frac{\log(a+bx)}{b} & \text{otherwise} \end{cases}\right)}{b^{3}}}{c^{\frac{5}{2}}} \end{cases}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(b*x+a)/(d*x+c)**(5/2),x)
```

```
[Out] Piecewise((2*(D*sqrt(c + d*x)/(b*d**2) - (-A*b*d**3 + B*a*d**3 - 2*C*a*c*d*
*2 + C*b*c**2*d + 3*D*a*c**2*d - 2*D*b*c**3)/(d**2*sqrt(c + d*x)*(a*d - b*c
)**2) + (-A*d**3 + B*c*d**2 - C*c**2*d + D*c**3)/(3*d**2*(c + d*x)**(3/2)*(
a*d - b*c)) - d*(-A*b**3 + B*a*b**2 - C*a**2*b + D*a**3)*atan(sqrt(c + d*x)
/sqrt((a*d - b*c)/b))/(b**2*sqrt((a*d - b*c)/b)*(a*d - b*c)**2))/d, Ne(d, 0
)), ((D*x**3/(3*b) + x**2*(C*b - D*a)/(2*b**2) + x*(B*b**2 - C*a*b + D*a**2
)/b**3 - (-A*b**3 + B*a*b**2 - C*a**2*b + D*a**3)*Piecewise((x/a, Eq(b, 0))
, (log(a + b*x)/b, True))/b**3)/c**(5/2), True))
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{5/2}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)/(d*x+c)^(5/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 281, normalized size of antiderivative = 1.34

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{5/2}}\,dx = -\frac{2\,(Da^3-Ca^2b+Bab^2-Ab^3)\arctan\left(\frac{\sqrt{dx+c}b}{\sqrt{-b^2c+abd}}\right)}{(b^3c^2-2\,ab^2cd+a^2bd^2)\sqrt{-b^2c+abd}}$$

$$+\frac{2\,(6\,(dx+c)Dbc^3-Dbc^4-9\,(dx+c)Dac^2d-3\,(dx+c)Cbc^2d+Dac^3d+Cbc^3d+6\,(dx+c)Cacd^2-C}{3\,(b^2c^2d^3-2\,abcd^4+a^2d^5)(dx+}$$

$$+\frac{2\sqrt{dx+c}D}{bd^3}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)/(d*x+c)^(5/2),x, algorithm="giac")
```

```
[Out] -2*(D*a^3 - C*a^2*b + B*a*b^2 - A*b^3)*arctan(sqrt(d*x + c)*b/sqrt(-b^2*c +
 a*b*d))/((b^3*c^2 - 2*a*b^2*c*d + a^2*b*d^2)*sqrt(-b^2*c + a*b*d)) + 2/3*(
6*(d*x + c)*D*b*c^3 - D*b*c^4 - 9*(d*x + c)*D*a*c^2*d - 3*(d*x + c)*C*b*c^2
*d + D*a*c^3*d + C*b*c^3*d + 6*(d*x + c)*C*a*c*d^2 - C*a*c^2*d^2 - B*b*c^2*
d^2 - 3*(d*x + c)*B*a*d^3 + 3*(d*x + c)*A*b*d^3 + B*a*c*d^3 + A*b*c*d^3 - A
*a*d^4)/((b^2*c^2*d^3 - 2*a*b*c*d^4 + a^2*d^5)*(d*x + c)^(3/2)) + 2*sqrt(d*
x + c)*D/(b*d^3)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)(c+dx)^{5/2}}\,dx = \int \frac{A+B\,x+C\,x^2+x^3\,D}{(a+b\,x)\,\left(c+d\,x\right)^{5/2}}\,dx$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)*(c + d*x)^(5/2)),x)
```

```
[Out] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)*(c + d*x)^(5/2)), x)
```

3.23 $\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{5/2}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 336

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{5/2}}\,dx=\frac{3ab^2Bd^3-3a^2bCd^3+3a^3d^3D-b^3(2c^2Cd-2Bcd^2+5Ad^3-2c^3D)}{3b^3d^2(bc-ad)^2(c+dx)^{3/2}}$$
$$-\frac{A-\frac{a(b^2B-abC+a^2D)}{b^3}}{(bc-ad)(a+bx)(c+dx)^{3/2}}$$
$$-\frac{a^2bCd^3-a^3d^3D+ab^2d(4cCd-3Bd^2-6c^2D)-b^3(2Bcd^2-5Ad^3-2c^3D)}{b^2d^2(bc-ad)^3\sqrt{c+dx}}$$
$$-\frac{(b^3(2Bc-5Ad)-ab^2(4cC-3Bd)-a^3dD-a^2b(Cd-6cD))\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{3/2}(bc-ad)^{7/2}}$$

```
[Out] 1/3*(3*a*b^2*B*d^3-3*a^2*b*C*d^3+3*a^3*d^3*D-b^3*(5*A*d^3-2*B*c*d^2+2*C*c^2
*d-2*D*c^3))/b^3/d^2/(-a*d+b*c)^2/(d*x+c)^(3/2)+(-A+a*(B*b^2-C*a*b+D*a^2)/b
^3)/(-a*d+b*c)/(b*x+a)/(d*x+c)^(3/2)-(b^3*(-5*A*d+2*B*c)-a*b^2*(-3*B*d+4*C*
c)-a^3*d*D-a^2*b*(C*d-6*D*c))*arctanh(b^(1/2)*(d*x+c)^(1/2)/(-a*d+b*c)^(1/2
))/b^(3/2)/(-a*d+b*c)^(7/2)+(-a^2*b*C*d^3+a^3*d^3*D-a*b^2*d*(-3*B*d^2+4*C*c
*d-6*D*c^2)+b^3*(-5*A*d^3+2*B*c*d^2-2*D*c^3))/b^2/d^2/(-a*d+b*c)^3/(d*x+c)^
(1/2)
```

Rubi [A] (verified)

Time = 0.53 (sec) , antiderivative size = 336, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {1635, 911, 1275, 214}

$$\begin{aligned}\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{5/2}}\,dx = & -\frac{A-\frac{a(a^2D-abC+b^2B)}{b^3}}{(a+bx)(c+dx)^{3/2}(bc-ad)} \\ & -\frac{\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)(a^3(-d)D-a^2b(Cd-6cD)-ab^2(4cC-3Bd)+b^3(2Bc-5Ad))}{b^{3/2}(bc-ad)^{7/2}} \\ & -\frac{-a^3d^3D+a^2bCd^3+ab^2d(-3Bd^2-6c^2D+4cCd)-(b^3(-5Ad^3+2Bcd^2-2c^3D))}{b^2d^2\sqrt{c+dx}(bc-ad)^3} \\ & +\frac{3a^3d^3D-3a^2bCd^3+3ab^2Bd^3-(b^3(5Ad^3-2Bcd^2-2c^3D+2c^2Cd))}{3b^3d^2(c+dx)^{3/2}(bc-ad)^2}\end{aligned}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^2*(c + d*x)^(5/2)),x]
```

```
[Out] (3*a*b^2*B*d^3 - 3*a^2*b*C*d^3 + 3*a^3*d^3*D - b^3*(2*c^2*C*d - 2*B*c*d^2 +
 5*A*d^3 - 2*c^3*D))/(3*b^3*d^2*(b*c - a*d)^2*(c + d*x)^(3/2)) - (A - (a*(b
^2*B - a*b*C + a^2*D))/b^3)/((b*c - a*d)*(a + b*x)*(c + d*x)^(3/2)) - (a^2*
b*C*d^3 - a^3*d^3*D + a*b^2*d*(4*c*C*d - 3*B*d^2 - 6*c^2*D) - b^3*(2*B*c*d^
2 - 5*A*d^3 - 2*c^3*D))/(b^2*d^2*(b*c - a*d)^3*Sqrt[c + d*x]) - ((b^3*(2*B*
c - 5*A*d) - a*b^2*(4*c*C - 3*B*d) - a^3*d*D - a^2*b*(C*d - 6*c*D))*ArcTanh
[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[b*c - a*d]])/(b^(3/2)*(b*c - a*d)^(7/2))
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 911

```
Int[((d_.) + (e_.)*(x_))^(m_)*((f_.) + (g_.)*(x_))^(n_)*((a_.) + (b_.)*(x_)
 + (c_.)*(x_)^2)^(p_.), x_Symbol] :> With[{q = Denominator[m]}, Dist[q/e, S
ubst[Int[x^(q*(m + 1) - 1)*((e*f - d*g)/e + g*(x^q/e))^n*((c*d^2 - b*d*e +
a*e^2)/e^2 - (2*c*d - b*e)*(x^q/e^2) + c*(x^(2*q)/e^2))^p, x], x, (d + e*x)
^(1/q)], x]] /; FreeQ[{a, b, c, d, e, f, g}, x] && NeQ[e*f - d*g, 0] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IntegersQ[n, p] && Fra
ctionQ[m]
```

Rule 1275

```
Int[((f_.)*(x_))^(m_.)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (
c_.)*(x_)^4)^(p_.), x_Symbol] :> Int[ExpandIntegrand[(f*x)^m*(d + e*x^2)^q*
(a + b*x^2 + c*x^4)^p, x], x] /; FreeQ[{a, b, c, d, e, f, m, q}, x] && NeQ[
```

```
b^2 - 4*a*c, 0] && IGtQ[p, 0] && IGtQ[q, -2]
```

Rule 1635

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol] :
> With[{Qx = PolynomialQuotient[Px, a + b*x, x], R = PolynomialRemainder[Px
, a + b*x, x]}, Simp[R*(a + b*x)^(m + 1)*((c + d*x)^(n + 1)/((m + 1)*(b*c -
 a*d))), x] + Dist[1/((m + 1)*(b*c - a*d)), Int[(a + b*x)^(m + 1)*(c + d*x)
^n*ExpandToSum[(m + 1)*(b*c - a*d)*Qx - d*R*(m + n + 2), x], x], x]] /; Fre
eQ[{a, b, c, d, n}, x] && PolyQ[Px, x] && ILtQ[m, -1] && GtQ[Expon[Px, x],
2]
```

Rubi steps

$$\text{integral} = -\frac{A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}}{(bc-ad)(a+bx)(c+dx)^{3/2}} + \frac{\int \frac{-\frac{b^3(2Bc-5Ad)-ab^2(2cC-3Bd)+3a^3dD-a^2b(3Cd-2cD)}{2b^3}-\frac{(bc-ad)(bC-aD)x}{b^2}-\left(c-\frac{ad}{b}\right)Dx^2}{(a+bx)(c+dx)^{5/2}}\,dx}{-bc+ad}$$

$$= -\frac{A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}}{(bc-ad)(a+bx)(c+dx)^{3/2}} - \frac{2\text{Subst}\left(\int \frac{\frac{-c^2\left(c-\frac{ad}{b}\right)D+\frac{cd(bc-ad)(bC-aD)}{b^2}-\frac{d^2\left(b^3(2Bc-5Ad)-ab^2(2cC-3Bd)+3a^3dD-a^2b(3Cd-2cD)\right)}{2b^3}}{d^2} - \frac{\left(-2c\left(c-\frac{ad}{b}\right)D+\frac{d(bc-ad}{}\right.}{d^2}}{x^4\left(\frac{-bc+ad}{d}+\frac{bx^2}{d}\right)}\right.}{d(bc-ad)}$$

$$= -\frac{A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}}{(bc-ad)(a+bx)(c+dx)^{3/2}} - \frac{2\text{Subst}\left(\int\left(\frac{3ab^2Bd^3-3a^2bCd^3+3a^3d^3D-b^3\left(2c^2Cd-2Bcd^2+5Ad^3-2c^3D\right)}{2b^3d(bc-ad)x^4}+\frac{-a^2bCd^3+a^3d^3D-ab^2d\left(4cCd-3Bd^2-6c^2D\right)-}{2b^2d(bc-ad)^2x^2}\right.\right.}{d(bc-ad)}$$

$$= \frac{3ab^2Bd^3-3a^2bCd^3+3a^3d^3D-b^3(2c^2Cd-2Bcd^2+5Ad^3-2c^3D)}{3b^3d^2(bc-ad)^2(c+dx)^{3/2}} - \frac{A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}}{(bc-ad)(a+bx)(c+dx)^{3/2}} - \frac{a^2bCd^3-a^3d^3D+ab^2d(4cCd-3Bd^2-6c^2D)-b^3(2Bcd^2-5Ad^3-2c^3D)}{b^2d^2(bc-ad)^3\sqrt{c+dx}} - \frac{(b^3(2Bc-5Ad)-ab^2(4cC-3Bd)-a^3dD-a^2b(Cd-6cD))\,\text{Subst}\left(\int\frac{1}{bc-ad-bx^2}\,dx,x,\sqrt{c+dx}\right)}{b(bc-ad)^3}$$

$$= \frac{3ab^2Bd^3 - 3a^2bCd^3 + 3a^3d^3D - b^3(2c^2Cd - 2Bcd^2 + 5Ad^3 - 2c^3D)}{3b^3d^2(bc-ad)^2(c+dx)^{3/2}}$$

$$- \frac{A - \frac{a(b^2B-abC+a^2D)}{b^3}}{(bc-ad)(a+bx)(c+dx)^{3/2}}$$

$$- \frac{a^2bCd^3 - a^3d^3D + ab^2d(4cCd - 3Bd^2 - 6c^2D) - b^3(2Bcd^2 - 5Ad^3 - 2c^3D)}{b^2d^2(bc-ad)^3\sqrt{c+dx}}$$

$$- \frac{(b^3(2Bc-5Ad) - ab^2(4cC-3Bd) - a^3dD - a^2b(Cd-6cD))\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)}{b^{3/2}(bc-ad)^{7/2}}$$

Mathematica [A] (verified)

Time = 0.90 (sec) , antiderivative size = 354, normalized size of antiderivative = 1.05

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{5/2}}\,dx = \frac{-3a^3d^2D(c+dx)^2 - a^2bd(16c^3D + 2cd^2(2B-9Cx) + c^2(-13Cd+18dDx}{}$$

$$- \frac{(b^3(2Bc-5Ad) + ab^2(-4cC+3Bd) - a^3dD + a^2b(-Cd+6cD))\arctan\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{-bc+ad}}\right)}{b^{3/2}(-bc+ad)^{7/2}}$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^2*(c + d*x)^(5/2)),x]
```

```
[Out] (-3*a^3*d^2*D*(c + d*x)^2 - a^2*b*d*(16*c^3*D + 2*c*d^2*(2*B - 9*C*x) + c^2
*(-13*C*d + 18*d*D*x) + d^3*(2*A + 6*B*x - 3*C*x^2)) + a*b^2*(4*c^4*D + d^4
*x*(10*A - 9*B*x) + 2*c^3*d*(C - 5*D*x) + 2*c*d^3*(7*A - 8*B*x + 6*C*x^2) +
 c^2*d^2*(-11*B + 2*x*(5*C - 9*D*x))) + b^3*(A*d^2*(3*c^2 + 20*c*d*x + 15*d
^2*x^2) + 2*c*x*(-4*B*c*d^2 + 2*c^3*D - 3*B*d^3*x + c^2*d*(C + 3*D*x))))/(3
*b*d^2*(-(b*c) + a*d)^3*(a + b*x)*(c + d*x)^(3/2)) - ((b^3*(2*B*c - 5*A*d)
+ a*b^2*(-4*c*C + 3*B*d) - a^3*d*D + a^2*b*(-(C*d) + 6*c*D))*ArcTan[(Sqrt[b
]*Sqrt[c + d*x])/Sqrt[-(b*c) + a*d]])/(b^(3/2)*(-(b*c) + a*d)^(7/2))
```

Maple [A] (verified)

Time = 2.00 (sec) , antiderivative size = 275, normalized size of antiderivative = 0.82

method	result
derivativedivides	$\frac{\frac{2d^{2}\left(\frac{d\left(b^{3}A-a\,b^{2}B+C\,a^{2}b-Da^{3}\right)\sqrt{dx+c}}{2b((dx+c)b+ad-bc)}+\frac{\left(5A\,b^{3}d-3Ba\,b^{2}d-2B\,b^{3}c+C\,a^{2}bd+4Ca\,b^{2}c+a^{3}dD-6Da^{2}bc\right)\arctan\left(\frac{b\sqrt{dx+c}}{\sqrt{(ad-bc)b}}\right)}{2b\sqrt{(ad-bc)b}}\right)}{(ad-bc)^{3}}-}{d^{2}}$
default	$\frac{\frac{2d^{2}\left(\frac{d\left(b^{3}A-a\,b^{2}B+C\,a^{2}b-Da^{3}\right)\sqrt{dx+c}}{2b((dx+c)b+ad-bc)}+\frac{\left(5A\,b^{3}d-3Ba\,b^{2}d-2B\,b^{3}c+C\,a^{2}bd+4Ca\,b^{2}c+a^{3}dD-6Da^{2}bc\right)\arctan\left(\frac{b\sqrt{dx+c}}{\sqrt{(ad-bc)b}}\right)}{2b\sqrt{(ad-bc)b}}\right)}{(ad-bc)^{3}}-}{d^{2}}$
pseudoelliptic	$\frac{5(dx+c)^{\frac{3}{2}}\left(\left(b^{3}A-\frac{3}{5}a\,b^{2}B+\frac{1}{5}C\,a^{2}b+\frac{1}{5}Da^{3}\right)d-\frac{2bc\left(B\,b^{2}-2Cab+3Da^{2}\right)}{5}\right)(bx+a)d^{2}\arctan\left(\frac{b\sqrt{dx+c}}{\sqrt{(ad-bc)b}}\right)-\frac{2\sqrt{(ad-bc)b}\left(\left(-\right.\right.}{}}{}$

```
[In] int((D*x^3+C*x^2+B*x+A)/(b*x+a)^2/(d*x+c)^(5/2),x,method=_RETURNVERBOSE)
```

```
[Out] 2/d^2*(d^2/(a*d-b*c)^3*(1/2*d*(A*b^3-B*a*b^2+C*a^2*b-D*a^3)/b*(d*x+c)^(1/2)
/((d*x+c)*b+a*d-b*c)+1/2*(5*A*b^3*d-3*B*a*b^2*d-2*B*b^3*c+C*a^2*b*d+4*C*a*b
^2*c+D*a^3*d-6*D*a^2*b*c)/b/((a*d-b*c)*b)^(1/2)*arctan(b*(d*x+c)^(1/2)/((a*
d-b*c)*b)^(1/2)))-1/3*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)/(a*d-b*c)^2/(d*x+c)^(3/
2)-1/(a*d-b*c)^3*(-2*A*b*d^3+B*a*d^3+B*b*c*d^2-2*C*a*c*d^2+3*D*a*c^2*d-D*b*
c^3)/(d*x+c)^(1/2))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1215 vs. 2(315) = 630.

Time = 0.37 (sec) , antiderivative size = 2444, normalized size of antiderivative = 7.27

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{5/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^2/(d*x+c)^(5/2),x, algorithm="fricas"
)
```

```
[Out] [-1/6*(3*((D*a^4*c^2 + (C*a^3*b - 3*B*a^2*b^2 + 5*A*a*b^3)*c^2)*d^3 + ((D*a
^3*b + C*a^2*b^2 - 3*B*a*b^3 + 5*A*b^4)*d^5 - 2*(3*D*a^2*b^2*c - (2*C*a*b^3
 - B*b^4)*c)*d^4)*x^3 - 2*(3*D*a^3*b*c^3 - (2*C*a^2*b^2 - B*a*b^3)*c^3)*d^2
 + ((D*a^4 + C*a^3*b - 3*B*a^2*b^2 + 5*A*a*b^3)*d^5 - 2*(2*D*a^3*b*c - (3*C
*a^2*b^2 - 4*B*a*b^3 + 5*A*b^4)*c)*d^4 - 4*(3*D*a^2*b^2*c^2 - (2*C*a*b^3 -
B*b^4)*c^2)*d^3)*x^2 + (2*(D*a^4*c + (C*a^3*b - 3*B*a^2*b^2 + 5*A*a*b^3)*c)
```

```
*d^4 - (11*D*a^3*b*c^2 - (9*C*a^2*b^2 - 7*B*a*b^3 + 5*A*b^4)*c^2)*d^3 - 2*(
3*D*a^2*b^2*c^3 - (2*C*a*b^3 - B*b^4)*c^3)*d^2)*x)*sqrt(b^2*c - a*b*d)*log(
(b*d*x + 2*b*c - a*d - 2*sqrt(b^2*c - a*b*d)*sqrt(d*x + c))/(b*x + a)) + 2*
(4*D*a*b^4*c^5 + 2*A*a^3*b^2*d^5 + 4*(B*a^3*b^2 - 4*A*a^2*b^3)*c*d^4 + (3*D
*a^4*b*c^2 - (13*C*a^3*b^2 - 7*B*a^2*b^3 - 11*A*a*b^4)*c^2)*d^3 + (13*D*a^3
*b^2*c^3 + (11*C*a^2*b^3 - 11*B*a*b^4 + 3*A*b^5)*c^3)*d^2 + 3*(2*D*b^5*c^4*
d - 8*D*a*b^4*c^3*d^2 + (D*a^4*b - C*a^3*b^2 + 3*B*a^2*b^3 - 5*A*a*b^4)*d^5
 - (D*a^3*b^2*c + (3*C*a^2*b^3 + B*a*b^4 - 5*A*b^5)*c)*d^4 + 2*(3*D*a^2*b^3
*c^2 + (2*C*a*b^4 - B*b^5)*c^2)*d^3)*x^2 - 2*(10*D*a^2*b^3*c^4 - C*a*b^4*c^
4)*d + 2*(2*D*b^5*c^5 + (3*B*a^3*b^2 - 5*A*a^2*b^3)*d^5 + (3*D*a^4*b*c - (9
*C*a^3*b^2 - 5*B*a^2*b^3 + 5*A*a*b^4)*c)*d^4 + 2*(3*D*a^3*b^2*c^2 + (2*C*a^
2*b^3 - 2*B*a*b^4 + 5*A*b^5)*c^2)*d^3 - 4*(D*a^2*b^3*c^3 - (C*a*b^4 - B*b^5
)*c^3)*d^2 - (7*D*a*b^4*c^4 - C*b^5*c^4)*d)*x)*sqrt(d*x + c))/(a*b^6*c^6*d^
2 - 4*a^2*b^5*c^5*d^3 + 6*a^3*b^4*c^4*d^4 - 4*a^4*b^3*c^3*d^5 + a^5*b^2*c^2
*d^6 + (b^7*c^4*d^4 - 4*a*b^6*c^3*d^5 + 6*a^2*b^5*c^2*d^6 - 4*a^3*b^4*c*d^7
 + a^4*b^3*d^8)*x^3 + (2*b^7*c^5*d^3 - 7*a*b^6*c^4*d^4 + 8*a^2*b^5*c^3*d^5
- 2*a^3*b^4*c^2*d^6 - 2*a^4*b^3*c*d^7 + a^5*b^2*d^8)*x^2 + (b^7*c^6*d^2 - 2
*a*b^6*c^5*d^3 - 2*a^2*b^5*c^4*d^4 + 8*a^3*b^4*c^3*d^5 - 7*a^4*b^3*c^2*d^6
+ 2*a^5*b^2*c*d^7)*x), -1/3*(3*((D*a^4*c^2 + (C*a^3*b - 3*B*a^2*b^2 + 5*A*a
*b^3)*c^2)*d^3 + ((D*a^3*b + C*a^2*b^2 - 3*B*a*b^3 + 5*A*b^4)*d^5 - 2*(3*D*
a^2*b^2*c - (2*C*a*b^3 - B*b^4)*c)*d^4)*x^3 - 2*(3*D*a^3*b*c^3 - (2*C*a^2*b
^2 - B*a*b^3)*c^3)*d^2 + ((D*a^4 + C*a^3*b - 3*B*a^2*b^2 + 5*A*a*b^3)*d^5 -
 2*(2*D*a^3*b*c - (3*C*a^2*b^2 - 4*B*a*b^3 + 5*A*b^4)*c)*d^4 - 4*(3*D*a^2*b
^2*c^2 - (2*C*a*b^3 - B*b^4)*c^2)*d^3)*x^2 + (2*(D*a^4*c + (C*a^3*b - 3*B*a
^2*b^2 + 5*A*a*b^3)*c)*d^4 - (11*D*a^3*b*c^2 - (9*C*a^2*b^2 - 7*B*a*b^3 + 5
*A*b^4)*c^2)*d^3 - 2*(3*D*a^2*b^2*c^3 - (2*C*a*b^3 - B*b^4)*c^3)*d^2)*x)*sq
rt(-b^2*c + a*b*d)*arctan(sqrt(-b^2*c + a*b*d)*sqrt(d*x + c)/(b*d*x + b*c))
 + (4*D*a*b^4*c^5 + 2*A*a^3*b^2*d^5 + 4*(B*a^3*b^2 - 4*A*a^2*b^3)*c*d^4 + (
3*D*a^4*b*c^2 - (13*C*a^3*b^2 - 7*B*a^2*b^3 - 11*A*a*b^4)*c^2)*d^3 + (13*D*
a^3*b^2*c^3 + (11*C*a^2*b^3 - 11*B*a*b^4 + 3*A*b^5)*c^3)*d^2 + 3*(2*D*b^5*c
^4*d - 8*D*a*b^4*c^3*d^2 + (D*a^4*b - C*a^3*b^2 + 3*B*a^2*b^3 - 5*A*a*b^4)*
d^5 - (D*a^3*b^2*c + (3*C*a^2*b^3 + B*a*b^4 - 5*A*b^5)*c)*d^4 + 2*(3*D*a^2*
b^3*c^2 + (2*C*a*b^4 - B*b^5)*c^2)*d^3)*x^2 - 2*(10*D*a^2*b^3*c^4 - C*a*b^4
*c^4)*d + 2*(2*D*b^5*c^5 + (3*B*a^3*b^2 - 5*A*a^2*b^3)*d^5 + (3*D*a^4*b*c -
 (9*C*a^3*b^2 - 5*B*a^2*b^3 + 5*A*a*b^4)*c)*d^4 + 2*(3*D*a^3*b^2*c^2 + (2*C
*a^2*b^3 - 2*B*a*b^4 + 5*A*b^5)*c^2)*d^3 - 4*(D*a^2*b^3*c^3 - (C*a*b^4 - B*
b^5)*c^3)*d^2 - (7*D*a*b^4*c^4 - C*b^5*c^4)*d)*x)*sqrt(d*x + c))/(a*b^6*c^6
*d^2 - 4*a^2*b^5*c^5*d^3 + 6*a^3*b^4*c^4*d^4 - 4*a^4*b^3*c^3*d^5 + a^5*b^2*
c^2*d^6 + (b^7*c^4*d^4 - 4*a*b^6*c^3*d^5 + 6*a^2*b^5*c^2*d^6 - 4*a^3*b^4*c*
d^7 + a^4*b^3*d^8)*x^3 + (2*b^7*c^5*d^3 - 7*a*b^6*c^4*d^4 + 8*a^2*b^5*c^3*d
^5 - 2*a^3*b^4*c^2*d^6 - 2*a^4*b^3*c*d^7 + a^5*b^2*d^8)*x^2 + (b^7*c^6*d^2
- 2*a*b^6*c^5*d^3 - 2*a^2*b^5*c^4*d^4 + 8*a^3*b^4*c^3*d^5 - 7*a^4*b^3*c^2*d
^6 + 2*a^5*b^2*c*d^7)*x)]
```

Sympy [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(b*x+a)**2/(d*x+c)**(5/2),x)
```

```
[Out] Timed out
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{5/2}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^2/(d*x+c)^(5/2),x, algorithm="maxima"
)
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 439, normalized size of antiderivative = 1.31

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{5/2}}\,dx = \frac{(6\,Da^2bc - 4\,Cab^2c + 2\,Bb^3c - Da^3d - Ca^2bd + 3\,Bab^2d - 5\,Ab^3d)\arctan\Big(}{(b^4c^3 - 3\,ab^3c^2d + 3\,a^2b^2cd^2 - a^3bd^3)\sqrt{-b^2c+abd}}$$
$$+ \frac{\sqrt{dx+c}Da^3d - \sqrt{dx+c}Ca^2bd + \sqrt{dx+c}Bab^2d - \sqrt{dx+c}Ab^3d}{(b^4c^3 - 3\,ab^3c^2d + 3\,a^2b^2cd^2 - a^3bd^3)((dx+c)b - bc + ad)}$$
$$- \frac{2\,(3\,(dx+c)Dbc^3 - Dbc^4 - 9\,(dx+c)Dac^2d + Dac^3d + Cbc^3d + 6\,(dx+c)Cacd^2 - 3\,(dx+c)Bbcd^2 - C}{3\,(b^3c^3d^2 - 3\,ab^2c^2d^3 + 3\,a^2bcd^4 - a^3d^5}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^2/(d*x+c)^(5/2),x, algorithm="giac")
```

```
[Out] (6*D*a^2*b*c - 4*C*a*b^2*c + 2*B*b^3*c - D*a^3*d - C*a^2*b*d + 3*B*a*b^2*d
- 5*A*b^3*d)*arctan(sqrt(d*x + c)*b/sqrt(-b^2*c + a*b*d))/((b^4*c^3 - 3*a*b
^3*c^2*d + 3*a^2*b^2*c*d^2 - a^3*b*d^3)*sqrt(-b^2*c + a*b*d)) + (sqrt(d*x +
 c)*D*a^3*d - sqrt(d*x + c)*C*a^2*b*d + sqrt(d*x + c)*B*a*b^2*d - sqrt(d*x
```

```
+ c)*A*b^3*d)/((b^4*c^3 - 3*a*b^3*c^2*d + 3*a^2*b^2*c*d^2 - a^3*b*d^3)*((d*
x + c)*b - b*c + a*d)) - 2/3*(3*(d*x + c)*D*b*c^3 - D*b*c^4 - 9*(d*x + c)*D
*a*c^2*d + D*a*c^3*d + C*b*c^3*d + 6*(d*x + c)*C*a*c*d^2 - 3*(d*x + c)*B*b*
c*d^2 - C*a*c^2*d^2 - B*b*c^2*d^2 - 3*(d*x + c)*B*a*d^3 + 6*(d*x + c)*A*b*d
^3 + B*a*c*d^3 + A*b*c*d^3 - A*a*d^4)/((b^3*c^3*d^2 - 3*a*b^2*c^2*d^3 + 3*a
^2*b*c*d^4 - a^3*d^5)*(d*x + c)^(3/2))
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^2(c+dx)^{5/2}}\,dx=\int \frac{A+B\,x+C\,x^2+x^3\,D}{{\left(a+b\,x\right)}^2\,{\left(c+d\,x\right)}^{5/2}}\,dx$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^2*(c + d*x)^(5/2)),x)
[Out] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^2*(c + d*x)^(5/2)), x)
```

3.24 $\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{5/2}} \, dx$

Optimal result

Integrand size = 32, antiderivative size = 438

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{5/2}} \, dx = -\frac{3ab^2Bd^3-3a^2bCd^3+3a^3d^3D-b^3(4c^2Cd-4Bcd^2+7Ad^3-4c^3D)}{6b^3d(bc-ad)^3(c+dx)^{3/2}} - \frac{Ab^3-a(b^2B-abC+a^2D)}{2b^3(bc-ad)(a+bx)^2(c+dx)^{3/2}} + \frac{a^2bCd^2+b^3(2c^2C-4Bcd+7Ad^2)-a^3d^2D+ab^2(4cCd-3Bd^2-6c^2D)}{b^2(bc-ad)^4\sqrt{c+dx}} - \frac{(b^3(4Bc-7Ad)-ab^2(8cC-3Bd)-5a^3dD+a^2b(Cd+12cD))\sqrt{c+dx}}{4b(bc-ad)^4(a+bx)} - \frac{(b^3(8c^2C-20Bcd+35Ad^2)+a^3d^2D+3a^2bd(Cd-4cD)+3ab^2(8cCd-5Bd^2-8c^2D))\,\mathrm{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d}}{\sqrt{bc-ad}}\right.}{4b^{3/2}(bc-ad)^{9/2}}$$

```
[Out] 1/6*(-3*a*b^2*B*d^3+3*a^2*b*C*d^3-3*a^3*d^3*D+b^3*(7*A*d^3-4*B*c*d^2+4*C*c^
2*d-4*D*c^3))/b^3/d/(-a*d+b*c)^3/(d*x+c)^(3/2)+1/2*(-A*b^3+a*(B*b^2-C*a*b+D
*a^2))/b^3/(-a*d+b*c)/(b*x+a)^2/(d*x+c)^(3/2)-1/4*(b^3*(35*A*d^2-20*B*c*d+8
*C*c^2)+a^3*d^2*D+3*a^2*b*d*(C*d-4*D*c)+3*a*b^2*(-5*B*d^2+8*C*c*d-8*D*c^2))
*arctanh(b^(1/2)*(d*x+c)^(1/2)/(-a*d+b*c)^(1/2))/b^(3/2)/(-a*d+b*c)^(9/2)+(
a^2*b*C*d^2+b^3*(7*A*d^2-4*B*c*d+2*C*c^2)-a^3*d^2*D+a*b^2*(-3*B*d^2+4*C*c*d
-6*D*c^2))/b^2/(-a*d+b*c)^4/(d*x+c)^(1/2)-1/4*(b^3*(-7*A*d+4*B*c)-a*b^2*(-3
*B*d+8*C*c)-5*a^3*d*D+a^2*b*(C*d+12*D*c))*(d*x+c)^(1/2)/b/(-a*d+b*c)^4/(b*x
+a)
```

Rubi [A] (verified)

Time = 0.82 (sec) , antiderivative size = 438, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}} = 0.156$, Rules used = {1635, 911, 1273, 1275, 214}

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{5/2}}\,dx = -\frac{Ab^3-a(a^2D-abC+b^2B)}{2b^3(a+bx)^2(c+dx)^{3/2}(bc-ad)}$$
$$-\frac{\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+dx}}{\sqrt{bc-ad}}\right)(a^3d^2D+3a^2bd(Cd-4cD)+3ab^2(-5Bd^2-8c^2D+8cCd)+b^3(35Ad^2-20Bcd+8}{4b^{3/2}(bc-ad)^{9/2}}$$
$$+\frac{a^3(-d^2)D+a^2bCd^2+ab^2(-3Bd^2-6c^2D+4cCd)+b^3(7Ad^2-4Bcd+2c^2C)}{b^2\sqrt{c+dx}(bc-ad)^4}$$
$$-\frac{3a^3d^3D-3a^2bCd^3+3ab^2Bd^3-(b^3(7Ad^3-4Bcd^2-4c^3D+4c^2Cd))}{6b^3d(c+dx)^{3/2}(bc-ad)^3}$$
$$-\frac{\sqrt{c+dx}(-5a^3dD+a^2b(12cD+Cd)-ab^2(8cC-3Bd)+b^3(4Bc-7Ad))}{4b(a+bx)(bc-ad)^4}$$

```
[In] Int[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^3*(c + d*x)^(5/2)),x]
```

```
[Out] -1/6*(3*a*b^2*B*d^3 - 3*a^2*b*C*d^3 + 3*a^3*d^3*D - b^3*(4*c^2*C*d - 4*B*c*
d^2 + 7*A*d^3 - 4*c^3*D))/(b^3*d*(b*c - a*d)^3*(c + d*x)^(3/2)) - (A*b^3 -
a*(b^2*B - a*b*C + a^2*D))/(2*b^3*(b*c - a*d)*(a + b*x)^2*(c + d*x)^(3/2))
+ (a^2*b*C*d^2 + b^3*(2*c^2*C - 4*B*c*d + 7*A*d^2) - a^3*d^2*D + a*b^2*(4*c
*C*d - 3*B*d^2 - 6*c^2*D))/(b^2*(b*c - a*d)^4*Sqrt[c + d*x]) - ((b^3*(4*B*c
 - 7*A*d) - a*b^2*(8*c*C - 3*B*d) - 5*a^3*d*D + a^2*b*(C*d + 12*c*D))*Sqrt[
c + d*x])/(4*b*(b*c - a*d)^4*(a + b*x)) - ((b^3*(8*c^2*C - 20*B*c*d + 35*A*
d^2) + a^3*d^2*D + 3*a^2*b*d*(C*d - 4*c*D) + 3*a*b^2*(8*c*C*d - 5*B*d^2 - 8
*c^2*D))*ArcTanh[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[b*c - a*d]])/(4*b^(3/2)*(b*c
- a*d)^(9/2))
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 911

```
Int[((d_.) + (e_.)*(x_))^(m_)*((f_.) + (g_.)*(x_))^(n_)*((a_.) + (b_.)*(x_)
 + (c_.)*(x_)^2)^(p_.), x_Symbol] :> With[{q = Denominator[m]}, Dist[q/e, S
ubst[Int[x^(q*(m + 1) - 1)*((e*f - d*g)/e + g*(x^q/e))^n*((c*d^2 - b*d*e +
a*e^2)/e^2 - (2*c*d - b*e)*(x^q/e^2) + c*(x^(2*q)/e^2))^p, x], x, (d + e*x)
^(1/q)], x]] /; FreeQ[{a, b, c, d, e, f, g}, x] && NeQ[e*f - d*g, 0] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IntegersQ[n, p] && Fra
ctionQ[m]
```

Rule 1273

```
Int[(x_)^(m_)*((d_) + (e_.)*(x_)^2)^(q_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^
4)^(p_.), x_Symbol] :> Simp[(-d)^(m/2 - 1)*(c*d^2 - b*d*e + a*e^2)^p*x*((d
+ e*x^2)^(q + 1)/(2*e^(2*p + m/2)*(q + 1))), x] + Dist[(-d)^(m/2 - 1)/(2*e^
(2*p)*(q + 1)), Int[x^m*(d + e*x^2)^(q + 1)*ExpandToSum[Together[(1/(d + e*
x^2))*(2*(-d)^(-m/2 + 1)*e^(2*p)*(q + 1)*(a + b*x^2 + c*x^4)^p - ((c*d^2 -
b*d*e + a*e^2)^p/(e^(m/2)*x^m))*(d + e*(2*q + 3)*x^2))], x], x], x] /; Free
Q[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && IGtQ[p, 0] && ILtQ[q, -1] &
& ILtQ[m/2, 0]
```

Rule 1275

```
Int[((f_.)*(x_))^(m_.)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (
c_.)*(x_)^4)^(p_.), x_Symbol] :> Int[ExpandIntegrand[(f*x)^m*(d + e*x^2)^q*
(a + b*x^2 + c*x^4)^p, x], x] /; FreeQ[{a, b, c, d, e, f, m, q}, x] && NeQ[
b^2 - 4*a*c, 0] && IGtQ[p, 0] && IGtQ[q, -2]
```

Rule 1635

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol] :
> With[{Qx = PolynomialQuotient[Px, a + b*x, x], R = PolynomialRemainder[Px
, a + b*x, x]}, Simp[R*(a + b*x)^(m + 1)*((c + d*x)^(n + 1)/((m + 1)*(b*c -
 a*d))), x] + Dist[1/((m + 1)*(b*c - a*d)), Int[(a + b*x)^(m + 1)*(c + d*x)
^n*ExpandToSum[(m + 1)*(b*c - a*d)*Qx - d*R*(m + n + 2), x], x], x]] /; Fre
eQ[{a, b, c, d, n}, x] && PolyQ[Px, x] && ILtQ[m, -1] && GtQ[Expon[Px, x],
2]
```

Rubi steps

$$\text{integral} = -\frac{Ab^3-a(b^2B-abC+a^2D)}{2b^3(bc-ad)(a+bx)^2(c+dx)^{3/2}} - \frac{\int \frac{-\frac{b^3(4Bc-7Ad)-ab^2(4cC-3Bd)+3a^3dD-a^2b(3Cd-4cD)}{2b^3}-\frac{2(bc-ad)(bC-aD)x}{b^2}-2\left(c-\frac{ad}{b}\right)Dx^2}{(a+bx)^2(c+dx)^{5/2}}\,dx}{2(bc-ad)}$$

$$= -\frac{Ab^3-a(b^2B-abC+a^2D)}{2b^3(bc-ad)(a+bx)^2(c+dx)^{3/2}} - \frac{\text{Subst}\left(\int \frac{\frac{-2c^2\left(c-\frac{ad}{b}\right)D+\frac{2cd(bc-ad)(bC-aD)}{b^2}-\frac{d^2\left(b^3(4Bc-7Ad)-ab^2(4cC-3Bd)+3a^3dD-a^2b(3Cd-4cD)\right)}{2b^3}}{d^2}-\frac{\left(-4c\left(c-\frac{ad}{b}\right)D+\frac{2d(bc-a}{}\right.}{d^2}}{x^4\left(\frac{-bc+ad}{d}+\frac{bx^2}{d}\right)^2}\right.}{d(bc-ad)}$$

$$= -\frac{Ab^3 - a(b^2B - abC + a^2D)}{2b^3(bc-ad)(a+bx)^2(c+dx)^{3/2}}$$
$$-\frac{\left(b^3(4Bc-7Ad) - ab^2(8cC-3Bd) - 5a^3dD + a^2b(Cd+12cD)\right)\sqrt{c+dx}}{4b(bc-ad)^4(a+bx)}$$
$$+\frac{d^4\text{Subst}\left(\int \frac{-\frac{(bc-ad)^2\left(3ab^2Bd^3-3a^2bCd^3+3a^3d^3D-b^3\left(4c^2Cd-4Bcd^2+7Ad^3-4c^3D\right)\right)}{bd^6} - \frac{(bc-ad)\left(a^2bCd^3-a^3d^3D-ab^2d\left(8cCd-3Bd^2\right.\right.}{d^6}}{x^4\left(\frac{-bc+ad}{d}+\frac{bx^2}{d}\right)}\right.}{2b^2(bc-a}$$

$$= -\frac{Ab^3 - a(b^2B - abC + a^2D)}{2b^3(bc-ad)(a+bx)^2(c+dx)^{3/2}}$$
$$-\frac{\left(b^3(4Bc-7Ad) - ab^2(8cC-3Bd) - 5a^3dD + a^2b(Cd+12cD)\right)\sqrt{c+dx}}{4b(bc-ad)^4(a+bx)}$$
$$+\frac{d^4\text{Subst}\left(\int \left(\frac{(bc-ad)\left(3ab^2Bd^3-3a^2bCd^3+3a^3d^3D-b^3\left(4c^2Cd-4Bcd^2+7Ad^3-4c^3D\right)\right)}{bd^5x^4} + \frac{2\left(-a^2bCd^2-b^3\left(2c^2C-4Bcd+7A\right.\right.}{}\right.\right.}{}$$

$$= -\frac{3ab^2Bd^3 - 3a^2bCd^3 + 3a^3d^3D - b^3(4c^2Cd - 4Bcd^2 + 7Ad^3 - 4c^3D)}{6b^3d(bc-ad)^3(c+dx)^{3/2}}$$
$$-\frac{Ab^3 - a(b^2B - abC + a^2D)}{2b^3(bc-ad)(a+bx)^2(c+dx)^{3/2}}$$
$$+\frac{a^2bCd^2 + b^3(2c^2C - 4Bcd + 7Ad^2) - a^3d^2D + ab^2(4cCd - 3Bd^2 - 6c^2D)}{b^2(bc-ad)^4\sqrt{c+dx}}$$
$$-\frac{\left(b^3(4Bc-7Ad) - ab^2(8cC-3Bd) - 5a^3dD + a^2b(Cd+12cD)\right)\sqrt{c+dx}}{4b(bc-ad)^4(a+bx)}$$
$$-\frac{\left(b^3(8c^2C - 20Bcd + 35Ad^2) + a^3d^2D + 3a^2bd(Cd-4cD) + 3ab^2(8cCd - 5Bd^2 - 8c^2D)\right)\text{Subs}}{4b(bc-ad)^4}$$

$$= -\frac{3ab^2Bd^3 - 3a^2bCd^3 + 3a^3d^3D - b^3(4c^2Cd - 4Bcd^2 + 7Ad^3 - 4c^3D)}{6b^3d(bc-ad)^3(c+dx)^{3/2}}$$
$$-\frac{Ab^3 - a(b^2B - abC + a^2D)}{2b^3(bc-ad)(a+bx)^2(c+dx)^{3/2}}$$
$$+\frac{a^2bCd^2 + b^3(2c^2C - 4Bcd + 7Ad^2) - a^3d^2D + ab^2(4cCd - 3Bd^2 - 6c^2D)}{b^2(bc-ad)^4\sqrt{c+dx}}$$
$$-\frac{\left(b^3(4Bc-7Ad) - ab^2(8cC-3Bd) - 5a^3dD + a^2b(Cd+12cD)\right)\sqrt{c+dx}}{4b(bc-ad)^4(a+bx)}$$
$$-\frac{\left(b^3(8c^2C - 20Bcd + 35Ad^2) + a^3d^2D + 3a^2bd(Cd-4cD) + 3ab^2(8cCd - 5Bd^2 - 8c^2D)\right)\text{tanh}}{4b^{3/2}(bc-ad)^{9/2}}$$

Mathematica [A] (verified)

Time = 1.54 (sec) , antiderivative size = 554, normalized size of antiderivative = 1.26

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{5/2}}\,dx = \frac{-3a^4d^2D(c+dx)^2-4b^4cx(2cx(-4cCd+c^2D-3Cd^2x)+Bd(3c^2+20cdx+}{}$$

$$+\frac{(b^3(8c^2C-20Bcd+35Ad^2)+a^3d^2D+3a^2bd(Cd-4cD)-3ab^2(-8cCd+5Bd^2+8c^2D))\arctan\left(\frac{\sqrt{b}\sqrt{c+d}}{\sqrt{-bc+a}}\right.}{4b^{3/2}(-bc+ad)^{9/2}}$$

```
[In] Integrate[(A + B*x + C*x^2 + D*x^3)/((a + b*x)^3*(c + d*x)^(5/2)),x]
```

```
[Out] (-3*a^4*d^2*D*(c + d*x)^2 - 4*b^4*c*x*(2*c*x*(-4*c*C*d + c^2*D - 3*C*d^2*x)
 + B*d*(3*c^2 + 20*c*d*x + 15*d^2*x^2)) + a^3*b*d*(-94*c^3*D + c^2*d*(55*C
- 129*D*x) + 3*d^3*x*(-8*B + 5*C*x + D*x^2) - 2*c*d^2*(8*B - 39*C*x + 12*D*
x^2)) - a^2*b^2*(8*c^4*D + 3*d^4*x^2*(25*B - 3*C*x) + c^3*(-50*C*d + 164*d*
D*x) + 2*c*d^3*x*(67*B - 66*C*x + 18*D*x^2) + c^2*d^2*(83*B - 149*C*x + 216
*D*x^2)) + A*b*d*(-8*a^3*d^3 + 8*a^2*b*d^2*(10*c + 7*d*x) + a*b^2*d*(39*c^2
 + 238*c*d*x + 175*d^2*x^2) + b^3*(-6*c^3 + 21*c^2*d*x + 140*c*d^2*x^2 + 10
5*d^3*x^3)) - a*b^3*(B*d*(6*c^3 + 145*c^2*d*x + 160*c*d^2*x^2 + 45*d^3*x^3)
 + 8*c*x*(2*c^3*D - 9*C*d^3*x^2 + c*d^2*x*(-17*C + 9*D*x) + c^2*(-11*C*d +
8*d*D*x))))/(12*b*d*(b*c - a*d)^4*(a + b*x)^2*(c + d*x)^(3/2)) + ((b^3*(8*c
^2*C - 20*B*c*d + 35*A*d^2) + a^3*d^2*D + 3*a^2*b*d*(C*d - 4*c*D) - 3*a*b^2
*(-8*c*C*d + 5*B*d^2 + 8*c^2*D))*ArcTan[(Sqrt[b]*Sqrt[c + d*x])/Sqrt[-(b*c)
 + a*d]])/(4*b^(3/2)*(-(b*c) + a*d)^(9/2))
```

Maple [A] (verified)

Time = 1.96 (sec) , antiderivative size = 459, normalized size of antiderivative = 1.05

method	result
derivativedivides	$2d\left(\frac{\left(\frac{11}{8}A\,b^3d^2-\frac{7}{8}Ba\,b^2d^2-\frac{1}{2}B\,b^3cd+\frac{3}{8}a^2bC\,d^2+Ca\,b^2cd+\frac{1}{8}a^3d^2D-\frac{3}{2}Da^2bcd\right)(dx+c)^{\frac{3}{2}}+\frac{d\left(13Aa\,b^3d^2-13A\,b^4cd-9B\,a^2b^2d^2+5B\right.}{((dx+c)b+ad-bc)^2}}{}\right.$
default	$2d\left(\frac{\left(\frac{11}{8}A\,b^3d^2-\frac{7}{8}Ba\,b^2d^2-\frac{1}{2}B\,b^3cd+\frac{3}{8}a^2bC\,d^2+Ca\,b^2cd+\frac{1}{8}a^3d^2D-\frac{3}{2}Da^2bcd\right)(dx+c)^{\frac{3}{2}}+\frac{d\left(13Aa\,b^3d^2-13A\,b^4cd-9B\,a^2b^2d^2+5B\right.}{((dx+c)b+ad-bc)^2}}{}\right.$
pseudoelliptic	$\frac{35(dx+c)^{\frac{3}{2}}\left(\left(b^3A-\frac{3}{7}a\,b^2B+\frac{3}{35}C\,a^2b+\frac{1}{35}Da^3\right)d^2-\frac{4\left(B\,b^2-\frac{6}{5}Cab+\frac{3}{5}Da^2\right)bcd}{7}+\frac{8b^2c^2(Cb-3Da)}{35}\right)(bx+a)^2d\arctan\left(\frac{b\sqrt{dx+c}}{\sqrt{(ad-bc)b}}\right)}{4}-$

```
[In] int((D*x^3+C*x^2+B*x+A)/(b*x+a)^3/(d*x+c)^(5/2),x,method=_RETURNVERBOSE)
```

```
[Out] 2/d*(-1/3*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)/(a*d-b*c)^3/(d*x+c)^(3/2)+d*(3*A*b*
d^2-B*a*d^2-2*B*b*c*d+2*C*a*c*d+C*b*c^2-3*D*a*c^2)/(a*d-b*c)^4/(d*x+c)^(1/2
)+d/(a*d-b*c)^4*(((11/8*A*b^3*d^2-7/8*B*a*b^2*d^2-1/2*B*b^3*c*d+3/8*a^2*b*C
*d^2+C*a*b^2*c*d+1/8*a^3*d^2*D-3/2*D*a^2*b*c*d)*(d*x+c)^(3/2)+1/8*d*(13*A*a
*b^3*d^2-13*A*b^4*c*d-9*B*a^2*b^2*d^2+5*B*a*b^3*c*d+4*B*b^4*c^2+5*C*a^3*b*d
^2+3*C*a^2*b^2*c*d-8*C*a*b^3*c^2-D*a^4*d^2-11*D*a^3*b*c*d+12*D*a^2*b^2*c^2)
/b*(d*x+c)^(1/2))/((d*x+c)*b+a*d-b*c)^2+1/8*(35*A*b^3*d^2-15*B*a*b^2*d^2-20
*B*b^3*c*d+3*C*a^2*b*d^2+24*C*a*b^2*c*d+8*C*b^3*c^2+D*a^3*d^2-12*D*a^2*b*c*
d-24*D*a*b^2*c^2)/b/((a*d-b*c)*b)^(1/2)*arctan(b*(d*x+c)^(1/2)/((a*d-b*c)*b
)^(1/2))))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1938 vs. $2(415) = 830$.

Time = 0.51 (sec) , antiderivative size = 3889, normalized size of antiderivative = 8.88

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{5/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^3/(d*x+c)^(5/2),x, algorithm="fricas"
)
```

```
[Out] [1/24*(3*(((D*a^3*b^2 + 3*C*a^2*b^3 - 15*B*a*b^4 + 35*A*b^5)*d^5 - 4*(3*D*a
^2*b^3*c - (6*C*a*b^4 - 5*B*b^5)*c)*d^4 - 8*(3*D*a*b^4*c^2 - C*b^5*c^2)*d^3
)*x^4 + (D*a^5*c^2 + (3*C*a^4*b - 15*B*a^3*b^2 + 35*A*a^2*b^3)*c^2)*d^3 + 2
*((D*a^4*b + 3*C*a^3*b^2 - 15*B*a^2*b^3 + 35*A*a*b^4)*d^5 - (11*D*a^3*b^2*c
 - (27*C*a^2*b^3 - 35*B*a*b^4 + 35*A*b^5)*c)*d^4 - 4*(9*D*a^2*b^3*c^2 - (8*
C*a*b^4 - 5*B*b^5)*c^2)*d^3 - 8*(3*D*a*b^4*c^3 - C*b^5*c^3)*d^2)*x^3 - 4*(3
*D*a^4*b*c^3 - (6*C*a^3*b^2 - 5*B*a^2*b^3)*c^3)*d^2 + ((D*a^5 + 3*C*a^4*b -
 15*B*a^3*b^2 + 35*A*a^2*b^3)*d^5 - 4*(2*D*a^4*b*c - (9*C*a^3*b^2 - 20*B*a^
2*b^3 + 35*A*a*b^4)*c)*d^4 - (71*D*a^3*b^2*c^2 - (107*C*a^2*b^3 - 95*B*a*b^
4 + 35*A*b^5)*c^2)*d^3 - 4*(27*D*a^2*b^3*c^3 - (14*C*a*b^4 - 5*B*b^5)*c^3)*
d^2 - 8*(3*D*a*b^4*c^4 - C*b^5*c^4)*d)*x^2 - 8*(3*D*a^3*b^2*c^4 - C*a^2*b^3
*c^4)*d + 2*((D*a^5*c + (3*C*a^4*b - 15*B*a^3*b^2 + 35*A*a^2*b^3)*c)*d^4 -
(11*D*a^4*b*c^2 - (27*C*a^3*b^2 - 35*B*a^2*b^3 + 35*A*a*b^4)*c^2)*d^3 - 4*(
9*D*a^3*b^2*c^3 - (8*C*a^2*b^3 - 5*B*a*b^4)*c^3)*d^2 - 8*(3*D*a^2*b^3*c^4 -
 C*a*b^4*c^4)*d)*x)*sqrt(b^2*c - a*b*d)*log((b*d*x + 2*b*c - a*d - 2*sqrt(b
^2*c - a*b*d)*sqrt(d*x + c))/(b*x + a)) - 2*(8*D*a^2*b^4*c^5 - 8*A*a^4*b^2*
d^5 - 8*(2*B*a^4*b^2 - 11*A*a^3*b^3)*c*d^4 - (3*D*a^5*b*c^2 - (55*C*a^4*b^2
 - 67*B*a^3*b^3 - 41*A*a^2*b^4)*c^2)*d^3 + 3*((D*a^4*b^2 + 3*C*a^3*b^3 - 15
*B*a^2*b^4 + 35*A*a*b^5)*d^5 - (13*D*a^3*b^3*c - (21*C*a^2*b^4 - 5*B*a*b^5
- 35*A*b^6)*c)*d^4 - 4*(3*D*a^2*b^4*c^2 + (4*C*a*b^5 - 5*B*b^6)*c^2)*d^3 +
8*(3*D*a*b^5*c^3 - C*b^6*c^3)*d^2)*x^3 - (91*D*a^4*b^2*c^3 + (5*C*a^3*b^3 -
```

```
 77*B*a^2*b^4 + 45*A*a*b^5)*c^3)*d^2 + (8*D*b^6*c^5 - (3*D*a^5*b - 15*C*a^4
*b^2 + 75*B*a^3*b^3 - 175*A*a^2*b^4)*d^5 - (21*D*a^4*b^2*c - (117*C*a^3*b^3
 - 85*B*a^2*b^4 - 35*A*a*b^5)*c)*d^4 - 4*(48*D*a^3*b^3*c^2 - (C*a^2*b^4 + 2
0*B*a*b^5 - 35*A*b^6)*c^2)*d^3 + 8*(19*D*a^2*b^4*c^3 - (13*C*a*b^5 - 10*B*b
^6)*c^3)*d^2 + 8*(7*D*a*b^5*c^4 - 4*C*b^6*c^4)*d)*x^2 + 2*(43*D*a^3*b^3*c^4
 - (25*C*a^2*b^4 - 3*B*a*b^5 - 3*A*b^6)*c^4)*d + (16*D*a*b^5*c^5 - 8*(3*B*a
^4*b^2 - 7*A*a^3*b^3)*d^5 - 2*(3*D*a^5*b*c - (39*C*a^4*b^2 - 55*B*a^3*b^3 +
 91*A*a^2*b^4)*c)*d^4 - (123*D*a^4*b^2*c^2 - (71*C*a^3*b^3 - 11*B*a^2*b^4 -
 217*A*a*b^5)*c^2)*d^3 - (35*D*a^3*b^3*c^3 + (61*C*a^2*b^4 - 133*B*a*b^5 +
21*A*b^6)*c^3)*d^2 + 4*(37*D*a^2*b^4*c^4 - (22*C*a*b^5 - 3*B*b^6)*c^4)*d)*x
)*sqrt(d*x + c))/(a^2*b^7*c^7*d - 5*a^3*b^6*c^6*d^2 + 10*a^4*b^5*c^5*d^3 -
10*a^5*b^4*c^4*d^4 + 5*a^6*b^3*c^3*d^5 - a^7*b^2*c^2*d^6 + (b^9*c^5*d^3 - 5
*a*b^8*c^4*d^4 + 10*a^2*b^7*c^3*d^5 - 10*a^3*b^6*c^2*d^6 + 5*a^4*b^5*c*d^7
 - a^5*b^4*d^8)*x^4 + 2*(b^9*c^6*d^2 - 4*a*b^8*c^5*d^3 + 5*a^2*b^7*c^4*d^4 -
 5*a^4*b^5*c^2*d^6 + 4*a^5*b^4*c*d^7 - a^6*b^3*d^8)*x^3 + (b^9*c^7*d - a*b^
8*c^6*d^2 - 9*a^2*b^7*c^5*d^3 + 25*a^3*b^6*c^4*d^4 - 25*a^4*b^5*c^3*d^5 + 9
*a^5*b^4*c^2*d^6 + a^6*b^3*c*d^7 - a^7*b^2*d^8)*x^2 + 2*(a*b^8*c^7*d - 4*a^
2*b^7*c^6*d^2 + 5*a^3*b^6*c^5*d^3 - 5*a^5*b^4*c^3*d^5 + 4*a^6*b^3*c^2*d^6 -
 a^7*b^2*c*d^7)*x), 1/12*(3*(((D*a^3*b^2 + 3*C*a^2*b^3 - 15*B*a*b^4 + 35*A*
b^5)*d^5 - 4*(3*D*a^2*b^3*c - (6*C*a*b^4 - 5*B*b^5)*c)*d^4 - 8*(3*D*a*b^4*c
^2 - C*b^5*c^2)*d^3)*x^4 + (D*a^5*c^2 + (3*C*a^4*b - 15*B*a^3*b^2 + 35*A*a^
2*b^3)*c^2)*d^3 + 2*((D*a^4*b + 3*C*a^3*b^2 - 15*B*a^2*b^3 + 35*A*a*b^4)*d^
5 - (11*D*a^3*b^2*c - (27*C*a^2*b^3 - 35*B*a*b^4 + 35*A*b^5)*c)*d^4 - 4*(9*
D*a^2*b^3*c^2 - (8*C*a*b^4 - 5*B*b^5)*c^2)*d^3 - 8*(3*D*a*b^4*c^3 - C*b^5*c
^3)*d^2)*x^3 - 4*(3*D*a^4*b*c^3 - (6*C*a^3*b^2 - 5*B*a^2*b^3)*c^3)*d^2 + ((
D*a^5 + 3*C*a^4*b - 15*B*a^3*b^2 + 35*A*a^2*b^3)*d^5 - 4*(2*D*a^4*b*c - (9*
C*a^3*b^2 - 20*B*a^2*b^3 + 35*A*a*b^4)*c)*d^4 - (71*D*a^3*b^2*c^2 - (107*C*
a^2*b^3 - 95*B*a*b^4 + 35*A*b^5)*c^2)*d^3 - 4*(27*D*a^2*b^3*c^3 - (14*C*a*b
^4 - 5*B*b^5)*c^3)*d^2 - 8*(3*D*a*b^4*c^4 - C*b^5*c^4)*d)*x^2 - 8*(3*D*a^3*
b^2*c^4 - C*a^2*b^3*c^4)*d + 2*((D*a^5*c + (3*C*a^4*b - 15*B*a^3*b^2 + 35*A
*a^2*b^3)*c)*d^4 - (11*D*a^4*b*c^2 - (27*C*a^3*b^2 - 35*B*a^2*b^3 + 35*A*a*
b^4)*c^2)*d^3 - 4*(9*D*a^3*b^2*c^3 - (8*C*a^2*b^3 - 5*B*a*b^4)*c^3)*d^2 - 8
*(3*D*a^2*b^3*c^4 - C*a*b^4*c^4)*d)*x)*sqrt(-b^2*c + a*b*d)*arctan(sqrt(-b^
2*c + a*b*d)*sqrt(d*x + c)/(b*d*x + b*c)) - (8*D*a^2*b^4*c^5 - 8*A*a^4*b^2*
d^5 - 8*(2*B*a^4*b^2 - 11*A*a^3*b^3)*c*d^4 - (3*D*a^5*b*c^2 - (55*C*a^4*b^2
 - 67*B*a^3*b^3 - 41*A*a^2*b^4)*c^2)*d^3 + 3*((D*a^4*b^2 + 3*C*a^3*b^3 - 15
*B*a^2*b^4 + 35*A*a*b^5)*d^5 - (13*D*a^3*b^3*c - (21*C*a^2*b^4 - 5*B*a*b^5
 - 35*A*b^6)*c)*d^4 - 4*(3*D*a^2*b^4*c^2 + (4*C*a*b^5 - 5*B*b^6)*c^2)*d^3 +
8*(3*D*a*b^5*c^3 - C*b^6*c^3)*d^2)*x^3 - (91*D*a^4*b^2*c^3 + (5*C*a^3*b^3 -
 77*B*a^2*b^4 + 45*A*a*b^5)*c^3)*d^2 + (8*D*b^6*c^5 - (3*D*a^5*b - 15*C*a^4
*b^2 + 75*B*a^3*b^3 - 175*A*a^2*b^4)*d^5 - (21*D*a^4*b^2*c - (117*C*a^3*b^3
 - 85*B*a^2*b^4 - 35*A*a*b^5)*c)*d^4 - 4*(48*D*a^3*b^3*c^2 - (C*a^2*b^4 + 2
0*B*a*b^5 - 35*A*b^6)*c^2)*d^3 + 8*(19*D*a^2*b^4*c^3 - (13*C*a*b^5 - 10*B*b
^6)*c^3)*d^2 + 8*(7*D*a*b^5*c^4 - 4*C*b^6*c^4)*d)*x^2 + 2*(43*D*a^3*b^3*c^4
 - (25*C*a^2*b^4 - 3*B*a*b^5 - 3*A*b^6)*c^4)*d + (16*D*a*b^5*c^5 - 8*(3*B*a
```

```
^4*b^2 - 7*A*a^3*b^3)*d^5 - 2*(3*D*a^5*b*c - (39*C*a^4*b^2 - 55*B*a^3*b^3 +
 91*A*a^2*b^4)*c)*d^4 - (123*D*a^4*b^2*c^2 - (71*C*a^3*b^3 - 11*B*a^2*b^4 -
 217*A*a*b^5)*c^2)*d^3 - (35*D*a^3*b^3*c^3 + (61*C*a^2*b^4 - 133*B*a*b^5 +
21*A*b^6)*c^3)*d^2 + 4*(37*D*a^2*b^4*c^4 - (22*C*a*b^5 - 3*B*b^6)*c^4)*d)*x
)*sqrt(d*x + c))/(a^2*b^7*c^7*d - 5*a^3*b^6*c^6*d^2 + 10*a^4*b^5*c^5*d^3 -
10*a^5*b^4*c^4*d^4 + 5*a^6*b^3*c^3*d^5 - a^7*b^2*c^2*d^6 + (b^9*c^5*d^3 - 5
*a*b^8*c^4*d^4 + 10*a^2*b^7*c^3*d^5 - 10*a^3*b^6*c^2*d^6 + 5*a^4*b^5*c*d^7
- a^5*b^4*d^8)*x^4 + 2*(b^9*c^6*d^2 - 4*a*b^8*c^5*d^3 + 5*a^2*b^7*c^4*d^4 -
 5*a^4*b^5*c^2*d^6 + 4*a^5*b^4*c*d^7 - a^6*b^3*d^8)*x^3 + (b^9*c^7*d - a*b^
8*c^6*d^2 - 9*a^2*b^7*c^5*d^3 + 25*a^3*b^6*c^4*d^4 - 25*a^4*b^5*c^3*d^5 + 9
*a^5*b^4*c^2*d^6 + a^6*b^3*c*d^7 - a^7*b^2*d^8)*x^2 + 2*(a*b^8*c^7*d - 4*a^
2*b^7*c^6*d^2 + 5*a^3*b^6*c^5*d^3 - 5*a^5*b^4*c^3*d^5 + 4*a^6*b^3*c^2*d^6 -
 a^7*b^2*c*d^7)*x)]
```

Sympy [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((D*x**3+C*x**2+B*x+A)/(b*x+a)**3/(d*x+c)**(5/2),x)
```

```
[Out] Timed out
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{5/2}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^3/(d*x+c)^(5/2),x, algorithm="maxima"
)
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [A] (verification not implemented)

none

Time = 0.33 (sec) , antiderivative size = 767, normalized size of antiderivative = 1.75

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{5/2}}\,dx =$$

$$-\frac{(24\,Dab^2c^2-8\,Cb^3c^2+12\,Da^2bcd-24\,Cab^2cd+20\,Bb^3cd-Da^3d^2-3\,Ca^2bd^2+15\,Bab^2d^2-35\,Ab^3d^2)}{4\,(b^5c^4-4\,ab^4c^3d+6\,a^2b^3c^2d^2-4\,a^3b^2cd^3+a^4bd^4)\sqrt{-b^2c+abd}}$$

$$-\frac{2\,(Dbc^4+9\,(dx+c)Dac^2d-3\,(dx+c)Cbc^2d-Dac^3d-Cbc^3d-6\,(dx+c)Cacd^2+6\,(dx+c)Bbcd^2+C}{3\,(b^4c^4d-4\,ab^3c^3d^2+6\,a^2b^2c^2d^3-4\,a^3bcd^4}$$

$$-\frac{12\,(dx+c)^{\frac{3}{2}}Da^2b^2cd-8\,(dx+c)^{\frac{3}{2}}Cab^3cd+4\,(dx+c)^{\frac{3}{2}}Bb^4cd-12\,\sqrt{dx+c}Da^2b^2c^2d+8\,\sqrt{dx+c}Cab^3c^2}{}$$

```
[In] integrate((D*x^3+C*x^2+B*x+A)/(b*x+a)^3/(d*x+c)^(5/2),x, algorithm="giac")
```

```
[Out] -1/4*(24*D*a*b^2*c^2 - 8*C*b^3*c^2 + 12*D*a^2*b*c*d - 24*C*a*b^2*c*d + 20*B
*b^3*c*d - D*a^3*d^2 - 3*C*a^2*b*d^2 + 15*B*a*b^2*d^2 - 35*A*b^3*d^2)*arcta
n(sqrt(d*x + c)*b/sqrt(-b^2*c + a*b*d))/((b^5*c^4 - 4*a*b^4*c^3*d + 6*a^2*b
^3*c^2*d^2 - 4*a^3*b^2*c*d^3 + a^4*b*d^4)*sqrt(-b^2*c + a*b*d)) - 2/3*(D*b*
c^4 + 9*(d*x + c)*D*a*c^2*d - 3*(d*x + c)*C*b*c^2*d - D*a*c^3*d - C*b*c^3*d
 - 6*(d*x + c)*C*a*c*d^2 + 6*(d*x + c)*B*b*c*d^2 + C*a*c^2*d^2 + B*b*c^2*d^
2 + 3*(d*x + c)*B*a*d^3 - 9*(d*x + c)*A*b*d^3 - B*a*c*d^3 - A*b*c*d^3 + A*a
*d^4)/((b^4*c^4*d - 4*a*b^3*c^3*d^2 + 6*a^2*b^2*c^2*d^3 - 4*a^3*b*c*d^4 + a
^4*d^5)*(d*x + c)^(3/2)) - 1/4*(12*(d*x + c)^(3/2)*D*a^2*b^2*c*d - 8*(d*x +
 c)^(3/2)*C*a*b^3*c*d + 4*(d*x + c)^(3/2)*B*b^4*c*d - 12*sqrt(d*x + c)*D*a^
2*b^2*c^2*d + 8*sqrt(d*x + c)*C*a*b^3*c^2*d - 4*sqrt(d*x + c)*B*b^4*c^2*d -
 (d*x + c)^(3/2)*D*a^3*b*d^2 - 3*(d*x + c)^(3/2)*C*a^2*b^2*d^2 + 7*(d*x + c
)^(3/2)*B*a*b^3*d^2 - 11*(d*x + c)^(3/2)*A*b^4*d^2 + 11*sqrt(d*x + c)*D*a^3
*b*c*d^2 - 3*sqrt(d*x + c)*C*a^2*b^2*c*d^2 - 5*sqrt(d*x + c)*B*a*b^3*c*d^2
+ 13*sqrt(d*x + c)*A*b^4*c*d^2 + sqrt(d*x + c)*D*a^4*d^3 - 5*sqrt(d*x + c)*
C*a^3*b*d^3 + 9*sqrt(d*x + c)*B*a^2*b^2*d^3 - 13*sqrt(d*x + c)*A*a*b^3*d^3)
/((b^5*c^4 - 4*a*b^4*c^3*d + 6*a^2*b^3*c^2*d^2 - 4*a^3*b^2*c*d^3 + a^4*b*d^
4)*((d*x + c)*b - b*c + a*d)^2)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx+Cx^2+Dx^3}{(a+bx)^3(c+dx)^{5/2}}\,dx=\int \frac{A+B\,x+C\,x^2+x^3\,D}{{\left(a+b\,x\right)}^3\,{\left(c+d\,x\right)}^{5/2}}\,dx$$

```
[In] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^3*(c + d*x)^(5/2)),x)
[Out] int((A + B*x + C*x^2 + x^3*D)/((a + b*x)^3*(c + d*x)^(5/2)), x)
```

3.25 $\int (a+bx)^3(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) dx$

Optimal result

Integrand size = 30, antiderivative size = 455

$$
\begin{aligned}
&\int (a+bx)^3(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) dx \\
&= -\frac{(bc-ad)^3\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)(c+dx)^{1+n}}{d^7(1+n)} \\
&\quad -\frac{(bc-ad)^2\left(ad\left(2cCd-Bd^2-3c^2D\right)-b\left(5c^2Cd-4Bcd^2+3Ad^3-6c^3D\right)\right)(c+dx)^{2+n}}{d^7(2+n)} \\
&\quad -\frac{(bc-ad)\left(a^2d^2(Cd-3cD)-abd\left(8cCd-3Bd^2-15c^2D\right)+b^2\left(10c^2Cd-6Bcd^2+3Ad^3-15c^3D\right)\right)(c+}{d^7(3+n)} \\
&\quad +\frac{\left(a^3d^3D+3a^2bd^2(Cd-4cD)-3ab^2d\left(4cCd-Bd^2-10c^2D\right)+b^3\left(10c^2Cd-4Bcd^2+Ad^3-20c^3D\right)\right)(c}{d^7(4+n)} \\
&\quad +\frac{b\left(3a^2d^2D+3abd(Cd-5cD)-b^2\left(5cCd-Bd^2-15c^2D\right)\right)(c+dx)^{5+n}}{d^7(5+n)} \\
&\quad +\frac{b^2(bCd-6bcD+3adD)(c+dx)^{6+n}}{d^7(6+n)}+\frac{b^3D(c+dx)^{7+n}}{d^7(7+n)}
\end{aligned}
$$

```
[Out] -(-a*d+b*c)^3*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)*(d*x+c)^(1+n)/d^7/(1+n)-(-a*d+b
*c)^2*(a*d*(-B*d^2+2*C*c*d-3*D*c^2)-b*(3*A*d^3-4*B*c*d^2+5*C*c^2*d-6*D*c^3)
)*(d*x+c)^(2+n)/d^7/(2+n)-(-a*d+b*c)*(a^2*d^2*(C*d-3*D*c)-a*b*d*(-3*B*d^2+8
*C*c*d-15*D*c^2)+b^2*(3*A*d^3-6*B*c*d^2+10*C*c^2*d-15*D*c^3))*(d*x+c)^(3+n)
/d^7/(3+n)+(a^3*d^3*D+3*a^2*b*d^2*(C*d-4*D*c)-3*a*b^2*d*(-B*d^2+4*C*c*d-10*
D*c^2)+b^3*(A*d^3-4*B*c*d^2+10*C*c^2*d-20*D*c^3))*(d*x+c)^(4+n)/d^7/(4+n)+b
*(3*a^2*d^2*D+3*a*b*d*(C*d-5*D*c)-b^2*(-B*d^2+5*C*c*d-15*D*c^2))*(d*x+c)^(5
+n)/d^7/(5+n)+b^2*(C*b*d+3*D*a*d-6*D*b*c)*(d*x+c)^(6+n)/d^7/(6+n)+b^3*D*(d*
x+c)^(7+n)/d^7/(7+n)
```

Rubi [A] (verified)

Time = 0.26 (sec) , antiderivative size = 455, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.033, Rules used = {1634}

$$
\begin{aligned}
&\int (a+bx)^3(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right)\,dx = \\
&-\frac{(bc-ad)(c+dx)^{n+3}\left(a^2d^2(Cd-3cD)-abd(-3Bd^2-15c^2D+8cCd)+b^2(3Ad^3-6Bcd^2-15c^3D+\right.}{d^7(n+3)} \\
&+\frac{b(c+dx)^{n+5}\left(3a^2d^2D+3abd(Cd-5cD)-(b^2(-Bd^2-15c^2D+5cCd))\right)}{d^7(n+5)} \\
&+\frac{(c+dx)^{n+4}\left(a^3d^3D+3a^2bd^2(Cd-4cD)-3ab^2d(-Bd^2-10c^2D+4cCd)+b^3(Ad^3-4Bcd^2-20c^3D\right.}{d^7(n+4)} \\
&-\frac{(bc-ad)^3(c+dx)^{n+1}\left(Ad^3-Bcd^2+c^3(-D)+c^2Cd\right)}{d^7(n+1)} \\
&-\frac{(bc-ad)^2(c+dx)^{n+2}\left(ad(-Bd^2-3c^2D+2cCd)-b(3Ad^3-4Bcd^2-6c^3D+5c^2Cd)\right)}{d^7(n+2)} \\
&+\frac{b^2(c+dx)^{n+6}(3adD-6bcD+bCd)}{d^7(n+6)}+\frac{b^3D(c+dx)^{n+7}}{d^7(n+7)}
\end{aligned}
$$

```
[In] Int[(a + b*x)^3*(c + d*x)^n*(A + B*x + C*x^2 + D*x^3),x]
```

```
[Out] -(((b*c - a*d)^3*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D)*(c + d*x)^(1 + n))/(d^
7*(1 + n))) - ((b*c - a*d)^2*(a*d*(2*c*C*d - B*d^2 - 3*c^2*D) - b*(5*c^2*C*
d - 4*B*c*d^2 + 3*A*d^3 - 6*c^3*D))*(c + d*x)^(2 + n))/(d^7*(2 + n)) - ((b*
c - a*d)*(a^2*d^2*(C*d - 3*c*D) - a*b*d*(8*c*C*d - 3*B*d^2 - 15*c^2*D) + b^
2*(10*c^2*C*d - 6*B*c*d^2 + 3*A*d^3 - 15*c^3*D))*(c + d*x)^(3 + n))/(d^7*(3
 + n)) + ((a^3*d^3*D + 3*a^2*b*d^2*(C*d - 4*c*D) - 3*a*b^2*d*(4*c*C*d - B*d
^2 - 10*c^2*D) + b^3*(10*c^2*C*d - 4*B*c*d^2 + A*d^3 - 20*c^3*D))*(c + d*x)
^(4 + n))/(d^7*(4 + n)) + (b*(3*a^2*d^2*D + 3*a*b*d*(C*d - 5*c*D) - b^2*(5*
c*C*d - B*d^2 - 15*c^2*D))*(c + d*x)^(5 + n))/(d^7*(5 + n)) + (b^2*(b*C*d -
 6*b*c*D + 3*a*d*D)*(c + d*x)^(6 + n))/(d^7*(6 + n)) + (b^3*D*(c + d*x)^(7
+ n))/(d^7*(7 + n))
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

$$\text{integral} = \int \left(\frac{(-bc+ad)^3\,(c^2Cd - Bcd^2 + Ad^3 - c^3D)\,(c+dx)^n}{d^6} + \frac{(bc-ad)^2\,(-ad(2cCd - Bd^2 - 3c^2D) + b(5c^2Cd - 4Bcd^2 + 3Ad^3 - 6c^3D))\,(c+dx)^{1+n}}{d^6} + \frac{(bc-ad)\,(-a^2d^2(Cd-3cD) + abd(8cCd - 3Bd^2 - 15c^2D) - b^2(10c^2Cd - 6Bcd^2 + 3Ad^3 - 15c^3D))\,(c}{d^6} + \frac{(a^3d^3D + 3a^2bd^2(Cd-4cD) - 3ab^2d(4cCd - Bd^2 - 10c^2D) + b^3(10c^2Cd - 4Bcd^2 + Ad^3 - 20c^3D))\,(c}{d^6} + \frac{b(3a^2d^2D + 3abd(Cd-5cD) - b^2(5cCd - Bd^2 - 15c^2D))\,(c+dx)^{4+n}}{d^6} + \frac{b^2(bCd - 6bcD + 3adD)(c+dx)^{5+n}}{d^6} + \frac{b^3D(c+dx)^{6+n}}{d^6} \right) dx$$

$$= -\frac{(bc-ad)^3\,(c^2Cd - Bcd^2 + Ad^3 - c^3D)\,(c+dx)^{1+n}}{d^7(1+n)} - \frac{(bc-ad)^2\,(ad(2cCd - Bd^2 - 3c^2D) - b(5c^2Cd - 4Bcd^2 + 3Ad^3 - 6c^3D))\,(c+dx)^{2+n}}{d^7(2+n)} - \frac{(bc-ad)\,(a^2d^2(Cd-3cD) - abd(8cCd - 3Bd^2 - 15c^2D) + b^2(10c^2Cd - 6Bcd^2 + 3Ad^3 - 15c^3}{d^7(3+n)} + \frac{(a^3d^3D + 3a^2bd^2(Cd-4cD) - 3ab^2d(4cCd - Bd^2 - 10c^2D) + b^3(10c^2Cd - 4Bcd^2 + Ad^3 - 20c}{d^7(4+n)} + \frac{b(3a^2d^2D + 3abd(Cd-5cD) - b^2(5cCd - Bd^2 - 15c^2D))\,(c+dx)^{5+n}}{d^7(5+n)} + \frac{b^2(bCd - 6bcD + 3adD)(c+dx)^{6+n}}{d^7(6+n)} + \frac{b^3D(c+dx)^{7+n}}{d^7(7+n)}$$

Mathematica [A] (verified)

Time = 2.03 (sec) , antiderivative size = 418, normalized size of antiderivative = 0.92

$$\int (a+bx)^3(c+dx)^n \left(A + Bx + Cx^2 + Dx^3\right) dx$$

$$= \frac{(c+dx)^{1+n} \left(\frac{(bc-ad)^3(-c^2Cd+Bcd^2-Ad^3+c^3D)}{1+n} - \frac{(bc-ad)^2(-ad(-2cCd+Bd^2+3c^2D)+b(-5c^2Cd+4Bcd^2-3Ad^3+6c^3D))(c+dx)}{2+n} + \right.}{}$$

```
[In] Integrate[(a + b*x)^3*(c + d*x)^n*(A + B*x + C*x^2 + D*x^3),x]
```

```
[Out] ((c + d*x)^(1 + n)*(((b*c - a*d)^3*(-(c^2*C*d) + B*c*d^2 - A*d^3 + c^3*D))/
(1 + n) - ((b*c - a*d)^2*(-(a*d*(-2*c*C*d + B*d^2 + 3*c^2*D)) + b*(-5*c^2*C
*d + 4*B*c*d^2 - 3*A*d^3 + 6*c^3*D))*(c + d*x))/(2 + n) + ((b*c - a*d)*(a^2
*d^2*(-(C*d) + 3*c*D) + a*b*d*(8*c*C*d - 3*B*d^2 - 15*c^2*D) + b^2*(-10*c^2
```

```
*C*d + 6*B*c*d^2 - 3*A*d^3 + 15*c^3*D))*(c + d*x)^2)/(3 + n) + ((a^3*d^3*D
+ 3*a^2*b*d^2*(C*d - 4*c*D) + 3*a*b^2*d*(-4*c*C*d + B*d^2 + 10*c^2*D) + b^3
*(10*c^2*C*d - 4*B*c*d^2 + A*d^3 - 20*c^3*D))*(c + d*x)^3)/(4 + n) + (b*(3*
a^2*d^2*D + 3*a*b*d*(C*d - 5*c*D) + b^2*(-5*c*C*d + B*d^2 + 15*c^2*D))*(c +
 d*x)^4)/(5 + n) + (b^2*(b*C*d - 6*b*c*D + 3*a*d*D)*(c + d*x)^5)/(6 + n) +
(b^3*D*(c + d*x)^6)/(7 + n)))/d^7
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 3931 vs. 2(455) = 910.

Time = 1.77 (sec) , antiderivative size = 3932, normalized size of antiderivative = 8.64

method	result	size
norman	Expression too large to display	3932
gosper	Expression too large to display	5003
parallelrisch	Expression too large to display	9319

```
[In] int((b*x+a)^3*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x,method=_RETURNVERBOSE)
```

```
[Out] D*b^3/(7+n)*x^7*exp(n*ln(d*x+c))+c*(A*a^3*d^6*n^6+27*A*a^3*d^6*n^5-3*A*a^2*
b*c*d^5*n^5-B*a^3*c*d^5*n^5+295*A*a^3*d^6*n^4-75*A*a^2*b*c*d^5*n^4+6*A*a*b^
2*c^2*d^4*n^4-25*B*a^3*c*d^5*n^4+6*B*a^2*b*c^2*d^4*n^4+2*C*a^3*c^2*d^4*n^4+
1665*A*a^3*d^6*n^3-735*A*a^2*b*c*d^5*n^3+132*A*a*b^2*c^2*d^4*n^3-6*A*b^3*c^
3*d^3*n^3-245*B*a^3*c*d^5*n^3+132*B*a^2*b*c^2*d^4*n^3-18*B*a*b^2*c^3*d^3*n^
3+44*C*a^3*c^2*d^4*n^3-18*C*a^2*b*c^3*d^3*n^3-6*D*a^3*c^3*d^3*n^3+5104*A*a^
3*d^6*n^2-3525*A*a^2*b*c*d^5*n^2+1074*A*a*b^2*c^2*d^4*n^2-108*A*b^3*c^3*d^3
*n^2-1175*B*a^3*c*d^5*n^2+1074*B*a^2*b*c^2*d^4*n^2-324*B*a*b^2*c^3*d^3*n^2+
24*B*b^3*c^4*d^2*n^2+358*C*a^3*c^2*d^4*n^2-324*C*a^2*b*c^3*d^3*n^2+72*C*a*b
^2*c^4*d^2*n^2-108*D*a^3*c^3*d^3*n^2+72*D*a^2*b*c^4*d^2*n^2+8028*A*a^3*d^6*
n-8262*A*a^2*b*c*d^5*n+3828*A*a*b^2*c^2*d^4*n-642*A*b^3*c^3*d^3*n-2754*B*a^
3*c*d^5*n+3828*B*a^2*b*c^2*d^4*n-1926*B*a*b^2*c^3*d^3*n+312*B*b^3*c^4*d^2*n
+1276*C*a^3*c^2*d^4*n-1926*C*a^2*b*c^3*d^3*n+936*C*a*b^2*c^4*d^2*n-120*C*b^
3*c^5*d*n-642*D*a^3*c^3*d^3*n+936*D*a^2*b*c^4*d^2*n-360*D*a*b^2*c^5*d*n+504
0*A*a^3*d^6-7560*A*a^2*b*c*d^5+5040*A*a*b^2*c^2*d^4-1260*A*b^3*c^3*d^3-2520
*B*a^3*c*d^5+5040*B*a^2*b*c^2*d^4-3780*B*a*b^2*c^3*d^3+1008*B*b^3*c^4*d^2+1
680*C*a^3*c^2*d^4-3780*C*a^2*b*c^3*d^3+3024*C*a*b^2*c^4*d^2-840*C*b^3*c^5*d
-1260*D*a^3*c^3*d^3+3024*D*a^2*b*c^4*d^2-2520*D*a*b^2*c^5*d+720*D*b^3*c^6)/
d^7/(n^7+28*n^6+322*n^5+1960*n^4+6769*n^3+13132*n^2+13068*n+5040)*exp(n*ln(
d*x+c))+(A*b^3*d^3*n^3+3*B*a*b^2*d^3*n^3+B*b^3*c*d^2*n^3+3*C*a^2*b*d^3*n^3+
3*C*a*b^2*c*d^2*n^3+D*a^3*d^3*n^3+3*D*a^2*b*c*d^2*n^3+18*A*b^3*d^3*n^2+54*B
*a*b^2*d^3*n^2+13*B*b^3*c*d^2*n^2+54*C*a^2*b*d^3*n^2+39*C*a*b^2*c*d^2*n^2-5
*C*b^3*c^2*d*n^2+18*D*a^3*d^3*n^2+39*D*a^2*b*c*d^2*n^2-15*D*a*b^2*c^2*d*n^2
+107*A*b^3*d^3*n+321*B*a*b^2*d^3*n+42*B*b^3*c*d^2*n+321*C*a^2*b*d^3*n+126*C
*a*b^2*c*d^2*n-35*C*b^3*c^2*d*n+107*D*a^3*d^3*n+126*D*a^2*b*c*d^2*n-105*D*a
```

```
*b^2*c^2*d*n+30*D*b^3*c^3*n+210*A*b^3*d^3+630*B*a*b^2*d^3+630*C*a^2*b*d^3+2
10*D*a^3*d^3)/d^3/(n^4+22*n^3+179*n^2+638*n+840)*x^4*exp(n*ln(d*x+c))+(3*A*
a*b^2*d^4*n^4+A*b^3*c*d^3*n^4+3*B*a^2*b*d^4*n^4+3*B*a*b^2*c*d^3*n^4+C*a^3*d
^4*n^4+3*C*a^2*b*c*d^3*n^4+D*a^3*c*d^3*n^4+66*A*a*b^2*d^4*n^3+18*A*b^3*c*d^
3*n^3+66*B*a^2*b*d^4*n^3+54*B*a*b^2*c*d^3*n^3-4*B*b^3*c^2*d^2*n^3+22*C*a^3*
d^4*n^3+54*C*a^2*b*c*d^3*n^3-12*C*a*b^2*c^2*d^2*n^3+18*D*a^3*c*d^3*n^3-12*D
*a^2*b*c^2*d^2*n^3+537*A*a*b^2*d^4*n^2+107*A*b^3*c*d^3*n^2+537*B*a^2*b*d^4*
n^2+321*B*a*b^2*c*d^3*n^2-52*B*b^3*c^2*d^2*n^2+179*C*a^3*d^4*n^2+321*C*a^2*
b*c*d^3*n^2-156*C*a*b^2*c^2*d^2*n^2+20*C*b^3*c^3*d*n^2+107*D*a^3*c*d^3*n^2-
156*D*a^2*b*c^2*d^2*n^2+60*D*a*b^2*c^3*d*n^2+1914*A*a*b^2*d^4*n+210*A*b^3*c
*d^3*n+1914*B*a^2*b*d^4*n+630*B*a*b^2*c*d^3*n-168*B*b^3*c^2*d^2*n+638*C*a^3
*d^4*n+630*C*a^2*b*c*d^3*n-504*C*a*b^2*c^2*d^2*n+140*C*b^3*c^3*d*n+210*D*a^
3*c*d^3*n-504*D*a^2*b*c^2*d^2*n+420*D*a*b^2*c^3*d*n-120*D*b^3*c^4*n+2520*A*
a*b^2*d^4+2520*B*a^2*b*d^4+840*C*a^3*d^4)/d^4/(n^5+25*n^4+245*n^3+1175*n^2+
2754*n+2520)*x^3*exp(n*ln(d*x+c))+(3*A*a^2*b*d^5*n^5+3*A*a*b^2*c*d^4*n^5+B*
a^3*d^5*n^5+3*B*a^2*b*c*d^4*n^5+C*a^3*c*d^4*n^5+75*A*a^2*b*d^5*n^4+66*A*a*b
^2*c*d^4*n^4-3*A*b^3*c^2*d^3*n^4+25*B*a^3*d^5*n^4+66*B*a^2*b*c*d^4*n^4-9*B*
a*b^2*c^2*d^3*n^4+22*C*a^3*c*d^4*n^4-9*C*a^2*b*c^2*d^3*n^4-3*D*a^3*c^2*d^3*
n^4+735*A*a^2*b*d^5*n^3+537*A*a*b^2*c*d^4*n^3-54*A*b^3*c^2*d^3*n^3+245*B*a^
3*d^5*n^3+537*B*a^2*b*c*d^4*n^3-162*B*a*b^2*c^2*d^3*n^3+12*B*b^3*c^3*d^2*n^
3+179*C*a^3*c*d^4*n^3-162*C*a^2*b*c^2*d^3*n^3+36*C*a*b^2*c^3*d^2*n^3-54*D*a
^3*c^2*d^3*n^3+36*D*a^2*b*c^3*d^2*n^3+3525*A*a^2*b*d^5*n^2+1914*A*a*b^2*c*d
^4*n^2-321*A*b^3*c^2*d^3*n^2+1175*B*a^3*d^5*n^2+1914*B*a^2*b*c*d^4*n^2-963*
B*a*b^2*c^2*d^3*n^2+156*B*b^3*c^3*d^2*n^2+638*C*a^3*c*d^4*n^2-963*C*a^2*b*c
^2*d^3*n^2+468*C*a*b^2*c^3*d^2*n^2-60*C*b^3*c^4*d*n^2-321*D*a^3*c^2*d^3*n^2
+468*D*a^2*b*c^3*d^2*n^2-180*D*a*b^2*c^4*d*n^2+8262*A*a^2*b*d^5*n+2520*A*a*
b^2*c*d^4*n-630*A*b^3*c^2*d^3*n+2754*B*a^3*d^5*n+2520*B*a^2*b*c*d^4*n-1890*
B*a*b^2*c^2*d^3*n+504*B*b^3*c^3*d^2*n+840*C*a^3*c*d^4*n-1890*C*a^2*b*c^2*d^
3*n+1512*C*a*b^2*c^3*d^2*n-420*C*b^3*c^4*d*n-630*D*a^3*c^2*d^3*n+1512*D*a^2
*b*c^3*d^2*n-1260*D*a*b^2*c^4*d*n+360*D*b^3*c^5*n+7560*A*a^2*b*d^5+2520*B*a
^3*d^5)/d^5/(n^6+27*n^5+295*n^4+1665*n^3+5104*n^2+8028*n+5040)*x^2*exp(n*ln
(d*x+c))+(A*a^3*d^6*n^6+3*A*a^2*b*c*d^5*n^6+B*a^3*c*d^5*n^6+27*A*a^3*d^6*n^
5+75*A*a^2*b*c*d^5*n^5-6*A*a*b^2*c^2*d^4*n^5+25*B*a^3*c*d^5*n^5-6*B*a^2*b*c
^2*d^4*n^5-2*C*a^3*c^2*d^4*n^5+295*A*a^3*d^6*n^4+735*A*a^2*b*c*d^5*n^4-132*
A*a*b^2*c^2*d^4*n^4+6*A*b^3*c^3*d^3*n^4+245*B*a^3*c*d^5*n^4-132*B*a^2*b*c^2
*d^4*n^4+18*B*a*b^2*c^3*d^3*n^4-44*C*a^3*c^2*d^4*n^4+18*C*a^2*b*c^3*d^3*n^4
+6*D*a^3*c^3*d^3*n^4+1665*A*a^3*d^6*n^3+3525*A*a^2*b*c*d^5*n^3-1074*A*a*b^2
*c^2*d^4*n^3+108*A*b^3*c^3*d^3*n^3+1175*B*a^3*c*d^5*n^3-1074*B*a^2*b*c^2*d^
4*n^3+324*B*a*b^2*c^3*d^3*n^3-24*B*b^3*c^4*d^2*n^3-358*C*a^3*c^2*d^4*n^3+32
4*C*a^2*b*c^3*d^3*n^3-72*C*a*b^2*c^4*d^2*n^3+108*D*a^3*c^3*d^3*n^3-72*D*a^2
*b*c^4*d^2*n^3+5104*A*a^3*d^6*n^2+8262*A*a^2*b*c*d^5*n^2-3828*A*a*b^2*c^2*d
^4*n^2+642*A*b^3*c^3*d^3*n^2+2754*B*a^3*c*d^5*n^2-3828*B*a^2*b*c^2*d^4*n^2+
1926*B*a*b^2*c^3*d^3*n^2-312*B*b^3*c^4*d^2*n^2-1276*C*a^3*c^2*d^4*n^2+1926*
C*a^2*b*c^3*d^3*n^2-936*C*a*b^2*c^4*d^2*n^2+120*C*b^3*c^5*d*n^2+642*D*a^3*c
^3*d^3*n^2-936*D*a^2*b*c^4*d^2*n^2+360*D*a*b^2*c^5*d*n^2+8028*A*a^3*d^6*n+7
```

```
560*A*a^2*b*c*d^5*n-5040*A*a*b^2*c^2*d^4*n+1260*A*b^3*c^3*d^3*n+2520*B*a^3*
c*d^5*n-5040*B*a^2*b*c^2*d^4*n+3780*B*a*b^2*c^3*d^3*n-1008*B*b^3*c^4*d^2*n-
1680*C*a^3*c^2*d^4*n+3780*C*a^2*b*c^3*d^3*n-3024*C*a*b^2*c^4*d^2*n+840*C*b^
3*c^5*d*n+1260*D*a^3*c^3*d^3*n-3024*D*a^2*b*c^4*d^2*n+2520*D*a*b^2*c^5*d*n-
720*D*b^3*c^6*n+5040*A*a^3*d^6)/d^6/(n^7+28*n^6+322*n^5+1960*n^4+6769*n^3+1
3132*n^2+13068*n+5040)*x*exp(n*ln(d*x+c))+b*(B*b^2*d^2*n^2+3*C*a*b*d^2*n^2+
C*b^2*c*d*n^2+3*D*a^2*d^2*n^2+3*D*a*b*c*d*n^2+13*B*b^2*d^2*n+39*C*a*b*d^2*n
+7*C*b^2*c*d*n+39*D*a^2*d^2*n+21*D*a*b*c*d*n-6*D*b^2*c^2*n+42*B*b^2*d^2+126
*C*a*b*d^2+126*D*a^2*d^2)/d^2/(n^3+18*n^2+107*n+210)*x^5*exp(n*ln(d*x+c))+(
C*b*d*n+3*D*a*d*n+D*b*c*n+7*C*b*d+21*D*a*d)*b^2/d/(n^2+13*n+42)*x^6*exp(n*l
n(d*x+c))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 4115 vs. $2(455) = 910$.

Time = 0.36 (sec) , antiderivative size = 4115, normalized size of antiderivative = 9.04

$$\int (a+bx)^3(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) \, dx = \text{Too large to display}$$

```
[In] integrate((b*x+a)^3*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="fricas")
```

```
[Out] (A*a^3*c*d^6*n^6 + 720*D*b^3*c^7 + 5040*A*a^3*c*d^6 + 1680*(C*a^3 + 3*B*a^2
*b + 3*A*a*b^2)*c^3*d^4 - 2520*(B*a^3 + 3*A*a^2*b)*c^2*d^5 + (D*b^3*d^7*n^6
 + 21*D*b^3*d^7*n^5 + 175*D*b^3*d^7*n^4 + 735*D*b^3*d^7*n^3 + 1624*D*b^3*d^
7*n^2 + 1764*D*b^3*d^7*n + 720*D*b^3*d^7)*x^7 + (840*(3*D*a*b^2 + C*b^3)*d^
7 + (D*b^3*c*d^6 + (3*D*a*b^2 + C*b^3)*d^7)*n^6 + (15*D*b^3*c*d^6 + 22*(3*D
*a*b^2 + C*b^3)*d^7)*n^5 + 5*(17*D*b^3*c*d^6 + 38*(3*D*a*b^2 + C*b^3)*d^7)*
n^4 + 5*(45*D*b^3*c*d^6 + 164*(3*D*a*b^2 + C*b^3)*d^7)*n^3 + (274*D*b^3*c*d
^6 + 1849*(3*D*a*b^2 + C*b^3)*d^7)*n^2 + 2*(60*D*b^3*c*d^6 + 1019*(3*D*a*b^
2 + C*b^3)*d^7)*n)*x^6 + (27*A*a^3*c*d^6 - (B*a^3 + 3*A*a^2*b)*c^2*d^5)*n^5
 + (1008*(3*D*a^2*b + 3*C*a*b^2 + B*b^3)*d^7 + ((3*D*a^2*b + 3*C*a*b^2 + B*
b^3)*d^7 + (3*D*a*b^2*c + C*b^3*c)*d^6)*n^6 - (6*D*b^3*c^2*d^5 - 23*(3*D*a^
2*b + 3*C*a*b^2 + B*b^3)*d^7 - 17*(3*D*a*b^2*c + C*b^3*c)*d^6)*n^5 - 3*(20*
D*b^3*c^2*d^5 - 69*(3*D*a^2*b + 3*C*a*b^2 + B*b^3)*d^7 - 35*(3*D*a*b^2*c +
C*b^3*c)*d^6)*n^4 - 5*(42*D*b^3*c^2*d^5 - 185*(3*D*a^2*b + 3*C*a*b^2 + B*b^
3)*d^7 - 59*(3*D*a*b^2*c + C*b^3*c)*d^6)*n^3 - 2*(150*D*b^3*c^2*d^5 - 1072*
(3*D*a^2*b + 3*C*a*b^2 + B*b^3)*d^7 - 187*(3*D*a*b^2*c + C*b^3*c)*d^6)*n^2
- 12*(12*D*b^3*c^2*d^5 - 201*(3*D*a^2*b + 3*C*a*b^2 + B*b^3)*d^7 - 14*(3*D*
a*b^2*c + C*b^3*c)*d^6)*n)*x^5 + (295*A*a^3*c*d^6 + 2*(C*a^3 + 3*B*a^2*b +
3*A*a*b^2)*c^3*d^4 - 25*(B*a^3 + 3*A*a^2*b)*c^2*d^5)*n^4 + (1260*(D*a^3 + 3
*C*a^2*b + 3*B*a*b^2 + A*b^3)*d^7 + ((D*a^3 + 3*C*a^2*b + 3*B*a*b^2 + A*b^3
)*d^7 + (3*D*a^2*b*c + (3*C*a*b^2 + B*b^3)*c)*d^6)*n^6 + (24*(D*a^3 + 3*C*a
^2*b + 3*B*a*b^2 + A*b^3)*d^7 + 19*(3*D*a^2*b*c + (3*C*a*b^2 + B*b^3)*c)*d^
6 - 5*(3*D*a*b^2*c^2 + C*b^3*c^2)*d^5)*n^5 + (30*D*b^3*c^3*d^4 + 226*(D*a^3
```

```
 + 3*C*a^2*b + 3*B*a*b^2 + A*b^3)*d^7 + 131*(3*D*a^2*b*c + (3*C*a*b^2 + B*b
^3)*c)*d^6 - 65*(3*D*a*b^2*c^2 + C*b^3*c^2)*d^5)*n^4 + (180*D*b^3*c^3*d^4 +
 1056*(D*a^3 + 3*C*a^2*b + 3*B*a*b^2 + A*b^3)*d^7 + 401*(3*D*a^2*b*c + (3*C
*a*b^2 + B*b^3)*c)*d^6 - 265*(3*D*a*b^2*c^2 + C*b^3*c^2)*d^5)*n^3 + 5*(66*D
*b^3*c^3*d^4 + 509*(D*a^3 + 3*C*a^2*b + 3*B*a*b^2 + A*b^3)*d^7 + 108*(3*D*a
^2*b*c + (3*C*a*b^2 + B*b^3)*c)*d^6 - 83*(3*D*a*b^2*c^2 + C*b^3*c^2)*d^5)*n
^2 + 6*(30*D*b^3*c^3*d^4 + 492*(D*a^3 + 3*C*a^2*b + 3*B*a*b^2 + A*b^3)*d^7
 + 42*(3*D*a^2*b*c + (3*C*a*b^2 + B*b^3)*c)*d^6 - 35*(3*D*a*b^2*c^2 + C*b^3*
c^2)*d^5)*n)*x^4 - 1260*(D*a^3*c^4 + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^4)*d
^3 + (1665*A*a^3*c*d^6 + 44*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^3*d^4 - 245*(
B*a^3 + 3*A*a^2*b)*c^2*d^5 - 6*(D*a^3*c^4 + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)
*c^4)*d^3)*n^3 + (1680*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*d^7 + ((C*a^3 + 3*B*
a^2*b + 3*A*a*b^2)*d^7 + (D*a^3*c + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c)*d^6)
*n^6 + (25*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*d^7 + 21*(D*a^3*c + (3*C*a^2*b +
 3*B*a*b^2 + A*b^3)*c)*d^6 - 4*(3*D*a^2*b*c^2 + (3*C*a*b^2 + B*b^3)*c^2)*d^
5)*n^5 + (247*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*d^7 + 163*(D*a^3*c + (3*C*a^2
*b + 3*B*a*b^2 + A*b^3)*c)*d^6 - 64*(3*D*a^2*b*c^2 + (3*C*a*b^2 + B*b^3)*c^
2)*d^5 + 20*(3*D*a*b^2*c^3 + C*b^3*c^3)*d^4)*n^4 - (120*D*b^3*c^4*d^3 - 121
9*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*d^7 - 567*(D*a^3*c + (3*C*a^2*b + 3*B*a*b
^2 + A*b^3)*c)*d^6 + 332*(3*D*a^2*b*c^2 + (3*C*a*b^2 + B*b^3)*c^2)*d^5 - 20
0*(3*D*a*b^2*c^3 + C*b^3*c^3)*d^4)*n^3 - 4*(90*D*b^3*c^4*d^3 - 778*(C*a^3 +
 3*B*a^2*b + 3*A*a*b^2)*d^7 - 211*(D*a^3*c + (3*C*a^2*b + 3*B*a*b^2 + A*b^3
)*c)*d^6 + 152*(3*D*a^2*b*c^2 + (3*C*a*b^2 + B*b^3)*c^2)*d^5 - 115*(3*D*a*b
^2*c^3 + C*b^3*c^3)*d^4)*n^2 - 4*(60*D*b^3*c^4*d^3 - 949*(C*a^3 + 3*B*a^2*b
 + 3*A*a*b^2)*d^7 - 105*(D*a^3*c + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c)*d^6 +
 84*(3*D*a^2*b*c^2 + (3*C*a*b^2 + B*b^3)*c^2)*d^5 - 70*(3*D*a*b^2*c^3 + C*b
^3*c^3)*d^4)*n)*x^3 + 1008*(3*D*a^2*b*c^5 + (3*C*a*b^2 + B*b^3)*c^5)*d^2 +
(5104*A*a^3*c*d^6 + 358*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^3*d^4 - 1175*(B*a
^3 + 3*A*a^2*b)*c^2*d^5 - 108*(D*a^3*c^4 + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*
c^4)*d^3 + 24*(3*D*a^2*b*c^5 + (3*C*a*b^2 + B*b^3)*c^5)*d^2)*n^2 + (2520*(B
*a^3 + 3*A*a^2*b)*d^7 + ((C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c*d^6 + (B*a^3 + 3
*A*a^2*b)*d^7)*n^6 + (23*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c*d^6 + 26*(B*a^3
 + 3*A*a^2*b)*d^7 - 3*(D*a^3*c^2 + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^2)*d^5)
*n^5 + 3*(67*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c*d^6 + 90*(B*a^3 + 3*A*a^2*b)
*d^7 - 19*(D*a^3*c^2 + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^2)*d^5 + 4*(3*D*a^
2*b*c^3 + (3*C*a*b^2 + B*b^3)*c^3)*d^4)*n^4 + (817*(C*a^3 + 3*B*a^2*b + 3*A
*a*b^2)*c*d^6 + 1420*(B*a^3 + 3*A*a^2*b)*d^7 - 375*(D*a^3*c^2 + (3*C*a^2*b
 + 3*B*a*b^2 + A*b^3)*c^2)*d^5 + 168*(3*D*a^2*b*c^3 + (3*C*a*b^2 + B*b^3)*c^
3)*d^4 - 60*(3*D*a*b^2*c^4 + C*b^3*c^4)*d^3)*n^3 + (360*D*b^3*c^5*d^2 + 147
8*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c*d^6 + 3929*(B*a^3 + 3*A*a^2*b)*d^7 - 95
1*(D*a^3*c^2 + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^2)*d^5 + 660*(3*D*a^2*b*c^
3 + (3*C*a*b^2 + B*b^3)*c^3)*d^4 - 480*(3*D*a*b^2*c^4 + C*b^3*c^4)*d^3)*n^2
 + 6*(60*D*b^3*c^5*d^2 + 140*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c*d^6 + 879*(B
*a^3 + 3*A*a^2*b)*d^7 - 105*(D*a^3*c^2 + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^
2)*d^5 + 84*(3*D*a^2*b*c^3 + (3*C*a*b^2 + B*b^3)*c^3)*d^4 - 70*(3*D*a*b^2*c
```

```
^4 + C*b^3*c^4)*d^3)*n)*x^2 - 840*(3*D*a*b^2*c^6 + C*b^3*c^6)*d + 2*(4014*A
*a^3*c*d^6 + 638*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^3*d^4 - 1377*(B*a^3 + 3*
A*a^2*b)*c^2*d^5 - 321*(D*a^3*c^4 + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^4)*d^
3 + 156*(3*D*a^2*b*c^5 + (3*C*a*b^2 + B*b^3)*c^5)*d^2 - 60*(3*D*a*b^2*c^6 +
 C*b^3*c^6)*d)*n + (5040*A*a^3*d^7 + (A*a^3*d^7 + (B*a^3 + 3*A*a^2*b)*c*d^6
)*n^6 + (27*A*a^3*d^7 - 2*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^2*d^5 + 25*(B*a
^3 + 3*A*a^2*b)*c*d^6)*n^5 + (295*A*a^3*d^7 - 44*(C*a^3 + 3*B*a^2*b + 3*A*a
*b^2)*c^2*d^5 + 245*(B*a^3 + 3*A*a^2*b)*c*d^6 + 6*(D*a^3*c^3 + (3*C*a^2*b +
 3*B*a*b^2 + A*b^3)*c^3)*d^4)*n^4 + (1665*A*a^3*d^7 - 358*(C*a^3 + 3*B*a^2*
b + 3*A*a*b^2)*c^2*d^5 + 1175*(B*a^3 + 3*A*a^2*b)*c*d^6 + 108*(D*a^3*c^3 +
(3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^3)*d^4 - 24*(3*D*a^2*b*c^4 + (3*C*a*b^2 +
 B*b^3)*c^4)*d^3)*n^3 + 2*(2552*A*a^3*d^7 - 638*(C*a^3 + 3*B*a^2*b + 3*A*a*
b^2)*c^2*d^5 + 1377*(B*a^3 + 3*A*a^2*b)*c*d^6 + 321*(D*a^3*c^3 + (3*C*a^2*b
 + 3*B*a*b^2 + A*b^3)*c^3)*d^4 - 156*(3*D*a^2*b*c^4 + (3*C*a*b^2 + B*b^3)*c
^4)*d^3 + 60*(3*D*a*b^2*c^5 + C*b^3*c^5)*d^2)*n^2 - 12*(60*D*b^3*c^6*d - 66
9*A*a^3*d^7 + 140*(C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^2*d^5 - 210*(B*a^3 + 3*
A*a^2*b)*c*d^6 - 105*(D*a^3*c^3 + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^3)*d^4
+ 84*(3*D*a^2*b*c^4 + (3*C*a*b^2 + B*b^3)*c^4)*d^3 - 70*(3*D*a*b^2*c^5 + C*
b^3*c^5)*d^2)*n)*x)*(d*x + c)^n/(d^7*n^7 + 28*d^7*n^6 + 322*d^7*n^5 + 1960*
d^7*n^4 + 6769*d^7*n^3 + 13132*d^7*n^2 + 13068*d^7*n + 5040*d^7)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 65321 vs. $2(445) = 890$.

Time = 11.92 (sec) , antiderivative size = 65321, normalized size of antiderivative = 143.56

$$\int (a+bx)^3(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx=\text{Too large to display}$$

```
[In] integrate((b*x+a)**3*(d*x+c)**n*(D*x**3+C*x**2+B*x+A),x)
```

```
[Out] Piecewise((c**n*(A*a**3*x + 3*A*a**2*b*x**2/2 + A*a*b**2*x**3 + A*b**3*x**4
/4 + B*a**3*x**2/2 + B*a**2*b*x**3 + 3*B*a*b**2*x**4/4 + B*b**3*x**5/5 + C*
a**3*x**3/3 + 3*C*a**2*b*x**4/4 + 3*C*a*b**2*x**5/5 + C*b**3*x**6/6 + D*a**
3*x**4/4 + 3*D*a**2*b*x**5/5 + D*a*b**2*x**6/2 + D*b**3*x**7/7), Eq(d, 0)),
 (-10*A*a**3*d**6/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 12
00*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6
) - 6*A*a**2*b*c*d**5/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2
+ 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*
x**6) - 36*A*a**2*b*d**6*x/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*
x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d
**13*x**6) - 3*A*a*b**2*c**2*d**4/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**
4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5
 + 60*d**13*x**6) - 18*A*a*b**2*c*d**5*x/(60*c**6*d**7 + 360*c**5*d**8*x +
900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**
```

```
12*x**5 + 60*d**13*x**6) - 45*A*a*b**2*d**6*x**2/(60*c**6*d**7 + 360*c**5*d
**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 3
60*c*d**12*x**5 + 60*d**13*x**6) - A*b**3*c**3*d**3/(60*c**6*d**7 + 360*c**
5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4
+ 360*c*d**12*x**5 + 60*d**13*x**6) - 6*A*b**3*c**2*d**4*x/(60*c**6*d**7 +
360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**1
1*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 15*A*b**3*c*d**5*x**2/(60*c**6
*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c
**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 20*A*b**3*d**6*x**3/(6
0*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 +
 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 2*B*a**3*c*d**5/
(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3
 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 12*B*a**3*d**6
*x/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x
**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 3*B*a**2*b*
c**2*d**4/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*
d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 18*B
*a**2*b*c*d**5*x/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 120
0*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6)
 - 45*B*a**2*b*d**6*x**2/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x*
*2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**
13*x**6) - 3*B*a*b**2*c**3*d**3/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*
d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 +
 60*d**13*x**6) - 18*B*a*b**2*c**2*d**4*x/(60*c**6*d**7 + 360*c**5*d**8*x +
 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d*
*12*x**5 + 60*d**13*x**6) - 45*B*a*b**2*c*d**5*x**2/(60*c**6*d**7 + 360*c**
5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4
+ 360*c*d**12*x**5 + 60*d**13*x**6) - 60*B*a*b**2*d**6*x**3/(60*c**6*d**7 +
 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**
11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 2*B*b**3*c**4*d**2/(60*c**6*d
**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**
2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 12*B*b**3*c**3*d**3*x/(6
0*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 +
 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 30*B*b**3*c**2*d
**4*x**2/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d
**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 40*B*
b**3*c*d**5*x**3/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 120
0*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6)
 - 30*B*b**3*d**6*x**4/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2
 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13
*x**6) - C*a**3*c**2*d**4/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x
**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d*
*13*x**6) - 6*C*a**3*c*d**5*x/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d*
*9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 6
0*d**13*x**6) - 15*C*a**3*d**6*x**2/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c
```

```
**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x*
*5 + 60*d**13*x**6) - 3*C*a**2*b*c**3*d**3/(60*c**6*d**7 + 360*c**5*d**8*x
+ 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d
**12*x**5 + 60*d**13*x**6) - 18*C*a**2*b*c**2*d**4*x/(60*c**6*d**7 + 360*c*
*5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4
 + 360*c*d**12*x**5 + 60*d**13*x**6) - 45*C*a**2*b*c*d**5*x**2/(60*c**6*d**
7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*
d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 60*C*a**2*b*d**6*x**3/(60*
c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 9
00*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 6*C*a*b**2*c**4*d*
*2/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x
**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 36*C*a*b**2
*c**3*d**3*x/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c*
*3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 9
0*C*a*b**2*c**2*d**4*x**2/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x
**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d*
*13*x**6) - 120*C*a*b**2*c*d**5*x**3/(60*c**6*d**7 + 360*c**5*d**8*x + 900*
c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x
**5 + 60*d**13*x**6) - 90*C*a*b**2*d**6*x**4/(60*c**6*d**7 + 360*c**5*d**8*
x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c
*d**12*x**5 + 60*d**13*x**6) - 10*C*b**3*c**5*d/(60*c**6*d**7 + 360*c**5*d*
*8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 36
0*c*d**12*x**5 + 60*d**13*x**6) - 60*C*b**3*c**4*d**2*x/(60*c**6*d**7 + 360
*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x
**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 150*C*b**3*c**3*d**3*x**2/(60*c**
6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*
c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 200*C*b**3*c**2*d**4*
x**3/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10
*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 150*C*b**
3*c*d**5*x**4/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c
**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) -
60*C*b**3*d**6*x**5/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 +
1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x*
*6) - D*a**3*c**3*d**3/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2
 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13
*x**6) - 6*D*a**3*c**2*d**4*x/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d*
*9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 6
0*d**13*x**6) - 15*D*a**3*c*d**5*x**2/(60*c**6*d**7 + 360*c**5*d**8*x + 900
*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*
x**5 + 60*d**13*x**6) - 20*D*a**3*d**6*x**3/(60*c**6*d**7 + 360*c**5*d**8*x
 + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*
d**12*x**5 + 60*d**13*x**6) - 6*D*a**2*b*c**4*d**2/(60*c**6*d**7 + 360*c**5
*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 +
 360*c*d**12*x**5 + 60*d**13*x**6) - 36*D*a**2*b*c**3*d**3*x/(60*c**6*d**7
+ 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d*
```

```
*11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 90*D*a**2*b*c**2*d**4*x**2/(
60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3
+ 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 120*D*a**2*b*c*
d**5*x**3/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*
d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 90*D
*a**2*b*d**6*x**4/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 12
00*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6
) - 30*D*a*b**2*c**5*d/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2
 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13
*x**6) - 180*D*a*b**2*c**4*d**2*x/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**
4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5
 + 60*d**13*x**6) - 450*D*a*b**2*c**3*d**3*x**2/(60*c**6*d**7 + 360*c**5*d*
*8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 36
0*c*d**12*x**5 + 60*d**13*x**6) - 600*D*a*b**2*c**2*d**4*x**3/(60*c**6*d**7
 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d
**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 450*D*a*b**2*c*d**5*x**4/(6
0*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 +
 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) - 180*D*a*b**2*d**
6*x**5/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**
10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) + 60*D*b*
*3*c**6*log(c/d + x)/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 +
 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x
**6) + 147*D*b**3*c**6/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2
 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13
*x**6) + 360*D*b**3*c**5*d*x*log(c/d + x)/(60*c**6*d**7 + 360*c**5*d**8*x +
 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d*
*12*x**5 + 60*d**13*x**6) + 822*D*b**3*c**5*d*x/(60*c**6*d**7 + 360*c**5*d*
*8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 36
0*c*d**12*x**5 + 60*d**13*x**6) + 900*D*b**3*c**4*d**2*x**2*log(c/d + x)/(6
0*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 +
 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) + 1875*D*b**3*c**4
*d**2*x**2/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3
*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) + 120
0*D*b**3*c**3*d**3*x**3*log(c/d + x)/(60*c**6*d**7 + 360*c**5*d**8*x + 900*
c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x
**5 + 60*d**13*x**6) + 2200*D*b**3*c**3*d**3*x**3/(60*c**6*d**7 + 360*c**5*
d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 +
360*c*d**12*x**5 + 60*d**13*x**6) + 900*D*b**3*c**2*d**4*x**4*log(c/d + x)/
(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3
 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) + 1350*D*b**3*c*
*2*d**4*x**4/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c*
*3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6) + 3
60*D*b**3*c*d**5*x**5*log(c/d + x)/(60*c**6*d**7 + 360*c**5*d**8*x + 900*c*
*4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*d**12*x**
5 + 60*d**13*x**6) + 360*D*b**3*c*d**5*x**5/(60*c**6*d**7 + 360*c**5*d**8*x
```

```
 + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*d**11*x**4 + 360*c*
d**12*x**5 + 60*d**13*x**6) + 60*D*b**3*d**6*x**6*log(c/d + x)/(60*c**6*d**
7 + 360*c**5*d**8*x + 900*c**4*d**9*x**2 + 1200*c**3*d**10*x**3 + 900*c**2*
d**11*x**4 + 360*c*d**12*x**5 + 60*d**13*x**6), Eq(n, -7)), (-12*A*a**3*d**
6/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**
3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 9*A*a**2*b*c*d**5/(60*c**5*d**7 + 3
00*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**
4 + 60*d**12*x**5) - 45*A*a**2*b*d**6*x/(60*c**5*d**7 + 300*c**4*d**8*x + 6
00*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5)
 - 6*A*a*b**2*c**2*d**4/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**
2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 30*A*a*b**2*c
*d**5*x/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**
10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 60*A*a*b**2*d**6*x**2/(60*c**
5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c
*d**11*x**4 + 60*d**12*x**5) - 3*A*b**3*c**3*d**3/(60*c**5*d**7 + 300*c**4*
d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d
**12*x**5) - 15*A*b**3*c**2*d**4*x/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c*
*3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 30
*A*b**3*c*d**5*x**2/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 +
600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 30*A*b**3*d**6*x*
*3/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x*
*3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 3*B*a**3*c*d**5/(60*c**5*d**7 + 30
0*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4
 + 60*d**12*x**5) - 15*B*a**3*d**6*x/(60*c**5*d**7 + 300*c**4*d**8*x + 600*
c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) -
6*B*a**2*b*c**2*d**4/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 +
 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 30*B*a**2*b*c*d*
*5*x/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*
x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 60*B*a**2*b*d**6*x**2/(60*c**5*d
**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d*
*11*x**4 + 60*d**12*x**5) - 9*B*a*b**2*c**3*d**3/(60*c**5*d**7 + 300*c**4*d
**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d*
*12*x**5) - 45*B*a*b**2*c**2*d**4*x/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c
**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 9
0*B*a*b**2*c*d**5*x**2/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2
 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 90*B*a*b**2*d*
*6*x**3/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**
10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 12*B*b**3*c**4*d**2/(60*c**5*
d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d
**11*x**4 + 60*d**12*x**5) - 60*B*b**3*c**3*d**3*x/(60*c**5*d**7 + 300*c**4
*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*
d**12*x**5) - 120*B*b**3*c**2*d**4*x**2/(60*c**5*d**7 + 300*c**4*d**8*x + 6
00*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5)
 - 120*B*b**3*c*d**5*x**3/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x
**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 60*B*b**3*d
```

```
**6*x**4/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d*
*10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 2*C*a**3*c**2*d**4/(60*c**5*
d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d
**11*x**4 + 60*d**12*x**5) - 10*C*a**3*c*d**5*x/(60*c**5*d**7 + 300*c**4*d*
*8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**
12*x**5) - 20*C*a**3*d**6*x**2/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d
**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 9*C*a*
*2*b*c**3*d**3/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c
**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 45*C*a**2*b*c**2*d**4*
x/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**
3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 90*C*a**2*b*c*d**5*x**2/(60*c**5*d*
*7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**
11*x**4 + 60*d**12*x**5) - 90*C*a**2*b*d**6*x**3/(60*c**5*d**7 + 300*c**4*d
**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d*
*12*x**5) - 36*C*a*b**2*c**4*d**2/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**
3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 180
*C*a*b**2*c**3*d**3*x/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2
 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 360*C*a*b**2*c*
*2*d**4*x**2/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**
2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 360*C*a*b**2*c*d**5*x**3
/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3
 + 300*c*d**11*x**4 + 60*d**12*x**5) - 180*C*a*b**2*d**6*x**4/(60*c**5*d**7
 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11
*x**4 + 60*d**12*x**5) + 60*C*b**3*c**5*d*log(c/d + x)/(60*c**5*d**7 + 300*
c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 +
 60*d**12*x**5) + 137*C*b**3*c**5*d/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c
**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) + 3
00*C*b**3*c**4*d**2*x*log(c/d + x)/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c*
*3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) + 62
5*C*b**3*c**4*d**2*x/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 +
 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) + 600*C*b**3*c**3*
d**3*x**2*log(c/d + x)/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2
 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) + 1100*C*b**3*c*
*3*d**3*x**2/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**
2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) + 600*C*b**3*c**2*d**4*x**
3*log(c/d + x)/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c
**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) + 900*C*b**3*c**2*d**4*x
**3/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x
**3 + 300*c*d**11*x**4 + 60*d**12*x**5) + 300*C*b**3*c*d**5*x**4*log(c/d +
x)/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x*
*3 + 300*c*d**11*x**4 + 60*d**12*x**5) + 300*C*b**3*c*d**5*x**4/(60*c**5*d*
*7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**
11*x**4 + 60*d**12*x**5) + 60*C*b**3*d**6*x**5*log(c/d + x)/(60*c**5*d**7 +
 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x
**4 + 60*d**12*x**5) - 3*D*a**3*c**3*d**3/(60*c**5*d**7 + 300*c**4*d**8*x +
```

```
 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**
5) - 15*D*a**3*c**2*d**4*x/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*
x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 30*D*a**3*
c*d**5*x**2/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2
*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 30*D*a**3*d**6*x**3/(60*c
**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300
*c*d**11*x**4 + 60*d**12*x**5) - 36*D*a**2*b*c**4*d**2/(60*c**5*d**7 + 300*
c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 +
 60*d**12*x**5) - 180*D*a**2*b*c**3*d**3*x/(60*c**5*d**7 + 300*c**4*d**8*x
+ 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x*
*5) - 360*D*a**2*b*c**2*d**4*x**2/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**
3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 360
*D*a**2*b*c*d**5*x**3/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2
+ 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 180*D*a**2*b*d*
*6*x**4/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**
10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) + 180*D*a*b**2*c**5*d*log(c/d +
 x)/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x
**3 + 300*c*d**11*x**4 + 60*d**12*x**5) + 411*D*a*b**2*c**5*d/(60*c**5*d**7
 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11
*x**4 + 60*d**12*x**5) + 900*D*a*b**2*c**4*d**2*x*log(c/d + x)/(60*c**5*d**
7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**1
1*x**4 + 60*d**12*x**5) + 1875*D*a*b**2*c**4*d**2*x/(60*c**5*d**7 + 300*c**
4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60
*d**12*x**5) + 1800*D*a*b**2*c**3*d**3*x**2*log(c/d + x)/(60*c**5*d**7 + 30
0*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4
 + 60*d**12*x**5) + 3300*D*a*b**2*c**3*d**3*x**2/(60*c**5*d**7 + 300*c**4*d
**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d*
*12*x**5) + 1800*D*a*b**2*c**2*d**4*x**3*log(c/d + x)/(60*c**5*d**7 + 300*c
**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 +
60*d**12*x**5) + 2700*D*a*b**2*c**2*d**4*x**3/(60*c**5*d**7 + 300*c**4*d**8
*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12
*x**5) + 900*D*a*b**2*c*d**5*x**4*log(c/d + x)/(60*c**5*d**7 + 300*c**4*d**
8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**1
2*x**5) + 900*D*a*b**2*c*d**5*x**4/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c*
*3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) + 18
0*D*a*b**2*d**6*x**5*log(c/d + x)/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**
3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 360
*D*b**3*c**6*log(c/d + x)/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x
**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 822*D*b**3*
c**6/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*
x**3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 1800*D*b**3*c**5*d*x*log(c/d + x
)/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**
3 + 300*c*d**11*x**4 + 60*d**12*x**5) - 3750*D*b**3*c**5*d*x/(60*c**5*d**7
+ 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*
x**4 + 60*d**12*x**5) - 3600*D*b**3*c**4*d**2*x**2*log(c/d + x)/(60*c**5*d*
```

```
*7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**
11*x**4 + 60*d**12*x**5) - 6600*D*b**3*c**4*d**2*x**2/(60*c**5*d**7 + 300*c
**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 +
60*d**12*x**5) - 3600*D*b**3*c**3*d**3*x**3*log(c/d + x)/(60*c**5*d**7 + 30
0*c**4*d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4
 + 60*d**12*x**5) - 5400*D*b**3*c**3*d**3*x**3/(60*c**5*d**7 + 300*c**4*d**
8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**1
2*x**5) - 1800*D*b**3*c**2*d**4*x**4*log(c/d + x)/(60*c**5*d**7 + 300*c**4*
d**8*x + 600*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d
**12*x**5) - 1800*D*b**3*c**2*d**4*x**4/(60*c**5*d**7 + 300*c**4*d**8*x + 6
00*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5)
 - 360*D*b**3*c*d**5*x**5*log(c/d + x)/(60*c**5*d**7 + 300*c**4*d**8*x + 60
0*c**3*d**9*x**2 + 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5)
+ 60*D*b**3*d**6*x**6/(60*c**5*d**7 + 300*c**4*d**8*x + 600*c**3*d**9*x**2
+ 600*c**2*d**10*x**3 + 300*c*d**11*x**4 + 60*d**12*x**5), Eq(n, -6)), (-3*
A*a**3*d**6/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10
*x**3 + 12*d**11*x**4) - 3*A*a**2*b*c*d**5/(12*c**4*d**7 + 48*c**3*d**8*x +
 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 12*A*a**2*b*d**6*x/
(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d
**11*x**4) - 3*A*a*b**2*c**2*d**4/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*
d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 12*A*a*b**2*c*d**5*x/(12*c**
4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x*
*4) - 18*A*a*b**2*d**6*x**2/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x
**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 3*A*b**3*c**3*d**3/(12*c**4*d**7 +
 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 12
*A*b**3*c**2*d**4*x/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48
*c*d**10*x**3 + 12*d**11*x**4) - 18*A*b**3*c*d**5*x**2/(12*c**4*d**7 + 48*c
**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 12*A*b*
*3*d**6*x**3/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**1
0*x**3 + 12*d**11*x**4) - B*a**3*c*d**5/(12*c**4*d**7 + 48*c**3*d**8*x + 72
*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 4*B*a**3*d**6*x/(12*c*
*4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x
**4) - 3*B*a**2*b*c**2*d**4/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x
**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 12*B*a**2*b*c*d**5*x/(12*c**4*d**7
 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) -
18*B*a**2*b*d**6*x**2/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 +
48*c*d**10*x**3 + 12*d**11*x**4) - 9*B*a*b**2*c**3*d**3/(12*c**4*d**7 + 48*
c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 36*B*a
*b**2*c**2*d**4*x/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c
*d**10*x**3 + 12*d**11*x**4) - 54*B*a*b**2*c*d**5*x**2/(12*c**4*d**7 + 48*c
**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 36*B*a*
b**2*d**6*x**3/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d*
*10*x**3 + 12*d**11*x**4) + 12*B*b**3*c**4*d**2*log(c/d + x)/(12*c**4*d**7
+ 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 2
5*B*b**3*c**4*d**2/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*
```

```
c*d**10*x**3 + 12*d**11*x**4) + 48*B*b**3*c**3*d**3*x*log(c/d + x)/(12*c**4
*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**
4) + 88*B*b**3*c**3*d**3*x/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x*
*2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 72*B*b**3*c**2*d**4*x**2*log(c/d +
x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 1
2*d**11*x**4) + 108*B*b**3*c**2*d**4*x**2/(12*c**4*d**7 + 48*c**3*d**8*x +
72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 48*B*b**3*c*d**5*x**
3*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d*
*10*x**3 + 12*d**11*x**4) + 48*B*b**3*c*d**5*x**3/(12*c**4*d**7 + 48*c**3*d
**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 12*B*b**3*d*
*6*x**4*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 4
8*c*d**10*x**3 + 12*d**11*x**4) - C*a**3*c**2*d**4/(12*c**4*d**7 + 48*c**3*
d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 4*C*a**3*c*
d**5*x/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3
 + 12*d**11*x**4) - 6*C*a**3*d**6*x**2/(12*c**4*d**7 + 48*c**3*d**8*x + 72*
c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 9*C*a**2*b*c**3*d**3/(1
2*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**
11*x**4) - 36*C*a**2*b*c**2*d**4*x/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2
*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 54*C*a**2*b*c*d**5*x**2/(12
*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**1
1*x**4) - 36*C*a**2*b*d**6*x**3/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d*
*9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 36*C*a*b**2*c**4*d**2*log(c/d
+ x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 +
 12*d**11*x**4) + 75*C*a*b**2*c**4*d**2/(12*c**4*d**7 + 48*c**3*d**8*x + 72
*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 144*C*a*b**2*c**3*d**3
*x*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d
**10*x**3 + 12*d**11*x**4) + 264*C*a*b**2*c**3*d**3*x/(12*c**4*d**7 + 48*c*
*3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 216*C*a*
b**2*c**2*d**4*x**2*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d
**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 324*C*a*b**2*c**2*d**4*x**2/(
12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d*
*11*x**4) + 144*C*a*b**2*c*d**5*x**3*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d
**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 144*C*a*b**2
*c*d**5*x**3/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**1
0*x**3 + 12*d**11*x**4) + 36*C*a*b**2*d**6*x**4*log(c/d + x)/(12*c**4*d**7
+ 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 6
0*C*b**3*c**5*d*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*
x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 125*C*b**3*c**5*d/(12*c**4*d**7 +
 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 24
0*C*b**3*c**4*d**2*x*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*
d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 440*C*b**3*c**4*d**2*x/(12*c
**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*
x**4) - 360*C*b**3*c**3*d**3*x**2*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8
*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 540*C*b**3*c**3
*d**3*x**2/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*
```

```
x**3 + 12*d**11*x**4) - 240*C*b**3*c**2*d**4*x**3*log(c/d + x)/(12*c**4*d**
7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) -
 240*C*b**3*c**2*d**4*x**3/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x*
*2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 60*C*b**3*c*d**5*x**4*log(c/d + x)/
(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d
**11*x**4) + 12*C*b**3*d**6*x**5/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d
**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 3*D*a**3*c**3*d**3/(12*c**4*d
**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4)
 - 12*D*a**3*c**2*d**4*x/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2
 + 48*c*d**10*x**3 + 12*d**11*x**4) - 18*D*a**3*c*d**5*x**2/(12*c**4*d**7 +
 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 12
*D*a**3*d**6*x**3/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c
*d**10*x**3 + 12*d**11*x**4) + 36*D*a**2*b*c**4*d**2*log(c/d + x)/(12*c**4*
d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4
) + 75*D*a**2*b*c**4*d**2/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**
2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 144*D*a**2*b*c**3*d**3*x*log(c/d + x
)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12
*d**11*x**4) + 264*D*a**2*b*c**3*d**3*x/(12*c**4*d**7 + 48*c**3*d**8*x + 72
*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 216*D*a**2*b*c**2*d**4
*x**2*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*
c*d**10*x**3 + 12*d**11*x**4) + 324*D*a**2*b*c**2*d**4*x**2/(12*c**4*d**7 +
 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 14
4*D*a**2*b*c*d**5*x**3*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**
2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 144*D*a**2*b*c*d**5*x**3/(
12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d*
*11*x**4) + 36*D*a**2*b*d**6*x**4*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8
*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 180*D*a*b**2*c*
*5*d*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c
*d**10*x**3 + 12*d**11*x**4) - 375*D*a*b**2*c**5*d/(12*c**4*d**7 + 48*c**3*
d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 720*D*a*b**
2*c**4*d**2*x*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x*
*2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 1320*D*a*b**2*c**4*d**2*x/(12*c**4*
d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4
) - 1080*D*a*b**2*c**3*d**3*x**2*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*
x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 1620*D*a*b**2*c*
*3*d**3*x**2/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**1
0*x**3 + 12*d**11*x**4) - 720*D*a*b**2*c**2*d**4*x**3*log(c/d + x)/(12*c**4
*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**
4) - 720*D*a*b**2*c**2*d**4*x**3/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d
**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 180*D*a*b**2*c*d**5*x**4*log(
c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x*
*3 + 12*d**11*x**4) + 36*D*a*b**2*d**6*x**5/(12*c**4*d**7 + 48*c**3*d**8*x
+ 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 180*D*b**3*c**6*lo
g(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*
x**3 + 12*d**11*x**4) + 375*D*b**3*c**6/(12*c**4*d**7 + 48*c**3*d**8*x + 72
```

```
*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 720*D*b**3*c**5*d*x*lo
g(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*
x**3 + 12*d**11*x**4) + 1320*D*b**3*c**5*d*x/(12*c**4*d**7 + 48*c**3*d**8*x
 + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 1080*D*b**3*c**4*
d**2*x**2*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 +
 48*c*d**10*x**3 + 12*d**11*x**4) + 1620*D*b**3*c**4*d**2*x**2/(12*c**4*d**
7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) +
 720*D*b**3*c**3*d**3*x**3*log(c/d + x)/(12*c**4*d**7 + 48*c**3*d**8*x + 72
*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) + 720*D*b**3*c**3*d**3*x
**3/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 +
12*d**11*x**4) + 180*D*b**3*c**2*d**4*x**4*log(c/d + x)/(12*c**4*d**7 + 48*
c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4) - 36*D*b
**3*c*d**5*x**5/(12*c**4*d**7 + 48*c**3*d**8*x + 72*c**2*d**9*x**2 + 48*c*d
**10*x**3 + 12*d**11*x**4) + 6*D*b**3*d**6*x**6/(12*c**4*d**7 + 48*c**3*d**
8*x + 72*c**2*d**9*x**2 + 48*c*d**10*x**3 + 12*d**11*x**4), Eq(n, -5)), (-2
*A*a**3*d**6/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3)
 - 3*A*a**2*b*c*d**5/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**
10*x**3) - 9*A*a**2*b*d**6*x/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2
 + 6*d**10*x**3) - 6*A*a*b**2*c**2*d**4/(6*c**3*d**7 + 18*c**2*d**8*x + 18*
c*d**9*x**2 + 6*d**10*x**3) - 18*A*a*b**2*c*d**5*x/(6*c**3*d**7 + 18*c**2*d
**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 18*A*a*b**2*d**6*x**2/(6*c**3*d**7
 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 6*A*b**3*c**3*d**3*log
(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) +
11*A*b**3*c**3*d**3/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**1
0*x**3) + 18*A*b**3*c**2*d**4*x*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x
+ 18*c*d**9*x**2 + 6*d**10*x**3) + 27*A*b**3*c**2*d**4*x/(6*c**3*d**7 + 18*
c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 18*A*b**3*c*d**5*x**2*log(c/
d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 18*
A*b**3*c*d**5*x**2/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10
*x**3) + 6*A*b**3*d**6*x**3*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18
*c*d**9*x**2 + 6*d**10*x**3) - B*a**3*c*d**5/(6*c**3*d**7 + 18*c**2*d**8*x
+ 18*c*d**9*x**2 + 6*d**10*x**3) - 3*B*a**3*d**6*x/(6*c**3*d**7 + 18*c**2*d
**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 6*B*a**2*b*c**2*d**4/(6*c**3*d**7
+ 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 18*B*a**2*b*c*d**5*x/(6
*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 18*B*a**2*b*
d**6*x**2/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) +
18*B*a*b**2*c**3*d**3*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**
9*x**2 + 6*d**10*x**3) + 33*B*a*b**2*c**3*d**3/(6*c**3*d**7 + 18*c**2*d**8*
x + 18*c*d**9*x**2 + 6*d**10*x**3) + 54*B*a*b**2*c**2*d**4*x*log(c/d + x)/(
6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 81*B*a*b**2
*c**2*d**4*x/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3)
 + 54*B*a*b**2*c*d**5*x**2*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*
c*d**9*x**2 + 6*d**10*x**3) + 54*B*a*b**2*c*d**5*x**2/(6*c**3*d**7 + 18*c**
2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 18*B*a*b**2*d**6*x**3*log(c/d +
 x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 24*B*b
```

```
**3*c**4*d**2*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 +
 6*d**10*x**3) - 44*B*b**3*c**4*d**2/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d
**9*x**2 + 6*d**10*x**3) - 72*B*b**3*c**3*d**3*x*log(c/d + x)/(6*c**3*d**7
+ 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 108*B*b**3*c**3*d**3*x/
(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 72*B*b**3*
c**2*d**4*x**2*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2
+ 6*d**10*x**3) - 72*B*b**3*c**2*d**4*x**2/(6*c**3*d**7 + 18*c**2*d**8*x +
18*c*d**9*x**2 + 6*d**10*x**3) - 24*B*b**3*c*d**5*x**3*log(c/d + x)/(6*c**3
*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 6*B*b**3*d**6*x**
4/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 2*C*a**3
*c**2*d**4/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) -
 6*C*a**3*c*d**5*x/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10
*x**3) - 6*C*a**3*d**6*x**2/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2
+ 6*d**10*x**3) + 18*C*a**2*b*c**3*d**3*log(c/d + x)/(6*c**3*d**7 + 18*c**2
*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 33*C*a**2*b*c**3*d**3/(6*c**3*d*
*7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 54*C*a**2*b*c**2*d**
4*x*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x
**3) + 81*C*a**2*b*c**2*d**4*x/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x*
*2 + 6*d**10*x**3) + 54*C*a**2*b*c*d**5*x**2*log(c/d + x)/(6*c**3*d**7 + 18
*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 54*C*a**2*b*c*d**5*x**2/(6*
c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 18*C*a**2*b*d
**6*x**3*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d*
*10*x**3) - 72*C*a*b**2*c**4*d**2*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*
x + 18*c*d**9*x**2 + 6*d**10*x**3) - 132*C*a*b**2*c**4*d**2/(6*c**3*d**7 +
18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 216*C*a*b**2*c**3*d**3*x*
log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3)
 - 324*C*a*b**2*c**3*d**3*x/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2
+ 6*d**10*x**3) - 216*C*a*b**2*c**2*d**4*x**2*log(c/d + x)/(6*c**3*d**7 + 1
8*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 216*C*a*b**2*c**2*d**4*x**
2/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 72*C*a*b
**2*c*d**5*x**3*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2
 + 6*d**10*x**3) + 18*C*a*b**2*d**6*x**4/(6*c**3*d**7 + 18*c**2*d**8*x + 18
*c*d**9*x**2 + 6*d**10*x**3) + 60*C*b**3*c**5*d*log(c/d + x)/(6*c**3*d**7 +
 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 110*C*b**3*c**5*d/(6*c**
3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 180*C*b**3*c**4*
d**2*x*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**1
0*x**3) + 270*C*b**3*c**4*d**2*x/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*
x**2 + 6*d**10*x**3) + 180*C*b**3*c**3*d**3*x**2*log(c/d + x)/(6*c**3*d**7
+ 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 180*C*b**3*c**3*d**3*x*
*2/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 60*C*b*
*3*c**2*d**4*x**3*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x*
*2 + 6*d**10*x**3) - 15*C*b**3*c*d**5*x**4/(6*c**3*d**7 + 18*c**2*d**8*x +
18*c*d**9*x**2 + 6*d**10*x**3) + 3*C*b**3*d**6*x**5/(6*c**3*d**7 + 18*c**2*
d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 6*D*a**3*c**3*d**3*log(c/d + x)/(
6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 11*D*a**3*c
```

```
**3*d**3/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 1
8*D*a**3*c**2*d**4*x*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9
*x**2 + 6*d**10*x**3) + 27*D*a**3*c**2*d**4*x/(6*c**3*d**7 + 18*c**2*d**8*x
 + 18*c*d**9*x**2 + 6*d**10*x**3) + 18*D*a**3*c*d**5*x**2*log(c/d + x)/(6*c
**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 18*D*a**3*c*d*
*5*x**2/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 6*
D*a**3*d**6*x**3*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**
2 + 6*d**10*x**3) - 72*D*a**2*b*c**4*d**2*log(c/d + x)/(6*c**3*d**7 + 18*c*
*2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 132*D*a**2*b*c**4*d**2/(6*c**3
*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 216*D*a**2*b*c**3
*d**3*x*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**
10*x**3) - 324*D*a**2*b*c**3*d**3*x/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d*
*9*x**2 + 6*d**10*x**3) - 216*D*a**2*b*c**2*d**4*x**2*log(c/d + x)/(6*c**3*
d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 216*D*a**2*b*c**2*
d**4*x**2/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) -
72*D*a**2*b*c*d**5*x**3*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d
**9*x**2 + 6*d**10*x**3) + 18*D*a**2*b*d**6*x**4/(6*c**3*d**7 + 18*c**2*d**
8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 180*D*a*b**2*c**5*d*log(c/d + x)/(6*
c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 330*D*a*b**2*
c**5*d/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 540
*D*a*b**2*c**4*d**2*x*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**
9*x**2 + 6*d**10*x**3) + 810*D*a*b**2*c**4*d**2*x/(6*c**3*d**7 + 18*c**2*d*
*8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 540*D*a*b**2*c**3*d**3*x**2*log(c/d
 + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 540*
D*a*b**2*c**3*d**3*x**2/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*
d**10*x**3) + 180*D*a*b**2*c**2*d**4*x**3*log(c/d + x)/(6*c**3*d**7 + 18*c*
*2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 45*D*a*b**2*c*d**5*x**4/(6*c**
3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 9*D*a*b**2*d**6*
x**5/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 120*D
*b**3*c**6*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*
d**10*x**3) - 220*D*b**3*c**6/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**
2 + 6*d**10*x**3) - 360*D*b**3*c**5*d*x*log(c/d + x)/(6*c**3*d**7 + 18*c**2
*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 540*D*b**3*c**5*d*x/(6*c**3*d**7
 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 360*D*b**3*c**4*d**2*x
**2*log(c/d + x)/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x
**3) - 360*D*b**3*c**4*d**2*x**2/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*
x**2 + 6*d**10*x**3) - 120*D*b**3*c**3*d**3*x**3*log(c/d + x)/(6*c**3*d**7
+ 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) + 30*D*b**3*c**2*d**4*x**
4/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3) - 6*D*b**3
*c*d**5*x**5/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d**10*x**3)
 + 2*D*b**3*d**6*x**6/(6*c**3*d**7 + 18*c**2*d**8*x + 18*c*d**9*x**2 + 6*d*
*10*x**3), Eq(n, -4)), (-6*A*a**3*d**6/(12*c**2*d**7 + 24*c*d**8*x + 12*d**
9*x**2) - 18*A*a**2*b*c*d**5/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) -
36*A*a**2*b*d**6*x/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 36*A*a*b**
2*c**2*d**4*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 54*A
```

```
*a*b**2*c**2*d**4/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 72*A*a*b**2
*c*d**5*x*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 72*A*a
*b**2*c*d**5*x/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 36*A*a*b**2*d*
*6*x**2*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 36*A*b**
3*c**3*d**3*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 54*A
*b**3*c**3*d**3/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 72*A*b**3*c**
2*d**4*x*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 72*A*b*
*3*c**2*d**4*x/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 36*A*b**3*c*d*
*5*x**2*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 12*A*b**
3*d**6*x**3/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 6*B*a**3*c*d**5/(
12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 12*B*a**3*d**6*x/(12*c**2*d**7
 + 24*c*d**8*x + 12*d**9*x**2) + 36*B*a**2*b*c**2*d**4*log(c/d + x)/(12*c**
2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 54*B*a**2*b*c**2*d**4/(12*c**2*d**7
+ 24*c*d**8*x + 12*d**9*x**2) + 72*B*a**2*b*c*d**5*x*log(c/d + x)/(12*c**2*
d**7 + 24*c*d**8*x + 12*d**9*x**2) + 72*B*a**2*b*c*d**5*x/(12*c**2*d**7 + 2
4*c*d**8*x + 12*d**9*x**2) + 36*B*a**2*b*d**6*x**2*log(c/d + x)/(12*c**2*d*
*7 + 24*c*d**8*x + 12*d**9*x**2) - 108*B*a*b**2*c**3*d**3*log(c/d + x)/(12*
c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 162*B*a*b**2*c**3*d**3/(12*c**2*d
**7 + 24*c*d**8*x + 12*d**9*x**2) - 216*B*a*b**2*c**2*d**4*x*log(c/d + x)/(
12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 216*B*a*b**2*c**2*d**4*x/(12*c
**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 108*B*a*b**2*c*d**5*x**2*log(c/d +
 x)/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 36*B*a*b**2*d**6*x**3/(12
*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 72*B*b**3*c**4*d**2*log(c/d + x)
/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 108*B*b**3*c**4*d**2/(12*c**
2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 144*B*b**3*c**3*d**3*x*log(c/d + x)/
(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 144*B*b**3*c**3*d**3*x/(12*c*
*2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 72*B*b**3*c**2*d**4*x**2*log(c/d +
x)/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 24*B*b**3*c*d**5*x**3/(12*
c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 6*B*b**3*d**6*x**4/(12*c**2*d**7
+ 24*c*d**8*x + 12*d**9*x**2) + 12*C*a**3*c**2*d**4*log(c/d + x)/(12*c**2*d
**7 + 24*c*d**8*x + 12*d**9*x**2) + 18*C*a**3*c**2*d**4/(12*c**2*d**7 + 24*
c*d**8*x + 12*d**9*x**2) + 24*C*a**3*c*d**5*x*log(c/d + x)/(12*c**2*d**7 +
24*c*d**8*x + 12*d**9*x**2) + 24*C*a**3*c*d**5*x/(12*c**2*d**7 + 24*c*d**8*
x + 12*d**9*x**2) + 12*C*a**3*d**6*x**2*log(c/d + x)/(12*c**2*d**7 + 24*c*d
**8*x + 12*d**9*x**2) - 108*C*a**2*b*c**3*d**3*log(c/d + x)/(12*c**2*d**7 +
 24*c*d**8*x + 12*d**9*x**2) - 162*C*a**2*b*c**3*d**3/(12*c**2*d**7 + 24*c*
d**8*x + 12*d**9*x**2) - 216*C*a**2*b*c**2*d**4*x*log(c/d + x)/(12*c**2*d**
7 + 24*c*d**8*x + 12*d**9*x**2) - 216*C*a**2*b*c**2*d**4*x/(12*c**2*d**7 +
24*c*d**8*x + 12*d**9*x**2) - 108*C*a**2*b*c*d**5*x**2*log(c/d + x)/(12*c**
2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 36*C*a**2*b*d**6*x**3/(12*c**2*d**7
+ 24*c*d**8*x + 12*d**9*x**2) + 216*C*a*b**2*c**4*d**2*log(c/d + x)/(12*c**
2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 324*C*a*b**2*c**4*d**2/(12*c**2*d**7
 + 24*c*d**8*x + 12*d**9*x**2) + 432*C*a*b**2*c**3*d**3*x*log(c/d + x)/(12*
c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 432*C*a*b**2*c**3*d**3*x/(12*c**2
*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 216*C*a*b**2*c**2*d**4*x**2*log(c/d +
```

```
 x)/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 72*C*a*b**2*c*d**5*x**3/(
12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 18*C*a*b**2*d**6*x**4/(12*c**2
*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 120*C*b**3*c**5*d*log(c/d + x)/(12*c*
*2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 180*C*b**3*c**5*d/(12*c**2*d**7 + 2
4*c*d**8*x + 12*d**9*x**2) - 240*C*b**3*c**4*d**2*x*log(c/d + x)/(12*c**2*d
**7 + 24*c*d**8*x + 12*d**9*x**2) - 240*C*b**3*c**4*d**2*x/(12*c**2*d**7 +
24*c*d**8*x + 12*d**9*x**2) - 120*C*b**3*c**3*d**3*x**2*log(c/d + x)/(12*c*
*2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 40*C*b**3*c**2*d**4*x**3/(12*c**2*d
**7 + 24*c*d**8*x + 12*d**9*x**2) - 10*C*b**3*c*d**5*x**4/(12*c**2*d**7 + 2
4*c*d**8*x + 12*d**9*x**2) + 4*C*b**3*d**6*x**5/(12*c**2*d**7 + 24*c*d**8*x
 + 12*d**9*x**2) - 36*D*a**3*c**3*d**3*log(c/d + x)/(12*c**2*d**7 + 24*c*d*
*8*x + 12*d**9*x**2) - 54*D*a**3*c**3*d**3/(12*c**2*d**7 + 24*c*d**8*x + 12
*d**9*x**2) - 72*D*a**3*c**2*d**4*x*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*
x + 12*d**9*x**2) - 72*D*a**3*c**2*d**4*x/(12*c**2*d**7 + 24*c*d**8*x + 12*
d**9*x**2) - 36*D*a**3*c*d**5*x**2*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*x
 + 12*d**9*x**2) + 12*D*a**3*d**6*x**3/(12*c**2*d**7 + 24*c*d**8*x + 12*d**
9*x**2) + 216*D*a**2*b*c**4*d**2*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*x +
 12*d**9*x**2) + 324*D*a**2*b*c**4*d**2/(12*c**2*d**7 + 24*c*d**8*x + 12*d*
*9*x**2) + 432*D*a**2*b*c**3*d**3*x*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*
x + 12*d**9*x**2) + 432*D*a**2*b*c**3*d**3*x/(12*c**2*d**7 + 24*c*d**8*x +
12*d**9*x**2) + 216*D*a**2*b*c**2*d**4*x**2*log(c/d + x)/(12*c**2*d**7 + 24
*c*d**8*x + 12*d**9*x**2) - 72*D*a**2*b*c*d**5*x**3/(12*c**2*d**7 + 24*c*d*
*8*x + 12*d**9*x**2) + 18*D*a**2*b*d**6*x**4/(12*c**2*d**7 + 24*c*d**8*x +
12*d**9*x**2) - 360*D*a*b**2*c**5*d*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*
x + 12*d**9*x**2) - 540*D*a*b**2*c**5*d/(12*c**2*d**7 + 24*c*d**8*x + 12*d*
*9*x**2) - 720*D*a*b**2*c**4*d**2*x*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*
x + 12*d**9*x**2) - 720*D*a*b**2*c**4*d**2*x/(12*c**2*d**7 + 24*c*d**8*x +
12*d**9*x**2) - 360*D*a*b**2*c**3*d**3*x**2*log(c/d + x)/(12*c**2*d**7 + 24
*c*d**8*x + 12*d**9*x**2) + 120*D*a*b**2*c**2*d**4*x**3/(12*c**2*d**7 + 24*
c*d**8*x + 12*d**9*x**2) - 30*D*a*b**2*c*d**5*x**4/(12*c**2*d**7 + 24*c*d**
8*x + 12*d**9*x**2) + 12*D*a*b**2*d**6*x**5/(12*c**2*d**7 + 24*c*d**8*x + 1
2*d**9*x**2) + 180*D*b**3*c**6*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*x + 1
2*d**9*x**2) + 270*D*b**3*c**6/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2)
+ 360*D*b**3*c**5*d*x*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x*
*2) + 360*D*b**3*c**5*d*x/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 180
*D*b**3*c**4*d**2*x**2*log(c/d + x)/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x
**2) - 60*D*b**3*c**3*d**3*x**3/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2)
 + 15*D*b**3*c**2*d**4*x**4/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) - 6
*D*b**3*c*d**5*x**5/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2) + 3*D*b**3*
d**6*x**6/(12*c**2*d**7 + 24*c*d**8*x + 12*d**9*x**2), Eq(n, -3)), (-60*A*a
**3*d**6/(60*c*d**7 + 60*d**8*x) + 180*A*a**2*b*c*d**5*log(c/d + x)/(60*c*d
**7 + 60*d**8*x) + 180*A*a**2*b*c*d**5/(60*c*d**7 + 60*d**8*x) + 180*A*a**2
*b*d**6*x*log(c/d + x)/(60*c*d**7 + 60*d**8*x) - 360*A*a*b**2*c**2*d**4*log
(c/d + x)/(60*c*d**7 + 60*d**8*x) - 360*A*a*b**2*c**2*d**4/(60*c*d**7 + 60*
d**8*x) - 360*A*a*b**2*c*d**5*x*log(c/d + x)/(60*c*d**7 + 60*d**8*x) + 180*
```

```
A*a*b**2*d**6*x**2/(60*c*d**7 + 60*d**8*x) + 180*A*b**3*c**3*d**3*log(c/d +
 x)/(60*c*d**7 + 60*d**8*x) + 180*A*b**3*c**3*d**3/(60*c*d**7 + 60*d**8*x)
+ 180*A*b**3*c**2*d**4*x*log(c/d + x)/(60*c*d**7 + 60*d**8*x) - 90*A*b**3*c
*d**5*x**2/(60*c*d**7 + 60*d**8*x) + 30*A*b**3*d**6*x**3/(60*c*d**7 + 60*d*
*8*x) + 60*B*a**3*c*d**5*log(c/d + x)/(60*c*d**7 + 60*d**8*x) + 60*B*a**3*c
*d**5/(60*c*d**7 + 60*d**8*x) + 60*B*a**3*d**6*x*log(c/d + x)/(60*c*d**7 +
60*d**8*x) - 360*B*a**2*b*c**2*d**4*log(c/d + x)/(60*c*d**7 + 60*d**8*x) -
360*B*a**2*b*c**2*d**4/(60*c*d**7 + 60*d**8*x) - 360*B*a**2*b*c*d**5*x*log(
c/d + x)/(60*c*d**7 + 60*d**8*x) + 180*B*a**2*b*d**6*x**2/(60*c*d**7 + 60*d
**8*x) + 540*B*a*b**2*c**3*d**3*log(c/d + x)/(60*c*d**7 + 60*d**8*x) + 540*
B*a*b**2*c**3*d**3/(60*c*d**7 + 60*d**8*x) + 540*B*a*b**2*c**2*d**4*x*log(c
/d + x)/(60*c*d**7 + 60*d**8*x) - 270*B*a*b**2*c*d**5*x**2/(60*c*d**7 + 60*
d**8*x) + 90*B*a*b**2*d**6*x**3/(60*c*d**7 + 60*d**8*x) - 240*B*b**3*c**4*d
**2*log(c/d + x)/(60*c*d**7 + 60*d**8*x) - 240*B*b**3*c**4*d**2/(60*c*d**7
+ 60*d**8*x) - 240*B*b**3*c**3*d**3*x*log(c/d + x)/(60*c*d**7 + 60*d**8*x)
+ 120*B*b**3*c**2*d**4*x**2/(60*c*d**7 + 60*d**8*x) - 40*B*b**3*c*d**5*x**3
/(60*c*d**7 + 60*d**8*x) + 20*B*b**3*d**6*x**4/(60*c*d**7 + 60*d**8*x) - 12
0*C*a**3*c**2*d**4*log(c/d + x)/(60*c*d**7 + 60*d**8*x) - 120*C*a**3*c**2*d
**4/(60*c*d**7 + 60*d**8*x) - 120*C*a**3*c*d**5*x*log(c/d + x)/(60*c*d**7 +
 60*d**8*x) + 60*C*a**3*d**6*x**2/(60*c*d**7 + 60*d**8*x) + 540*C*a**2*b*c*
*3*d**3*log(c/d + x)/(60*c*d**7 + 60*d**8*x) + 540*C*a**2*b*c**3*d**3/(60*c
*d**7 + 60*d**8*x) + 540*C*a**2*b*c**2*d**4*x*log(c/d + x)/(60*c*d**7 + 60*
d**8*x) - 270*C*a**2*b*c*d**5*x**2/(60*c*d**7 + 60*d**8*x) + 90*C*a**2*b*d*
*6*x**3/(60*c*d**7 + 60*d**8*x) - 720*C*a*b**2*c**4*d**2*log(c/d + x)/(60*c
*d**7 + 60*d**8*x) - 720*C*a*b**2*c**4*d**2/(60*c*d**7 + 60*d**8*x) - 720*C
*a*b**2*c**3*d**3*x*log(c/d + x)/(60*c*d**7 + 60*d**8*x) + 360*C*a*b**2*c**
2*d**4*x**2/(60*c*d**7 + 60*d**8*x) - 120*C*a*b**2*c*d**5*x**3/(60*c*d**7 +
 60*d**8*x) + 60*C*a*b**2*d**6*x**4/(60*c*d**7 + 60*d**8*x) + 300*C*b**3*c*
*5*d*log(c/d + x)/(60*c*d**7 + 60*d**8*x) + 300*C*b**3*c**5*d/(60*c*d**7 +
60*d**8*x) + 300*C*b**3*c**4*d**2*x*log(c/d + x)/(60*c*d**7 + 60*d**8*x) -
150*C*b**3*c**3*d**3*x**2/(60*c*d**7 + 60*d**8*x) + 50*C*b**3*c**2*d**4*x**
3/(60*c*d**7 + 60*d**8*x) - 25*C*b**3*c*d**5*x**4/(60*c*d**7 + 60*d**8*x) +
 15*C*b**3*d**6*x**5/(60*c*d**7 + 60*d**8*x) + 180*D*a**3*c**3*d**3*log(c/d
 + x)/(60*c*d**7 + 60*d**8*x) + 180*D*a**3*c**3*d**3/(60*c*d**7 + 60*d**8*x
) + 180*D*a**3*c**2*d**4*x*log(c/d + x)/(60*c*d**7 + 60*d**8*x) - 90*D*a**3
*c*d**5*x**2/(60*c*d**7 + 60*d**8*x) + 30*D*a**3*d**6*x**3/(60*c*d**7 + 60*
d**8*x) - 720*D*a**2*b*c**4*d**2*log(c/d + x)/(60*c*d**7 + 60*d**8*x) - 720
*D*a**2*b*c**4*d**2/(60*c*d**7 + 60*d**8*x) - 720*D*a**2*b*c**3*d**3*x*log(
c/d + x)/(60*c*d**7 + 60*d**8*x) + 360*D*a**2*b*c**2*d**4*x**2/(60*c*d**7 +
 60*d**8*x) - 120*D*a**2*b*c*d**5*x**3/(60*c*d**7 + 60*d**8*x) + 60*D*a**2*
b*d**6*x**4/(60*c*d**7 + 60*d**8*x) + 900*D*a*b**2*c**5*d*log(c/d + x)/(60*
c*d**7 + 60*d**8*x) + 900*D*a*b**2*c**5*d/(60*c*d**7 + 60*d**8*x) + 900*D*a
*b**2*c**4*d**2*x*log(c/d + x)/(60*c*d**7 + 60*d**8*x) - 450*D*a*b**2*c**3*
d**3*x**2/(60*c*d**7 + 60*d**8*x) + 150*D*a*b**2*c**2*d**4*x**3/(60*c*d**7
+ 60*d**8*x) - 75*D*a*b**2*c*d**5*x**4/(60*c*d**7 + 60*d**8*x) + 45*D*a*b**
```

```
2*d**6*x**5/(60*c*d**7 + 60*d**8*x) - 360*D*b**3*c**6*log(c/d + x)/(60*c*d*
*7 + 60*d**8*x) - 360*D*b**3*c**6/(60*c*d**7 + 60*d**8*x) - 360*D*b**3*c**5
*d*x*log(c/d + x)/(60*c*d**7 + 60*d**8*x) + 180*D*b**3*c**4*d**2*x**2/(60*c
*d**7 + 60*d**8*x) - 60*D*b**3*c**3*d**3*x**3/(60*c*d**7 + 60*d**8*x) + 30*
D*b**3*c**2*d**4*x**4/(60*c*d**7 + 60*d**8*x) - 18*D*b**3*c*d**5*x**5/(60*c
*d**7 + 60*d**8*x) + 12*D*b**3*d**6*x**6/(60*c*d**7 + 60*d**8*x), Eq(n, -2)
), (A*a**3*log(c/d + x)/d - 3*A*a**2*b*c*log(c/d + x)/d**2 + 3*A*a**2*b*x/d
 + 3*A*a*b**2*c**2*log(c/d + x)/d**3 - 3*A*a*b**2*c*x/d**2 + 3*A*a*b**2*x**
2/(2*d) - A*b**3*c**3*log(c/d + x)/d**4 + A*b**3*c**2*x/d**3 - A*b**3*c*x**
2/(2*d**2) + A*b**3*x**3/(3*d) - B*a**3*c*log(c/d + x)/d**2 + B*a**3*x/d +
3*B*a**2*b*c**2*log(c/d + x)/d**3 - 3*B*a**2*b*c*x/d**2 + 3*B*a**2*b*x**2/(
2*d) - 3*B*a*b**2*c**3*log(c/d + x)/d**4 + 3*B*a*b**2*c**2*x/d**3 - 3*B*a*b
**2*c*x**2/(2*d**2) + B*a*b**2*x**3/d + B*b**3*c**4*log(c/d + x)/d**5 - B*b
**3*c**3*x/d**4 + B*b**3*c**2*x**2/(2*d**3) - B*b**3*c*x**3/(3*d**2) + B*b*
*3*x**4/(4*d) + C*a**3*c**2*log(c/d + x)/d**3 - C*a**3*c*x/d**2 + C*a**3*x*
*2/(2*d) - 3*C*a**2*b*c**3*log(c/d + x)/d**4 + 3*C*a**2*b*c**2*x/d**3 - 3*C
*a**2*b*c*x**2/(2*d**2) + C*a**2*b*x**3/d + 3*C*a*b**2*c**4*log(c/d + x)/d*
*5 - 3*C*a*b**2*c**3*x/d**4 + 3*C*a*b**2*c**2*x**2/(2*d**3) - C*a*b**2*c*x*
*3/d**2 + 3*C*a*b**2*x**4/(4*d) - C*b**3*c**5*log(c/d + x)/d**6 + C*b**3*c*
*4*x/d**5 - C*b**3*c**3*x**2/(2*d**4) + C*b**3*c**2*x**3/(3*d**3) - C*b**3*
c*x**4/(4*d**2) + C*b**3*x**5/(5*d) - D*a**3*c**3*log(c/d + x)/d**4 + D*a**
3*c**2*x/d**3 - D*a**3*c*x**2/(2*d**2) + D*a**3*x**3/(3*d) + 3*D*a**2*b*c**
4*log(c/d + x)/d**5 - 3*D*a**2*b*c**3*x/d**4 + 3*D*a**2*b*c**2*x**2/(2*d**3
) - D*a**2*b*c*x**3/d**2 + 3*D*a**2*b*x**4/(4*d) - 3*D*a*b**2*c**5*log(c/d
 + x)/d**6 + 3*D*a*b**2*c**4*x/d**5 - 3*D*a*b**2*c**3*x**2/(2*d**4) + D*a*b*
*2*c**2*x**3/d**3 - 3*D*a*b**2*c*x**4/(4*d**2) + 3*D*a*b**2*x**5/(5*d) + D*
b**3*c**6*log(c/d + x)/d**7 - D*b**3*c**5*x/d**6 + D*b**3*c**4*x**2/(2*d**5
) - D*b**3*c**3*x**3/(3*d**4) + D*b**3*c**2*x**4/(4*d**3) - D*b**3*c*x**5/(
5*d**2) + D*b**3*x**6/(6*d), Eq(n, -1)), (A*a**3*c*d**6*n**6*(c + d*x)**n/(
d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3
+ 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 27*A*a**3*c*d**6*n**5*(c +
d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d
**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 295*A*a**3*c*d**6*
n**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**
4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1665*A*a
**3*c*d**6*n**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 19
60*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
 + 5104*A*a**3*c*d**6*n**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**
7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n +
 5040*d**7) + 8028*A*a**3*c*d**6*n*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*
d**7*n + 5040*d**7) + 5040*A*a**3*c*d**6*(c + d*x)**n/(d**7*n**7 + 28*d**7*
n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 +
13068*d**7*n + 5040*d**7) + A*a**3*d**7*n**6*x*(c + d*x)**n/(d**7*n**7 + 28
*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n
```

```
**2 + 13068*d**7*n + 5040*d**7) + 27*A*a**3*d**7*n**5*x*(c + d*x)**n/(d**7*
n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 131
32*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 295*A*a**3*d**7*n**4*x*(c + d*x)
**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*
n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1665*A*a**3*d**7*n**3*
x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 5104*A*a**3
*d**7*n**2*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*
d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) +
8028*A*a**3*d**7*n*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5
 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*
d**7) + 5040*A*a**3*d**7*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**
7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n +
 5040*d**7) - 3*A*a**2*b*c**2*d**5*n**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n
**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 1
3068*d**7*n + 5040*d**7) - 75*A*a**2*b*c**2*d**5*n**4*(c + d*x)**n/(d**7*n*
*7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132
*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 735*A*a**2*b*c**2*d**5*n**3*(c + d
*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d*
*7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 3525*A*a**2*b*c**2*
d**5*n**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**
7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 826
2*A*a**2*b*c**2*d**5*n*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n*
*5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 504
0*d**7) - 7560*A*a**2*b*c**2*d**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d
**7*n + 5040*d**7) + 3*A*a**2*b*c*d**6*n**6*x*(c + d*x)**n/(d**7*n**7 + 28*
d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n*
*2 + 13068*d**7*n + 5040*d**7) + 75*A*a**2*b*c*d**6*n**5*x*(c + d*x)**n/(d*
*7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 +
13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 735*A*a**2*b*c*d**6*n**4*x*(c
 + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 676
9*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3525*A*a**2*b*c
*d**6*n**3*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*
d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) +
8262*A*a**2*b*c*d**6*n**2*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d*
*7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
+ 5040*d**7) + 7560*A*a**2*b*c*d**6*n*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n
**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 1
3068*d**7*n + 5040*d**7) + 3*A*a**2*b*d**7*n**6*x**2*(c + d*x)**n/(d**7*n**
7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*
d**7*n**2 + 13068*d**7*n + 5040*d**7) + 78*A*a**2*b*d**7*n**5*x**2*(c + d*x
)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7
*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 810*A*a**2*b*d**7*n**
4*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n
```

```
**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 4260*A
*a**2*b*d**7*n**3*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n*
*5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 504
0*d**7) + 11787*A*a**2*b*d**7*n**2*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n
**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 1
3068*d**7*n + 5040*d**7) + 15822*A*a**2*b*d**7*n*x**2*(c + d*x)**n/(d**7*n*
*7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132
*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 7560*A*a**2*b*d**7*x**2*(c + d*x)*
*n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n
**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 6*A*a*b**2*c**3*d**4*n*
*4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 132*A*a*b*
*2*c**3*d**4*n**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 +
1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**
7) + 1074*A*a*b**2*c**3*d**4*n**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d
**7*n + 5040*d**7) + 3828*A*a*b**2*c**3*d**4*n*(c + d*x)**n/(d**7*n**7 + 28
*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n
**2 + 13068*d**7*n + 5040*d**7) + 5040*A*a*b**2*c**3*d**4*(c + d*x)**n/(d**
7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 1
3132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 6*A*a*b**2*c**2*d**5*n**5*x*(c
 + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 676
9*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 132*A*a*b**2*c*
*2*d**5*n**4*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 196
0*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
 - 1074*A*a*b**2*c**2*d**5*n**3*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 3
22*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d*
*7*n + 5040*d**7) - 3828*A*a*b**2*c**2*d**5*n**2*x*(c + d*x)**n/(d**7*n**7
 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d*
*7*n**2 + 13068*d**7*n + 5040*d**7) - 5040*A*a*b**2*c**2*d**5*n*x*(c + d*x)
**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*
n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3*A*a*b**2*c*d**6*n**6
*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n*
*4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 69*A*a*
b**2*c*d**6*n**5*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**
5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040
*d**7) + 603*A*a*b**2*c*d**6*n**4*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n*
*6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13
068*d**7*n + 5040*d**7) + 2451*A*a*b**2*c*d**6*n**3*x**2*(c + d*x)**n/(d**7
*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13
132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 4434*A*a*b**2*c*d**6*n**2*x**2*
(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6
769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 2520*A*a*b**2
*c*d**6*n*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 196
0*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
```

```
+ 3*A*a*b**2*d**7*n**6*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d*
*7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
+ 5040*d**7) + 75*A*a*b**2*d**7*n**5*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7
*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 +
 13068*d**7*n + 5040*d**7) + 741*A*a*b**2*d**7*n**4*x**3*(c + d*x)**n/(d**7
*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13
132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3657*A*a*b**2*d**7*n**3*x**3*(c
 + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 676
9*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 9336*A*a*b**2*d
**7*n**2*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960
*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) +
 11388*A*a*b**2*d**7*n*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d*
*7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
+ 5040*d**7) + 5040*A*a*b**2*d**7*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n*
*6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13
068*d**7*n + 5040*d**7) - 6*A*b**3*c**4*d**3*n**3*(c + d*x)**n/(d**7*n**7 +
 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**
7*n**2 + 13068*d**7*n + 5040*d**7) - 108*A*b**3*c**4*d**3*n**2*(c + d*x)**n
/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**
3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 642*A*b**3*c**4*d**3*n*(c
 + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 676
9*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 1260*A*b**3*c**
4*d**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n
**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 6*A*b*
*3*c**3*d**4*n**4*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5
+ 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d
**7) + 108*A*b**3*c**3*d**4*n**3*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*
d**7*n + 5040*d**7) + 642*A*b**3*c**3*d**4*n**2*x*(c + d*x)**n/(d**7*n**7 +
 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**
7*n**2 + 13068*d**7*n + 5040*d**7) + 1260*A*b**3*c**3*d**4*n*x*(c + d*x)**n
/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**
3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 3*A*b**3*c**2*d**5*n**5*x
**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 57*A*b**3
*c**2*d**5*n**4*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5
 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*
d**7) - 375*A*b**3*c**2*d**5*n**3*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n*
*6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13
068*d**7*n + 5040*d**7) - 951*A*b**3*c**2*d**5*n**2*x**2*(c + d*x)**n/(d**7
*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13
132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 630*A*b**3*c**2*d**5*n*x**2*(c
+ d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769
*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + A*b**3*c*d**6*n*
*6*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*
```

```
n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 21*A*
b**3*c*d**6*n**5*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**
5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040
*d**7) + 163*A*b**3*c*d**6*n**4*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6
 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 1306
8*d**7*n + 5040*d**7) + 567*A*b**3*c*d**6*n**3*x**3*(c + d*x)**n/(d**7*n**7
 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d
**7*n**2 + 13068*d**7*n + 5040*d**7) + 844*A*b**3*c*d**6*n**2*x**3*(c + d*x
)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7
*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 420*A*b**3*c*d**6*n*x
**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + A*b**3*d*
*7*n**6*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*
d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) +
24*A*b**3*d**7*n**5*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*
n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5
040*d**7) + 226*A*b**3*d**7*n**4*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**
6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 130
68*d**7*n + 5040*d**7) + 1056*A*b**3*d**7*n**3*x**4*(c + d*x)**n/(d**7*n**7
 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d
**7*n**2 + 13068*d**7*n + 5040*d**7) + 2545*A*b**3*d**7*n**2*x**4*(c + d*x)
**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*
n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 2952*A*b**3*d**7*n*x**
4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1260*A*b**3
*d**7*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d*
*7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - B*
a**3*c**2*d**5*n**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5
+ 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d
**7) - 25*B*a**3*c**2*d**5*n**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 32
2*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**
7*n + 5040*d**7) - 245*B*a**3*c**2*d**5*n**3*(c + d*x)**n/(d**7*n**7 + 28*d
**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**
2 + 13068*d**7*n + 5040*d**7) - 1175*B*a**3*c**2*d**5*n**2*(c + d*x)**n/(d*
*7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 +
13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 2754*B*a**3*c**2*d**5*n*(c +
d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d
**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 2520*B*a**3*c**2*d
**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + B*a**3*c*
d**6*n**6*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d
**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 2
5*B*a**3*c*d**6*n**5*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n*
*5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 504
0*d**7) + 245*B*a**3*c*d**6*n**4*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
```

```
 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*
d**7*n + 5040*d**7) + 1175*B*a**3*c*d**6*n**3*x*(c + d*x)**n/(d**7*n**7 + 2
8*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*
n**2 + 13068*d**7*n + 5040*d**7) + 2754*B*a**3*c*d**6*n**2*x*(c + d*x)**n/(
d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3
+ 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 2520*B*a**3*c*d**6*n*x*(c +
 d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*
d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + B*a**3*d**7*n**6*
x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**
4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 26*B*a**
3*d**7*n**5*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1
960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7
) + 270*B*a**3*d**7*n**4*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*
d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*
n + 5040*d**7) + 1420*B*a**3*d**7*n**3*x**2*(c + d*x)**n/(d**7*n**7 + 28*d*
*7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2
 + 13068*d**7*n + 5040*d**7) + 3929*B*a**3*d**7*n**2*x**2*(c + d*x)**n/(d**
7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 1
3132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 5274*B*a**3*d**7*n*x**2*(c + d
*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d*
*7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 2520*B*a**3*d**7*x*
*2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
+ 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 6*B*a**2*b
*c**3*d**4*n**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 19
60*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
 + 132*B*a**2*b*c**3*d**4*n**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322
*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7
*n + 5040*d**7) + 1074*B*a**2*b*c**3*d**4*n**2*(c + d*x)**n/(d**7*n**7 + 28
*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n
**2 + 13068*d**7*n + 5040*d**7) + 3828*B*a**2*b*c**3*d**4*n*(c + d*x)**n/(d
**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 +
 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 5040*B*a**2*b*c**3*d**4*(c +
 d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*
d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 6*B*a**2*b*c**2*d
**5*n**5*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d*
*7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 13
2*B*a**2*b*c**2*d**5*n**4*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d*
*7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
+ 5040*d**7) - 1074*B*a**2*b*c**2*d**5*n**3*x*(c + d*x)**n/(d**7*n**7 + 28*
d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n*
*2 + 13068*d**7*n + 5040*d**7) - 3828*B*a**2*b*c**2*d**5*n**2*x*(c + d*x)**
n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n*
*3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 5040*B*a**2*b*c**2*d**5*
n*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3*B*a**2*
```

```
b*c*d**6*n**6*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 +
 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d*
*7) + 69*B*a**2*b*c*d**6*n**5*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*
d**7*n + 5040*d**7) + 603*B*a**2*b*c*d**6*n**4*x**2*(c + d*x)**n/(d**7*n**7
 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d
**7*n**2 + 13068*d**7*n + 5040*d**7) + 2451*B*a**2*b*c*d**6*n**3*x**2*(c +
d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d
**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 4434*B*a**2*b*c*d*
*6*n**2*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*
d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) +
2520*B*a**2*b*c*d**6*n*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d*
*7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
+ 5040*d**7) + 3*B*a**2*b*d**7*n**6*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*
n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 +
13068*d**7*n + 5040*d**7) + 75*B*a**2*b*d**7*n**5*x**3*(c + d*x)**n/(d**7*n
**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 1313
2*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 741*B*a**2*b*d**7*n**4*x**3*(c +
d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d
**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3657*B*a**2*b*d**7
*n**3*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d*
*7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 93
36*B*a**2*b*d**7*n**2*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**
7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n +
 5040*d**7) + 11388*B*a**2*b*d**7*n*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*
n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 +
13068*d**7*n + 5040*d**7) + 5040*B*a**2*b*d**7*x**3*(c + d*x)**n/(d**7*n**7
 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d
**7*n**2 + 13068*d**7*n + 5040*d**7) - 18*B*a*b**2*c**4*d**3*n**3*(c + d*x)
**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*
n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 324*B*a*b**2*c**4*d**3
*n**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n*
*4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 1926*B*
a*b**2*c**4*d**3*n*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 +
 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d*
*7) - 3780*B*a*b**2*c**4*d**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*
d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*
n + 5040*d**7) + 18*B*a*b**2*c**3*d**4*n**4*x*(c + d*x)**n/(d**7*n**7 + 28*
d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n*
*2 + 13068*d**7*n + 5040*d**7) + 324*B*a*b**2*c**3*d**4*n**3*x*(c + d*x)**n
/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**
3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1926*B*a*b**2*c**3*d**4*n
**2*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n*
*4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3780*B*
a*b**2*c**3*d**4*n*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5
```

```
 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*
d**7) - 9*B*a*b**2*c**2*d**5*n**5*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n*
*6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13
068*d**7*n + 5040*d**7) - 171*B*a*b**2*c**2*d**5*n**4*x**2*(c + d*x)**n/(d*
*7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 +
13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 1125*B*a*b**2*c**2*d**5*n**3*
x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**
4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 2853*B*a
*b**2*c**2*d**5*n**2*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7
*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n +
5040*d**7) - 1890*B*a*b**2*c**2*d**5*n*x**2*(c + d*x)**n/(d**7*n**7 + 28*d*
*7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2
 + 13068*d**7*n + 5040*d**7) + 3*B*a*b**2*c*d**6*n**6*x**3*(c + d*x)**n/(d*
*7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 +
13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 63*B*a*b**2*c*d**6*n**5*x**3*
(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6
769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 489*B*a*b**2*
c*d**6*n**4*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1
960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7
) + 1701*B*a*b**2*c*d**6*n**3*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*
d**7*n + 5040*d**7) + 2532*B*a*b**2*c*d**6*n**2*x**3*(c + d*x)**n/(d**7*n**
7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*
d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1260*B*a*b**2*c*d**6*n*x**3*(c + d*
x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**
7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3*B*a*b**2*d**7*n**6
*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n*
*4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 72*B*a*
b**2*d**7*n**5*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5
+ 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d
**7) + 678*B*a*b**2*d**7*n**4*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*
d**7*n + 5040*d**7) + 3168*B*a*b**2*d**7*n**3*x**4*(c + d*x)**n/(d**7*n**7
+ 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d*
*7*n**2 + 13068*d**7*n + 5040*d**7) + 7635*B*a*b**2*d**7*n**2*x**4*(c + d*x
)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7
*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 8856*B*a*b**2*d**7*n*
x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**
4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3780*B*a
*b**2*d**7*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 19
60*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
 + 24*B*b**3*c**5*d**2*n**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d*
*7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
+ 5040*d**7) + 312*B*b**3*c**5*d**2*n*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**
6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 130
```

```
68*d**7*n + 5040*d**7) + 1008*B*b**3*c**5*d**2*(c + d*x)**n/(d**7*n**7 + 28
*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n
**2 + 13068*d**7*n + 5040*d**7) - 24*B*b**3*c**4*d**3*n**3*x*(c + d*x)**n/(
d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3
+ 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 312*B*b**3*c**4*d**3*n**2*x
*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 1008*B*b**3*
c**4*d**3*n*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960
*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) +
 12*B*b**3*c**3*d**4*n**4*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322
*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7
*n + 5040*d**7) + 168*B*b**3*c**3*d**4*n**3*x**2*(c + d*x)**n/(d**7*n**7 +
28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7
*n**2 + 13068*d**7*n + 5040*d**7) + 660*B*b**3*c**3*d**4*n**2*x**2*(c + d*x
)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7
*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 504*B*b**3*c**3*d**4*
n*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n
**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 4*B*b*
*3*c**2*d**5*n**5*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n*
*5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 504
0*d**7) - 64*B*b**3*c**2*d**5*n**4*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n
**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 1
3068*d**7*n + 5040*d**7) - 332*B*b**3*c**2*d**5*n**3*x**3*(c + d*x)**n/(d**
7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 1
3132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 608*B*b**3*c**2*d**5*n**2*x**3
*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 336*B*b**3*c
**2*d**5*n*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 19
60*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
 + B*b**3*c*d**6*n**6*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**
7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n +
 5040*d**7) + 19*B*b**3*c*d**6*n**5*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*
n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 +
13068*d**7*n + 5040*d**7) + 131*B*b**3*c*d**6*n**4*x**4*(c + d*x)**n/(d**7*
n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 131
32*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 401*B*b**3*c*d**6*n**3*x**4*(c +
 d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*
d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 540*B*b**3*c*d**6
*n**2*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d*
*7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 25
2*B*b**3*c*d**6*n*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n*
*5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 504
0*d**7) + B*b**3*d**7*n**6*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 32
2*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**
7*n + 5040*d**7) + 23*B*b**3*d**7*n**5*x**5*(c + d*x)**n/(d**7*n**7 + 28*d*
```

```
*7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2
 + 13068*d**7*n + 5040*d**7) + 207*B*b**3*d**7*n**4*x**5*(c + d*x)**n/(d**7
*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13
132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 925*B*b**3*d**7*n**3*x**5*(c +
d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d
**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 2144*B*b**3*d**7*n
**2*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7
*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 2412
*B*b**3*d**7*n*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5
+ 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d
**7) + 1008*B*b**3*d**7*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d
**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
 + 5040*d**7) + 2*C*a**3*c**3*d**4*n**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n
**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 1
3068*d**7*n + 5040*d**7) + 44*C*a**3*c**3*d**4*n**3*(c + d*x)**n/(d**7*n**7
 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d
**7*n**2 + 13068*d**7*n + 5040*d**7) + 358*C*a**3*c**3*d**4*n**2*(c + d*x)*
*n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n
**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1276*C*a**3*c**3*d**4*n
*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1680*C*a**3*
c**3*d**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**
7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 2*C
*a**3*c**2*d**5*n**5*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n*
*5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 504
0*d**7) - 44*C*a**3*c**2*d**5*n**4*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6
 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 1306
8*d**7*n + 5040*d**7) - 358*C*a**3*c**2*d**5*n**3*x*(c + d*x)**n/(d**7*n**7
 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d
**7*n**2 + 13068*d**7*n + 5040*d**7) - 1276*C*a**3*c**2*d**5*n**2*x*(c + d*
x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**
7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 1680*C*a**3*c**2*d**
5*n*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n*
*4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + C*a**3*
c*d**6*n**6*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1
960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7
) + 23*C*a**3*c*d**6*n**5*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322
*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7
*n + 5040*d**7) + 201*C*a**3*c*d**6*n**4*x**2*(c + d*x)**n/(d**7*n**7 + 28*
d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n*
*2 + 13068*d**7*n + 5040*d**7) + 817*C*a**3*c*d**6*n**3*x**2*(c + d*x)**n/(
d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3
+ 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1478*C*a**3*c*d**6*n**2*x**
2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 840*C*a**3*
```

```
c*d**6*n*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960
*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) +
 C*a**3*d**7*n**6*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n*
*5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 504
0*d**7) + 25*C*a**3*d**7*n**5*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*
d**7*n + 5040*d**7) + 247*C*a**3*d**7*n**4*x**3*(c + d*x)**n/(d**7*n**7 + 2
8*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*
n**2 + 13068*d**7*n + 5040*d**7) + 1219*C*a**3*d**7*n**3*x**3*(c + d*x)**n/
(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3
 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3112*C*a**3*d**7*n**2*x**3
*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3796*C*a**3*
d**7*n*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d
**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1
680*C*a**3*d**7*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5
 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*
d**7) - 18*C*a**2*b*c**4*d**3*n**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*
d**7*n + 5040*d**7) - 324*C*a**2*b*c**4*d**3*n**2*(c + d*x)**n/(d**7*n**7 +
 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**
7*n**2 + 13068*d**7*n + 5040*d**7) - 1926*C*a**2*b*c**4*d**3*n*(c + d*x)**n
/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**
3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 3780*C*a**2*b*c**4*d**3*(
c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 67
69*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 18*C*a**2*b*c*
*3*d**4*n**4*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 196
0*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
+ 324*C*a**2*b*c**3*d**4*n**3*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 32
2*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**
7*n + 5040*d**7) + 1926*C*a**2*b*c**3*d**4*n**2*x*(c + d*x)**n/(d**7*n**7 +
 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**
7*n**2 + 13068*d**7*n + 5040*d**7) + 3780*C*a**2*b*c**3*d**4*n*x*(c + d*x)*
*n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n
**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 9*C*a**2*b*c**2*d**5*n*
*5*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*
n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 171*C
*a**2*b*c**2*d**5*n**4*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d*
*7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
+ 5040*d**7) - 1125*C*a**2*b*c**2*d**5*n**3*x**2*(c + d*x)**n/(d**7*n**7 +
28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7
*n**2 + 13068*d**7*n + 5040*d**7) - 2853*C*a**2*b*c**2*d**5*n**2*x**2*(c +
d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d
**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 1890*C*a**2*b*c**2
*d**5*n*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*
```

```
d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) +
3*C*a**2*b*c*d**6*n**6*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d*
*7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
+ 5040*d**7) + 63*C*a**2*b*c*d**6*n**5*x**3*(c + d*x)**n/(d**7*n**7 + 28*d*
*7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2
 + 13068*d**7*n + 5040*d**7) + 489*C*a**2*b*c*d**6*n**4*x**3*(c + d*x)**n/(
d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3
+ 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1701*C*a**2*b*c*d**6*n**3*x
**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 2532*C*a*
*2*b*c*d**6*n**2*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**
5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040
*d**7) + 1260*C*a**2*b*c*d**6*n*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6
 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 1306
8*d**7*n + 5040*d**7) + 3*C*a**2*b*d**7*n**6*x**4*(c + d*x)**n/(d**7*n**7 +
 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**
7*n**2 + 13068*d**7*n + 5040*d**7) + 72*C*a**2*b*d**7*n**5*x**4*(c + d*x)**
n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n*
*3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 678*C*a**2*b*d**7*n**4*x
**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3168*C*a*
*2*b*d**7*n**3*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5
+ 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d
**7) + 7635*C*a**2*b*d**7*n**2*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6
+ 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068
*d**7*n + 5040*d**7) + 8856*C*a**2*b*d**7*n*x**4*(c + d*x)**n/(d**7*n**7 +
28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7
*n**2 + 13068*d**7*n + 5040*d**7) + 3780*C*a**2*b*d**7*x**4*(c + d*x)**n/(d
**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 +
 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 72*C*a*b**2*c**5*d**2*n**2*(
c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 67
69*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 936*C*a*b**2*c
**5*d**2*n*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d*
*7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 30
24*C*a*b**2*c**5*d**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**
5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040
*d**7) - 72*C*a*b**2*c**4*d**3*n**3*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**
6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 130
68*d**7*n + 5040*d**7) - 936*C*a*b**2*c**4*d**3*n**2*x*(c + d*x)**n/(d**7*n
**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 1313
2*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 3024*C*a*b**2*c**4*d**3*n*x*(c +
d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d
**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 36*C*a*b**2*c**3*d
**4*n**4*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960
*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) +
```

```
 504*C*a*b**2*c**3*d**4*n**3*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d
**7*n + 5040*d**7) + 1980*C*a*b**2*c**3*d**4*n**2*x**2*(c + d*x)**n/(d**7*n
**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 1313
2*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1512*C*a*b**2*c**3*d**4*n*x**2*(c
 + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 676
9*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 12*C*a*b**2*c**
2*d**5*n**5*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1
960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7
) - 192*C*a*b**2*c**2*d**5*n**4*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6
 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 1306
8*d**7*n + 5040*d**7) - 996*C*a*b**2*c**2*d**5*n**3*x**3*(c + d*x)**n/(d**7
*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13
132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 1824*C*a*b**2*c**2*d**5*n**2*x*
*3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
+ 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 1008*C*a*b
**2*c**2*d**5*n*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5
 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*
d**7) + 3*C*a*b**2*c*d**6*n**6*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6
+ 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068
*d**7*n + 5040*d**7) + 57*C*a*b**2*c*d**6*n**5*x**4*(c + d*x)**n/(d**7*n**7
 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d
**7*n**2 + 13068*d**7*n + 5040*d**7) + 393*C*a*b**2*c*d**6*n**4*x**4*(c + d
*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d*
*7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1203*C*a*b**2*c*d**
6*n**3*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d
**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1
620*C*a*b**2*c*d**6*n**2*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*
d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*
n + 5040*d**7) + 756*C*a*b**2*c*d**6*n*x**4*(c + d*x)**n/(d**7*n**7 + 28*d*
*7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2
 + 13068*d**7*n + 5040*d**7) + 3*C*a*b**2*d**7*n**6*x**5*(c + d*x)**n/(d**7
*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13
132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 69*C*a*b**2*d**7*n**5*x**5*(c +
 d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*
d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 621*C*a*b**2*d**7
*n**4*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d*
*7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 27
75*C*a*b**2*d**7*n**3*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**
7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n +
 5040*d**7) + 6432*C*a*b**2*d**7*n**2*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**
7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2
+ 13068*d**7*n + 5040*d**7) + 7236*C*a*b**2*d**7*n*x**5*(c + d*x)**n/(d**7*
n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 131
32*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3024*C*a*b**2*d**7*x**5*(c + d*x
```

```
)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7
*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 120*C*b**3*c**6*d*n*(
c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 67
69*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 840*C*b**3*c**
6*d*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 120*C*b**
3*c**5*d**2*n**2*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 +
 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d*
*7) + 840*C*b**3*c**5*d**2*n*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322
*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7
*n + 5040*d**7) - 60*C*b**3*c**4*d**3*n**3*x**2*(c + d*x)**n/(d**7*n**7 + 2
8*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*
n**2 + 13068*d**7*n + 5040*d**7) - 480*C*b**3*c**4*d**3*n**2*x**2*(c + d*x)
**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*
n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 420*C*b**3*c**4*d**3*n
*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n*
*4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 20*C*b*
*3*c**3*d**4*n**4*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n*
*5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 504
0*d**7) + 200*C*b**3*c**3*d**4*n**3*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*
n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 +
13068*d**7*n + 5040*d**7) + 460*C*b**3*c**3*d**4*n**2*x**3*(c + d*x)**n/(d*
*7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 +
13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 280*C*b**3*c**3*d**4*n*x**3*(
c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 67
69*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 5*C*b**3*c**2*
d**5*n**5*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 196
0*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
- 65*C*b**3*c**2*d**5*n**4*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 32
2*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**
7*n + 5040*d**7) - 265*C*b**3*c**2*d**5*n**3*x**4*(c + d*x)**n/(d**7*n**7 +
 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**
7*n**2 + 13068*d**7*n + 5040*d**7) - 415*C*b**3*c**2*d**5*n**2*x**4*(c + d*
x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**
7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 210*C*b**3*c**2*d**5
*n*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*
n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + C*b**
3*c*d**6*n**6*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 +
 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d*
*7) + 17*C*b**3*c*d**6*n**5*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 3
22*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d*
*7*n + 5040*d**7) + 105*C*b**3*c*d**6*n**4*x**5*(c + d*x)**n/(d**7*n**7 + 2
8*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*
n**2 + 13068*d**7*n + 5040*d**7) + 295*C*b**3*c*d**6*n**3*x**5*(c + d*x)**n
/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**
```

```
3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 374*C*b**3*c*d**6*n**2*x*
*5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
+ 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 168*C*b**3
*c*d**6*n*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 196
0*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
+ C*b**3*d**7*n**6*x**6*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n
**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 50
40*d**7) + 22*C*b**3*d**7*n**5*x**6*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6
+ 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068
*d**7*n + 5040*d**7) + 190*C*b**3*d**7*n**4*x**6*(c + d*x)**n/(d**7*n**7 +
28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7
*n**2 + 13068*d**7*n + 5040*d**7) + 820*C*b**3*d**7*n**3*x**6*(c + d*x)**n/
(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3
 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1849*C*b**3*d**7*n**2*x**6
*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 2038*C*b**3*
d**7*n*x**6*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d
**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 8
40*C*b**3*d**7*x**6*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5
+ 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d
**7) - 6*D*a**3*c**4*d**3*n**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322
*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7
*n + 5040*d**7) - 108*D*a**3*c**4*d**3*n**2*(c + d*x)**n/(d**7*n**7 + 28*d*
*7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2
 + 13068*d**7*n + 5040*d**7) - 642*D*a**3*c**4*d**3*n*(c + d*x)**n/(d**7*n*
*7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132
*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 1260*D*a**3*c**4*d**3*(c + d*x)**n
/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**
3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 6*D*a**3*c**3*d**4*n**4*x
*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 108*D*a**3*c
**3*d**4*n**3*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 19
60*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
 + 642*D*a**3*c**3*d**4*n**2*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322
*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7
*n + 5040*d**7) + 1260*D*a**3*c**3*d**4*n*x*(c + d*x)**n/(d**7*n**7 + 28*d*
*7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2
 + 13068*d**7*n + 5040*d**7) - 3*D*a**3*c**2*d**5*n**5*x**2*(c + d*x)**n/(d
**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 +
 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 57*D*a**3*c**2*d**5*n**4*x**
2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 375*D*a**3*
c**2*d**5*n**3*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5
+ 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d
**7) - 951*D*a**3*c**2*d**5*n**2*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**
```

```
6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 130
68*d**7*n + 5040*d**7) - 630*D*a**3*c**2*d**5*n*x**2*(c + d*x)**n/(d**7*n**
7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*
d**7*n**2 + 13068*d**7*n + 5040*d**7) + D*a**3*c*d**6*n**6*x**3*(c + d*x)**
n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n*
*3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 21*D*a**3*c*d**6*n**5*x*
*3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 163*D*a**3
*c*d**6*n**4*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 +
1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**
7) + 567*D*a**3*c*d**6*n**3*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 3
22*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d*
*7*n + 5040*d**7) + 844*D*a**3*c*d**6*n**2*x**3*(c + d*x)**n/(d**7*n**7 + 2
8*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*
n**2 + 13068*d**7*n + 5040*d**7) + 420*D*a**3*c*d**6*n*x**3*(c + d*x)**n/(d
**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 +
 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + D*a**3*d**7*n**6*x**4*(c + d
*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d*
*7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 24*D*a**3*d**7*n**5
*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n*
*4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 226*D*a
**3*d**7*n**4*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 +
 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d*
*7) + 1056*D*a**3*d**7*n**3*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 3
22*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d*
*7*n + 5040*d**7) + 2545*D*a**3*d**7*n**2*x**4*(c + d*x)**n/(d**7*n**7 + 28
*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n
**2 + 13068*d**7*n + 5040*d**7) + 2952*D*a**3*d**7*n*x**4*(c + d*x)**n/(d**
7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 1
3132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1260*D*a**3*d**7*x**4*(c + d*x
)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7
*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 72*D*a**2*b*c**5*d**2
*n**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n*
*4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 936*D*a
**2*b*c**5*d**2*n*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 +
1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**
7) + 3024*D*a**2*b*c**5*d**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d
**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
 + 5040*d**7) - 72*D*a**2*b*c**4*d**3*n**3*x*(c + d*x)**n/(d**7*n**7 + 28*d
**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**
2 + 13068*d**7*n + 5040*d**7) - 936*D*a**2*b*c**4*d**3*n**2*x*(c + d*x)**n/
(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3
 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 3024*D*a**2*b*c**4*d**3*n*
x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 36*D*a**2*b
```

```
*c**3*d**4*n**4*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5
 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*
d**7) + 504*D*a**2*b*c**3*d**4*n**3*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*
n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 +
13068*d**7*n + 5040*d**7) + 1980*D*a**2*b*c**3*d**4*n**2*x**2*(c + d*x)**n/
(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3
 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1512*D*a**2*b*c**3*d**4*n*
x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**
4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 12*D*a**
2*b*c**2*d**5*n**5*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n
**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 50
40*d**7) - 192*D*a**2*b*c**2*d**5*n**4*x**3*(c + d*x)**n/(d**7*n**7 + 28*d*
*7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2
 + 13068*d**7*n + 5040*d**7) - 996*D*a**2*b*c**2*d**5*n**3*x**3*(c + d*x)**
n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n*
*3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 1824*D*a**2*b*c**2*d**5*
n**2*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**
7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 100
8*D*a**2*b*c**2*d**5*n*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d*
*7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
+ 5040*d**7) + 3*D*a**2*b*c*d**6*n**6*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**
7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2
+ 13068*d**7*n + 5040*d**7) + 57*D*a**2*b*c*d**6*n**5*x**4*(c + d*x)**n/(d*
*7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 +
13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 393*D*a**2*b*c*d**6*n**4*x**4
*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1203*D*a**2*
b*c*d**6*n**3*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 +
 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d*
*7) + 1620*D*a**2*b*c*d**6*n**2*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6
 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 1306
8*d**7*n + 5040*d**7) + 756*D*a**2*b*c*d**6*n*x**4*(c + d*x)**n/(d**7*n**7
+ 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d*
*7*n**2 + 13068*d**7*n + 5040*d**7) + 3*D*a**2*b*d**7*n**6*x**5*(c + d*x)**
n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n*
*3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 69*D*a**2*b*d**7*n**5*x*
*5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
+ 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 621*D*a**2
*b*d**7*n**4*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 +
1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**
7) + 2775*D*a**2*b*d**7*n**3*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d
**7*n + 5040*d**7) + 6432*D*a**2*b*d**7*n**2*x**5*(c + d*x)**n/(d**7*n**7 +
 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**
7*n**2 + 13068*d**7*n + 5040*d**7) + 7236*D*a**2*b*d**7*n*x**5*(c + d*x)**n
```

```
/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**
3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3024*D*a**2*b*d**7*x**5*(
c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 67
69*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 360*D*a*b**2*c
**6*d*n*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*
n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 2520*
D*a*b**2*c**6*d*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 19
60*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
 + 360*D*a*b**2*c**5*d**2*n**2*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 3
22*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d*
*7*n + 5040*d**7) + 2520*D*a*b**2*c**5*d**2*n*x*(c + d*x)**n/(d**7*n**7 + 2
8*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*
n**2 + 13068*d**7*n + 5040*d**7) - 180*D*a*b**2*c**4*d**3*n**3*x**2*(c + d*
x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**
7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 1440*D*a*b**2*c**4*d
**3*n**2*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960
*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) -
 1260*D*a*b**2*c**4*d**3*n*x**2*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 32
2*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**
7*n + 5040*d**7) + 60*D*a*b**2*c**3*d**4*n**4*x**3*(c + d*x)**n/(d**7*n**7
 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d*
*7*n**2 + 13068*d**7*n + 5040*d**7) + 600*D*a*b**2*c**3*d**4*n**3*x**3*(c +
 d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*
d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 1380*D*a*b**2*c**
3*d**4*n**2*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1
960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7
) + 840*D*a*b**2*c**3*d**4*n*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d
**7*n + 5040*d**7) - 15*D*a*b**2*c**2*d**5*n**5*x**4*(c + d*x)**n/(d**7*n**
7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*
d**7*n**2 + 13068*d**7*n + 5040*d**7) - 195*D*a*b**2*c**2*d**5*n**4*x**4*(c
 + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 676
9*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 795*D*a*b**2*c*
*2*d**5*n**3*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 +
1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**
7) - 1245*D*a*b**2*c**2*d**5*n**2*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n*
*6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13
068*d**7*n + 5040*d**7) - 630*D*a*b**2*c**2*d**5*n*x**4*(c + d*x)**n/(d**7*
n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 131
32*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3*D*a*b**2*c*d**6*n**6*x**5*(c +
 d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*
d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 51*D*a*b**2*c*d**
6*n**5*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d
**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3
15*D*a*b**2*c*d**6*n**4*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d
```

```
**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
 + 5040*d**7) + 885*D*a*b**2*c*d**6*n**3*x**5*(c + d*x)**n/(d**7*n**7 + 28*
d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n*
*2 + 13068*d**7*n + 5040*d**7) + 1122*D*a*b**2*c*d**6*n**2*x**5*(c + d*x)**
n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n*
*3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 504*D*a*b**2*c*d**6*n*x*
*5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4
+ 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 3*D*a*b**2
*d**7*n**6*x**6*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 19
60*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
 + 66*D*a*b**2*d**7*n**5*x**6*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*
d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*
n + 5040*d**7) + 570*D*a*b**2*d**7*n**4*x**6*(c + d*x)**n/(d**7*n**7 + 28*d
**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**
2 + 13068*d**7*n + 5040*d**7) + 2460*D*a*b**2*d**7*n**3*x**6*(c + d*x)**n/(
d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3
+ 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 5547*D*a*b**2*d**7*n**2*x**
6*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 6114*D*a*b*
*2*d**7*n*x**6*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 196
0*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
+ 2520*D*a*b**2*d**7*x**6*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7
*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n +
5040*d**7) + 720*D*b**3*c**7*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d
**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
 + 5040*d**7) - 720*D*b**3*c**6*d*n*x*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**
6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 130
68*d**7*n + 5040*d**7) + 360*D*b**3*c**5*d**2*n**2*x**2*(c + d*x)**n/(d**7*
n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 131
32*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 360*D*b**3*c**5*d**2*n*x**2*(c +
 d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*
d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 120*D*b**3*c**4*d
**3*n**3*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960
*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) -
 360*D*b**3*c**4*d**3*n**2*x**3*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 32
2*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**
7*n + 5040*d**7) - 240*D*b**3*c**4*d**3*n*x**3*(c + d*x)**n/(d**7*n**7 + 28
*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n
**2 + 13068*d**7*n + 5040*d**7) + 30*D*b**3*c**3*d**4*n**4*x**4*(c + d*x)**
n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n*
*3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 180*D*b**3*c**3*d**4*n**
3*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n
**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 330*D*
b**3*c**3*d**4*n**2*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*
n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5
```

```
040*d**7) + 180*D*b**3*c**3*d**4*n*x**4*(c + d*x)**n/(d**7*n**7 + 28*d**7*n
**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 1
3068*d**7*n + 5040*d**7) - 6*D*b**3*c**2*d**5*n**5*x**5*(c + d*x)**n/(d**7*
n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 131
32*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 60*D*b**3*c**2*d**5*n**4*x**5*(c
 + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 676
9*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) - 210*D*b**3*c**2
*d**5*n**3*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 19
60*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7)
 - 300*D*b**3*c**2*d**5*n**2*x**5*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 +
322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d
**7*n + 5040*d**7) - 144*D*b**3*c**2*d**5*n*x**5*(c + d*x)**n/(d**7*n**7 +
28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7
*n**2 + 13068*d**7*n + 5040*d**7) + D*b**3*c*d**6*n**6*x**6*(c + d*x)**n/(d
**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 +
 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 15*D*b**3*c*d**6*n**5*x**6*(
c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 67
69*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 85*D*b**3*c*d*
*6*n**4*x**6*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*
d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) +
225*D*b**3*c*d**6*n**3*x**6*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d*
*7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n
+ 5040*d**7) + 274*D*b**3*c*d**6*n**2*x**6*(c + d*x)**n/(d**7*n**7 + 28*d**
7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2
+ 13068*d**7*n + 5040*d**7) + 120*D*b**3*c*d**6*n*x**6*(c + d*x)**n/(d**7*n
**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 1313
2*d**7*n**2 + 13068*d**7*n + 5040*d**7) + D*b**3*d**7*n**6*x**7*(c + d*x)**
n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n*
*3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 21*D*b**3*d**7*n**5*x**7
*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 +
6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) + 175*D*b**3*d
**7*n**4*x**7*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960
*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7) +
 735*D*b**3*d**7*n**3*x**7*(c + d*x)**n/(d**7*n**7 + 28*d**7*n**6 + 322*d**
7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 + 13068*d**7*n +
 5040*d**7) + 1624*D*b**3*d**7*n**2*x**7*(c + d*x)**n/(d**7*n**7 + 28*d**7*
n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d**7*n**2 +
13068*d**7*n + 5040*d**7) + 1764*D*b**3*d**7*n*x**7*(c + d*x)**n/(d**7*n**7
 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 + 13132*d
**7*n**2 + 13068*d**7*n + 5040*d**7) + 720*D*b**3*d**7*x**7*(c + d*x)**n/(d
**7*n**7 + 28*d**7*n**6 + 322*d**7*n**5 + 1960*d**7*n**4 + 6769*d**7*n**3 +
 13132*d**7*n**2 + 13068*d**7*n + 5040*d**7), True))
```

Maxima [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1802 vs. 2(455) = 910.

Time = 0.27 (sec) , antiderivative size = 1802, normalized size of antiderivative = 3.96

$$\int (a+bx)^3(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx=\text{Too large to display}$$

```
[In] integrate((b*x+a)^3*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="maxima")
```

```
[Out] (d^2*(n + 1)*x^2 + c*d*n*x - c^2)*(d*x + c)^n*B*a^3/((n^2 + 3*n + 2)*d^2) +
 3*(d^2*(n + 1)*x^2 + c*d*n*x - c^2)*(d*x + c)^n*A*a^2*b/((n^2 + 3*n + 2)*d
^2) + (d*x + c)^(n + 1)*A*a^3/(d*(n + 1)) + ((n^2 + 3*n + 2)*d^3*x^3 + (n^2
 + n)*c*d^2*x^2 - 2*c^2*d*n*x + 2*c^3)*(d*x + c)^n*C*a^3/((n^3 + 6*n^2 + 11
*n + 6)*d^3) + 3*((n^2 + 3*n + 2)*d^3*x^3 + (n^2 + n)*c*d^2*x^2 - 2*c^2*d*n
*x + 2*c^3)*(d*x + c)^n*B*a^2*b/((n^3 + 6*n^2 + 11*n + 6)*d^3) + 3*((n^2 +
3*n + 2)*d^3*x^3 + (n^2 + n)*c*d^2*x^2 - 2*c^2*d*n*x + 2*c^3)*(d*x + c)^n*A
*a*b^2/((n^3 + 6*n^2 + 11*n + 6)*d^3) + ((n^3 + 6*n^2 + 11*n + 6)*d^4*x^4 +
 (n^3 + 3*n^2 + 2*n)*c*d^3*x^3 - 3*(n^2 + n)*c^2*d^2*x^2 + 6*c^3*d*n*x - 6*
c^4)*(d*x + c)^n*D*a^3/((n^4 + 10*n^3 + 35*n^2 + 50*n + 24)*d^4) + 3*((n^3
+ 6*n^2 + 11*n + 6)*d^4*x^4 + (n^3 + 3*n^2 + 2*n)*c*d^3*x^3 - 3*(n^2 + n)*c
^2*d^2*x^2 + 6*c^3*d*n*x - 6*c^4)*(d*x + c)^n*C*a^2*b/((n^4 + 10*n^3 + 35*n
^2 + 50*n + 24)*d^4) + 3*((n^3 + 6*n^2 + 11*n + 6)*d^4*x^4 + (n^3 + 3*n^2 +
 2*n)*c*d^3*x^3 - 3*(n^2 + n)*c^2*d^2*x^2 + 6*c^3*d*n*x - 6*c^4)*(d*x + c)^
n*B*a*b^2/((n^4 + 10*n^3 + 35*n^2 + 50*n + 24)*d^4) + ((n^3 + 6*n^2 + 11*n
+ 6)*d^4*x^4 + (n^3 + 3*n^2 + 2*n)*c*d^3*x^3 - 3*(n^2 + n)*c^2*d^2*x^2 + 6*
c^3*d*n*x - 6*c^4)*(d*x + c)^n*A*b^3/((n^4 + 10*n^3 + 35*n^2 + 50*n + 24)*d
^4) + 3*((n^4 + 10*n^3 + 35*n^2 + 50*n + 24)*d^5*x^5 + (n^4 + 6*n^3 + 11*n^
2 + 6*n)*c*d^4*x^4 - 4*(n^3 + 3*n^2 + 2*n)*c^2*d^3*x^3 + 12*(n^2 + n)*c^3*d
^2*x^2 - 24*c^4*d*n*x + 24*c^5)*(d*x + c)^n*D*a^2*b/((n^5 + 15*n^4 + 85*n^3
 + 225*n^2 + 274*n + 120)*d^5) + 3*((n^4 + 10*n^3 + 35*n^2 + 50*n + 24)*d^5
*x^5 + (n^4 + 6*n^3 + 11*n^2 + 6*n)*c*d^4*x^4 - 4*(n^3 + 3*n^2 + 2*n)*c^2*d
^3*x^3 + 12*(n^2 + n)*c^3*d^2*x^2 - 24*c^4*d*n*x + 24*c^5)*(d*x + c)^n*C*a*
b^2/((n^5 + 15*n^4 + 85*n^3 + 225*n^2 + 274*n + 120)*d^5) + ((n^4 + 10*n^3
+ 35*n^2 + 50*n + 24)*d^5*x^5 + (n^4 + 6*n^3 + 11*n^2 + 6*n)*c*d^4*x^4 - 4*
(n^3 + 3*n^2 + 2*n)*c^2*d^3*x^3 + 12*(n^2 + n)*c^3*d^2*x^2 - 24*c^4*d*n*x +
 24*c^5)*(d*x + c)^n*B*b^3/((n^5 + 15*n^4 + 85*n^3 + 225*n^2 + 274*n + 120)
*d^5) + 3*((n^5 + 15*n^4 + 85*n^3 + 225*n^2 + 274*n + 120)*d^6*x^6 + (n^5 +
 10*n^4 + 35*n^3 + 50*n^2 + 24*n)*c*d^5*x^5 - 5*(n^4 + 6*n^3 + 11*n^2 + 6*n
)*c^2*d^4*x^4 + 20*(n^3 + 3*n^2 + 2*n)*c^3*d^3*x^3 - 60*(n^2 + n)*c^4*d^2*x
^2 + 120*c^5*d*n*x - 120*c^6)*(d*x + c)^n*D*a*b^2/((n^6 + 21*n^5 + 175*n^4
+ 735*n^3 + 1624*n^2 + 1764*n + 720)*d^6) + ((n^5 + 15*n^4 + 85*n^3 + 225*n
^2 + 274*n + 120)*d^6*x^6 + (n^5 + 10*n^4 + 35*n^3 + 50*n^2 + 24*n)*c*d^5*x
^5 - 5*(n^4 + 6*n^3 + 11*n^2 + 6*n)*c^2*d^4*x^4 + 20*(n^3 + 3*n^2 + 2*n)*c^
3*d^3*x^3 - 60*(n^2 + n)*c^4*d^2*x^2 + 120*c^5*d*n*x - 120*c^6)*(d*x + c)^n
```

```
*C*b^3/((n^6 + 21*n^5 + 175*n^4 + 735*n^3 + 1624*n^2 + 1764*n + 720)*d^6) +
 ((n^6 + 21*n^5 + 175*n^4 + 735*n^3 + 1624*n^2 + 1764*n + 720)*d^7*x^7 + (n
^6 + 15*n^5 + 85*n^4 + 225*n^3 + 274*n^2 + 120*n)*c*d^6*x^6 - 6*(n^5 + 10*n
^4 + 35*n^3 + 50*n^2 + 24*n)*c^2*d^5*x^5 + 30*(n^4 + 6*n^3 + 11*n^2 + 6*n)*
c^3*d^4*x^4 - 120*(n^3 + 3*n^2 + 2*n)*c^4*d^3*x^3 + 360*(n^2 + n)*c^5*d^2*x
^2 - 720*c^6*d*n*x + 720*c^7)*(d*x + c)^n*D*b^3/((n^7 + 28*n^6 + 322*n^5 +
1960*n^4 + 6769*n^3 + 13132*n^2 + 13068*n + 5040)*d^7)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 9032 vs. $2(455) = 910$.

Time = 0.35 (sec) , antiderivative size = 9032, normalized size of antiderivative = 19.85

$$\int (a+bx)^3(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right)\,dx = \text{Too large to display}$$

```
[In] integrate((b*x+a)^3*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="giac")
```

```
[Out] ((d*x + c)^n*D*b^3*d^7*n^6*x^7 + (d*x + c)^n*D*b^3*c*d^6*n^6*x^6 + 3*(d*x +
 c)^n*D*a*b^2*d^7*n^6*x^6 + (d*x + c)^n*C*b^3*d^7*n^6*x^6 + 21*(d*x + c)^n*
D*b^3*d^7*n^5*x^7 + 3*(d*x + c)^n*D*a*b^2*c*d^6*n^6*x^5 + (d*x + c)^n*C*b^3
*c*d^6*n^6*x^5 + 3*(d*x + c)^n*D*a^2*b*d^7*n^6*x^5 + 3*(d*x + c)^n*C*a*b^2*
d^7*n^6*x^5 + (d*x + c)^n*B*b^3*d^7*n^6*x^5 + 15*(d*x + c)^n*D*b^3*c*d^6*n^
5*x^6 + 66*(d*x + c)^n*D*a*b^2*d^7*n^5*x^6 + 22*(d*x + c)^n*C*b^3*d^7*n^5*x
^6 + 175*(d*x + c)^n*D*b^3*d^7*n^4*x^7 + 3*(d*x + c)^n*D*a^2*b*c*d^6*n^6*x^
4 + 3*(d*x + c)^n*C*a*b^2*c*d^6*n^6*x^4 + (d*x + c)^n*B*b^3*c*d^6*n^6*x^4 +
 (d*x + c)^n*D*a^3*d^7*n^6*x^4 + 3*(d*x + c)^n*C*a^2*b*d^7*n^6*x^4 + 3*(d*x
 + c)^n*B*a*b^2*d^7*n^6*x^4 + (d*x + c)^n*A*b^3*d^7*n^6*x^4 - 6*(d*x + c)^n
*D*b^3*c^2*d^5*n^5*x^5 + 51*(d*x + c)^n*D*a*b^2*c*d^6*n^5*x^5 + 17*(d*x + c
)^n*C*b^3*c*d^6*n^5*x^5 + 69*(d*x + c)^n*D*a^2*b*d^7*n^5*x^5 + 69*(d*x + c)
^n*C*a*b^2*d^7*n^5*x^5 + 23*(d*x + c)^n*B*b^3*d^7*n^5*x^5 + 85*(d*x + c)^n*
D*b^3*c*d^6*n^4*x^6 + 570*(d*x + c)^n*D*a*b^2*d^7*n^4*x^6 + 190*(d*x + c)^n
*C*b^3*d^7*n^4*x^6 + 735*(d*x + c)^n*D*b^3*d^7*n^3*x^7 + (d*x + c)^n*D*a^3*
c*d^6*n^6*x^3 + 3*(d*x + c)^n*C*a^2*b*c*d^6*n^6*x^3 + 3*(d*x + c)^n*B*a*b^2
*c*d^6*n^6*x^3 + (d*x + c)^n*A*b^3*c*d^6*n^6*x^3 + (d*x + c)^n*C*a^3*d^7*n^
6*x^3 + 3*(d*x + c)^n*B*a^2*b*d^7*n^6*x^3 + 3*(d*x + c)^n*A*a*b^2*d^7*n^6*x
^3 - 15*(d*x + c)^n*D*a*b^2*c^2*d^5*n^5*x^4 - 5*(d*x + c)^n*C*b^3*c^2*d^5*n
^5*x^4 + 57*(d*x + c)^n*D*a^2*b*c*d^6*n^5*x^4 + 57*(d*x + c)^n*C*a*b^2*c*d^
6*n^5*x^4 + 19*(d*x + c)^n*B*b^3*c*d^6*n^5*x^4 + 24*(d*x + c)^n*D*a^3*d^7*n
^5*x^4 + 72*(d*x + c)^n*C*a^2*b*d^7*n^5*x^4 + 72*(d*x + c)^n*B*a*b^2*d^7*n^
5*x^4 + 24*(d*x + c)^n*A*b^3*d^7*n^5*x^4 - 60*(d*x + c)^n*D*b^3*c^2*d^5*n^4
*x^5 + 315*(d*x + c)^n*D*a*b^2*c*d^6*n^4*x^5 + 105*(d*x + c)^n*C*b^3*c*d^6*
n^4*x^5 + 621*(d*x + c)^n*D*a^2*b*d^7*n^4*x^5 + 621*(d*x + c)^n*C*a*b^2*d^7
*n^4*x^5 + 207*(d*x + c)^n*B*b^3*d^7*n^4*x^5 + 225*(d*x + c)^n*D*b^3*c*d^6*
n^3*x^6 + 2460*(d*x + c)^n*D*a*b^2*d^7*n^3*x^6 + 820*(d*x + c)^n*C*b^3*d^7*
```

```
n^3*x^6 + 1624*(d*x + c)^n*D*b^3*d^7*n^2*x^7 + (d*x + c)^n*C*a^3*c*d^6*n^6*
x^2 + 3*(d*x + c)^n*B*a^2*b*c*d^6*n^6*x^2 + 3*(d*x + c)^n*A*a*b^2*c*d^6*n^6
*x^2 + (d*x + c)^n*B*a^3*d^7*n^6*x^2 + 3*(d*x + c)^n*A*a^2*b*d^7*n^6*x^2 -
12*(d*x + c)^n*D*a^2*b*c^2*d^5*n^5*x^3 - 12*(d*x + c)^n*C*a*b^2*c^2*d^5*n^5
*x^3 - 4*(d*x + c)^n*B*b^3*c^2*d^5*n^5*x^3 + 21*(d*x + c)^n*D*a^3*c*d^6*n^5
*x^3 + 63*(d*x + c)^n*C*a^2*b*c*d^6*n^5*x^3 + 63*(d*x + c)^n*B*a*b^2*c*d^6*
n^5*x^3 + 21*(d*x + c)^n*A*b^3*c*d^6*n^5*x^3 + 25*(d*x + c)^n*C*a^3*d^7*n^5
*x^3 + 75*(d*x + c)^n*B*a^2*b*d^7*n^5*x^3 + 75*(d*x + c)^n*A*a*b^2*d^7*n^5*
x^3 + 30*(d*x + c)^n*D*b^3*c^3*d^4*n^4*x^4 - 195*(d*x + c)^n*D*a*b^2*c^2*d^
5*n^4*x^4 - 65*(d*x + c)^n*C*b^3*c^2*d^5*n^4*x^4 + 393*(d*x + c)^n*D*a^2*b*
c*d^6*n^4*x^4 + 393*(d*x + c)^n*C*a*b^2*c*d^6*n^4*x^4 + 131*(d*x + c)^n*B*b
^3*c*d^6*n^4*x^4 + 226*(d*x + c)^n*D*a^3*d^7*n^4*x^4 + 678*(d*x + c)^n*C*a^
2*b*d^7*n^4*x^4 + 678*(d*x + c)^n*B*a*b^2*d^7*n^4*x^4 + 226*(d*x + c)^n*A*b
^3*d^7*n^4*x^4 - 210*(d*x + c)^n*D*b^3*c^2*d^5*n^3*x^5 + 885*(d*x + c)^n*D*
a*b^2*c*d^6*n^3*x^5 + 295*(d*x + c)^n*C*b^3*c*d^6*n^3*x^5 + 2775*(d*x + c)^
n*D*a^2*b*d^7*n^3*x^5 + 2775*(d*x + c)^n*C*a*b^2*d^7*n^3*x^5 + 925*(d*x + c
)^n*B*b^3*d^7*n^3*x^5 + 274*(d*x + c)^n*D*b^3*c*d^6*n^2*x^6 + 5547*(d*x + c
)^n*D*a*b^2*d^7*n^2*x^6 + 1849*(d*x + c)^n*C*b^3*d^7*n^2*x^6 + 1764*(d*x +
c)^n*D*b^3*d^7*n*x^7 + (d*x + c)^n*B*a^3*c*d^6*n^6*x + 3*(d*x + c)^n*A*a^2*
b*c*d^6*n^6*x + (d*x + c)^n*A*a^3*d^7*n^6*x - 3*(d*x + c)^n*D*a^3*c^2*d^5*n
^5*x^2 - 9*(d*x + c)^n*C*a^2*b*c^2*d^5*n^5*x^2 - 9*(d*x + c)^n*B*a*b^2*c^2*
d^5*n^5*x^2 - 3*(d*x + c)^n*A*b^3*c^2*d^5*n^5*x^2 + 23*(d*x + c)^n*C*a^3*c*
d^6*n^5*x^2 + 69*(d*x + c)^n*B*a^2*b*c*d^6*n^5*x^2 + 69*(d*x + c)^n*A*a*b^2
*c*d^6*n^5*x^2 + 26*(d*x + c)^n*B*a^3*d^7*n^5*x^2 + 78*(d*x + c)^n*A*a^2*b*
d^7*n^5*x^2 + 60*(d*x + c)^n*D*a*b^2*c^3*d^4*n^4*x^3 + 20*(d*x + c)^n*C*b^3
*c^3*d^4*n^4*x^3 - 192*(d*x + c)^n*D*a^2*b*c^2*d^5*n^4*x^3 - 192*(d*x + c)^
n*C*a*b^2*c^2*d^5*n^4*x^3 - 64*(d*x + c)^n*B*b^3*c^2*d^5*n^4*x^3 + 163*(d*x
 + c)^n*D*a^3*c*d^6*n^4*x^3 + 489*(d*x + c)^n*C*a^2*b*c*d^6*n^4*x^3 + 489*(
d*x + c)^n*B*a*b^2*c*d^6*n^4*x^3 + 163*(d*x + c)^n*A*b^3*c*d^6*n^4*x^3 + 24
7*(d*x + c)^n*C*a^3*d^7*n^4*x^3 + 741*(d*x + c)^n*B*a^2*b*d^7*n^4*x^3 + 741
*(d*x + c)^n*A*a*b^2*d^7*n^4*x^3 + 180*(d*x + c)^n*D*b^3*c^3*d^4*n^3*x^4 -
795*(d*x + c)^n*D*a*b^2*c^2*d^5*n^3*x^4 - 265*(d*x + c)^n*C*b^3*c^2*d^5*n^3
*x^4 + 1203*(d*x + c)^n*D*a^2*b*c*d^6*n^3*x^4 + 1203*(d*x + c)^n*C*a*b^2*c*
d^6*n^3*x^4 + 401*(d*x + c)^n*B*b^3*c*d^6*n^3*x^4 + 1056*(d*x + c)^n*D*a^3*
d^7*n^3*x^4 + 3168*(d*x + c)^n*C*a^2*b*d^7*n^3*x^4 + 3168*(d*x + c)^n*B*a*b
^2*d^7*n^3*x^4 + 1056*(d*x + c)^n*A*b^3*d^7*n^3*x^4 - 300*(d*x + c)^n*D*b^3
*c^2*d^5*n^2*x^5 + 1122*(d*x + c)^n*D*a*b^2*c*d^6*n^2*x^5 + 374*(d*x + c)^n
*C*b^3*c*d^6*n^2*x^5 + 6432*(d*x + c)^n*D*a^2*b*d^7*n^2*x^5 + 6432*(d*x + c
)^n*C*a*b^2*d^7*n^2*x^5 + 2144*(d*x + c)^n*B*b^3*d^7*n^2*x^5 + 120*(d*x + c
)^n*D*b^3*c*d^6*n*x^6 + 6114*(d*x + c)^n*D*a*b^2*d^7*n*x^6 + 2038*(d*x + c)
^n*C*b^3*d^7*n*x^6 + 720*(d*x + c)^n*D*b^3*d^7*x^7 + (d*x + c)^n*A*a^3*c*d^
6*n^6 - 2*(d*x + c)^n*C*a^3*c^2*d^5*n^5*x - 6*(d*x + c)^n*B*a^2*b*c^2*d^5*n
^5*x - 6*(d*x + c)^n*A*a*b^2*c^2*d^5*n^5*x + 25*(d*x + c)^n*B*a^3*c*d^6*n^5
*x + 75*(d*x + c)^n*A*a^2*b*c*d^6*n^5*x + 27*(d*x + c)^n*A*a^3*d^7*n^5*x +
36*(d*x + c)^n*D*a^2*b*c^3*d^4*n^4*x^2 + 36*(d*x + c)^n*C*a*b^2*c^3*d^4*n^4
```

```
*x^2 + 12*(d*x + c)^n*B*b^3*c^3*d^4*n^4*x^2 - 57*(d*x + c)^n*D*a^3*c^2*d^5*
n^4*x^2 - 171*(d*x + c)^n*C*a^2*b*c^2*d^5*n^4*x^2 - 171*(d*x + c)^n*B*a*b^2
*c^2*d^5*n^4*x^2 - 57*(d*x + c)^n*A*b^3*c^2*d^5*n^4*x^2 + 201*(d*x + c)^n*C
*a^3*c*d^6*n^4*x^2 + 603*(d*x + c)^n*B*a^2*b*c*d^6*n^4*x^2 + 603*(d*x + c)^
n*A*a*b^2*c*d^6*n^4*x^2 + 270*(d*x + c)^n*B*a^3*d^7*n^4*x^2 + 810*(d*x + c)
^n*A*a^2*b*d^7*n^4*x^2 - 120*(d*x + c)^n*D*b^3*c^4*d^3*n^3*x^3 + 600*(d*x +
 c)^n*D*a*b^2*c^3*d^4*n^3*x^3 + 200*(d*x + c)^n*C*b^3*c^3*d^4*n^3*x^3 - 996
*(d*x + c)^n*D*a^2*b*c^2*d^5*n^3*x^3 - 996*(d*x + c)^n*C*a*b^2*c^2*d^5*n^3*
x^3 - 332*(d*x + c)^n*B*b^3*c^2*d^5*n^3*x^3 + 567*(d*x + c)^n*D*a^3*c*d^6*n
^3*x^3 + 1701*(d*x + c)^n*C*a^2*b*c*d^6*n^3*x^3 + 1701*(d*x + c)^n*B*a*b^2*
c*d^6*n^3*x^3 + 567*(d*x + c)^n*A*b^3*c*d^6*n^3*x^3 + 1219*(d*x + c)^n*C*a^
3*d^7*n^3*x^3 + 3657*(d*x + c)^n*B*a^2*b*d^7*n^3*x^3 + 3657*(d*x + c)^n*A*a
*b^2*d^7*n^3*x^3 + 330*(d*x + c)^n*D*b^3*c^3*d^4*n^2*x^4 - 1245*(d*x + c)^n
*D*a*b^2*c^2*d^5*n^2*x^4 - 415*(d*x + c)^n*C*b^3*c^2*d^5*n^2*x^4 + 1620*(d*
x + c)^n*D*a^2*b*c*d^6*n^2*x^4 + 1620*(d*x + c)^n*C*a*b^2*c*d^6*n^2*x^4 + 5
40*(d*x + c)^n*B*b^3*c*d^6*n^2*x^4 + 2545*(d*x + c)^n*D*a^3*d^7*n^2*x^4 + 7
635*(d*x + c)^n*C*a^2*b*d^7*n^2*x^4 + 7635*(d*x + c)^n*B*a*b^2*d^7*n^2*x^4
 + 2545*(d*x + c)^n*A*b^3*d^7*n^2*x^4 - 144*(d*x + c)^n*D*b^3*c^2*d^5*n*x^5
 + 504*(d*x + c)^n*D*a*b^2*c*d^6*n*x^5 + 168*(d*x + c)^n*C*b^3*c*d^6*n*x^5 +
 7236*(d*x + c)^n*D*a^2*b*d^7*n*x^5 + 7236*(d*x + c)^n*C*a*b^2*d^7*n*x^5 +
2412*(d*x + c)^n*B*b^3*d^7*n*x^5 + 2520*(d*x + c)^n*D*a*b^2*d^7*x^6 + 840*(
d*x + c)^n*C*b^3*d^7*x^6 - (d*x + c)^n*B*a^3*c^2*d^5*n^5 - 3*(d*x + c)^n*A*
a^2*b*c^2*d^5*n^5 + 27*(d*x + c)^n*A*a^3*c*d^6*n^5 + 6*(d*x + c)^n*D*a^3*c^
3*d^4*n^4*x + 18*(d*x + c)^n*C*a^2*b*c^3*d^4*n^4*x + 18*(d*x + c)^n*B*a*b^2
*c^3*d^4*n^4*x + 6*(d*x + c)^n*A*b^3*c^3*d^4*n^4*x - 44*(d*x + c)^n*C*a^3*c
^2*d^5*n^4*x - 132*(d*x + c)^n*B*a^2*b*c^2*d^5*n^4*x - 132*(d*x + c)^n*A*a*
b^2*c^2*d^5*n^4*x + 245*(d*x + c)^n*B*a^3*c*d^6*n^4*x + 735*(d*x + c)^n*A*a
^2*b*c*d^6*n^4*x + 295*(d*x + c)^n*A*a^3*d^7*n^4*x - 180*(d*x + c)^n*D*a*b^
2*c^4*d^3*n^3*x^2 - 60*(d*x + c)^n*C*b^3*c^4*d^3*n^3*x^2 + 504*(d*x + c)^n*
D*a^2*b*c^3*d^4*n^3*x^2 + 504*(d*x + c)^n*C*a*b^2*c^3*d^4*n^3*x^2 + 168*(d*
x + c)^n*B*b^3*c^3*d^4*n^3*x^2 - 375*(d*x + c)^n*D*a^3*c^2*d^5*n^3*x^2 - 11
25*(d*x + c)^n*C*a^2*b*c^2*d^5*n^3*x^2 - 1125*(d*x + c)^n*B*a*b^2*c^2*d^5*n
^3*x^2 - 375*(d*x + c)^n*A*b^3*c^2*d^5*n^3*x^2 + 817*(d*x + c)^n*C*a^3*c*d^
6*n^3*x^2 + 2451*(d*x + c)^n*B*a^2*b*c*d^6*n^3*x^2 + 2451*(d*x + c)^n*A*a*b
^2*c*d^6*n^3*x^2 + 1420*(d*x + c)^n*B*a^3*d^7*n^3*x^2 + 4260*(d*x + c)^n*A*
a^2*b*d^7*n^3*x^2 - 360*(d*x + c)^n*D*b^3*c^4*d^3*n^2*x^3 + 1380*(d*x + c)^
n*D*a*b^2*c^3*d^4*n^2*x^3 + 460*(d*x + c)^n*C*b^3*c^3*d^4*n^2*x^3 - 1824*(d
*x + c)^n*D*a^2*b*c^2*d^5*n^2*x^3 - 1824*(d*x + c)^n*C*a*b^2*c^2*d^5*n^2*x^
3 - 608*(d*x + c)^n*B*b^3*c^2*d^5*n^2*x^3 + 844*(d*x + c)^n*D*a^3*c*d^6*n^2
*x^3 + 2532*(d*x + c)^n*C*a^2*b*c*d^6*n^2*x^3 + 2532*(d*x + c)^n*B*a*b^2*c*
d^6*n^2*x^3 + 844*(d*x + c)^n*A*b^3*c*d^6*n^2*x^3 + 3112*(d*x + c)^n*C*a^3*
d^7*n^2*x^3 + 9336*(d*x + c)^n*B*a^2*b*d^7*n^2*x^3 + 9336*(d*x + c)^n*A*a*b
^2*d^7*n^2*x^3 + 180*(d*x + c)^n*D*b^3*c^3*d^4*n*x^4 - 630*(d*x + c)^n*D*a*
b^2*c^2*d^5*n*x^4 - 210*(d*x + c)^n*C*b^3*c^2*d^5*n*x^4 + 756*(d*x + c)^n*D
*a^2*b*c*d^6*n*x^4 + 756*(d*x + c)^n*C*a*b^2*c*d^6*n*x^4 + 252*(d*x + c)^n*
```

```
B*b^3*c*d^6*n*x^4 + 2952*(d*x + c)^n*D*a^3*d^7*n*x^4 + 8856*(d*x + c)^n*C*a
^2*b*d^7*n*x^4 + 8856*(d*x + c)^n*B*a*b^2*d^7*n*x^4 + 2952*(d*x + c)^n*A*b^
3*d^7*n*x^4 + 3024*(d*x + c)^n*D*a^2*b*d^7*x^5 + 3024*(d*x + c)^n*C*a*b^2*d
^7*x^5 + 1008*(d*x + c)^n*B*b^3*d^7*x^5 + 2*(d*x + c)^n*C*a^3*c^3*d^4*n^4 +
 6*(d*x + c)^n*B*a^2*b*c^3*d^4*n^4 + 6*(d*x + c)^n*A*a*b^2*c^3*d^4*n^4 - 25
*(d*x + c)^n*B*a^3*c^2*d^5*n^4 - 75*(d*x + c)^n*A*a^2*b*c^2*d^5*n^4 + 295*(
d*x + c)^n*A*a^3*c*d^6*n^4 - 72*(d*x + c)^n*D*a^2*b*c^4*d^3*n^3*x - 72*(d*x
 + c)^n*C*a*b^2*c^4*d^3*n^3*x - 24*(d*x + c)^n*B*b^3*c^4*d^3*n^3*x + 108*(d
*x + c)^n*D*a^3*c^3*d^4*n^3*x + 324*(d*x + c)^n*C*a^2*b*c^3*d^4*n^3*x + 324
*(d*x + c)^n*B*a*b^2*c^3*d^4*n^3*x + 108*(d*x + c)^n*A*b^3*c^3*d^4*n^3*x -
358*(d*x + c)^n*C*a^3*c^2*d^5*n^3*x - 1074*(d*x + c)^n*B*a^2*b*c^2*d^5*n^3*
x - 1074*(d*x + c)^n*A*a*b^2*c^2*d^5*n^3*x + 1175*(d*x + c)^n*B*a^3*c*d^6*n
^3*x + 3525*(d*x + c)^n*A*a^2*b*c*d^6*n^3*x + 1665*(d*x + c)^n*A*a^3*d^7*n^
3*x + 360*(d*x + c)^n*D*b^3*c^5*d^2*n^2*x^2 - 1440*(d*x + c)^n*D*a*b^2*c^4*
d^3*n^2*x^2 - 480*(d*x + c)^n*C*b^3*c^4*d^3*n^2*x^2 + 1980*(d*x + c)^n*D*a^
2*b*c^3*d^4*n^2*x^2 + 1980*(d*x + c)^n*C*a*b^2*c^3*d^4*n^2*x^2 + 660*(d*x +
 c)^n*B*b^3*c^3*d^4*n^2*x^2 - 951*(d*x + c)^n*D*a^3*c^2*d^5*n^2*x^2 - 2853*
(d*x + c)^n*C*a^2*b*c^2*d^5*n^2*x^2 - 2853*(d*x + c)^n*B*a*b^2*c^2*d^5*n^2*
x^2 - 951*(d*x + c)^n*A*b^3*c^2*d^5*n^2*x^2 + 1478*(d*x + c)^n*C*a^3*c*d^6*
n^2*x^2 + 4434*(d*x + c)^n*B*a^2*b*c*d^6*n^2*x^2 + 4434*(d*x + c)^n*A*a*b^2
*c*d^6*n^2*x^2 + 3929*(d*x + c)^n*B*a^3*d^7*n^2*x^2 + 11787*(d*x + c)^n*A*a
^2*b*d^7*n^2*x^2 - 240*(d*x + c)^n*D*b^3*c^4*d^3*n*x^3 + 840*(d*x + c)^n*D*
a*b^2*c^3*d^4*n*x^3 + 280*(d*x + c)^n*C*b^3*c^3*d^4*n*x^3 - 1008*(d*x + c)^
n*D*a^2*b*c^2*d^5*n*x^3 - 1008*(d*x + c)^n*C*a*b^2*c^2*d^5*n*x^3 - 336*(d*x
 + c)^n*B*b^3*c^2*d^5*n*x^3 + 420*(d*x + c)^n*D*a^3*c*d^6*n*x^3 + 1260*(d*x
 + c)^n*C*a^2*b*c*d^6*n*x^3 + 1260*(d*x + c)^n*B*a*b^2*c*d^6*n*x^3 + 420*(d
*x + c)^n*A*b^3*c*d^6*n*x^3 + 3796*(d*x + c)^n*C*a^3*d^7*n*x^3 + 11388*(d*x
 + c)^n*B*a^2*b*d^7*n*x^3 + 11388*(d*x + c)^n*A*a*b^2*d^7*n*x^3 + 1260*(d*x
 + c)^n*D*a^3*d^7*x^4 + 3780*(d*x + c)^n*C*a^2*b*d^7*x^4 + 3780*(d*x + c)^n
*B*a*b^2*d^7*x^4 + 1260*(d*x + c)^n*A*b^3*d^7*x^4 - 6*(d*x + c)^n*D*a^3*c^4
*d^3*n^3 - 18*(d*x + c)^n*C*a^2*b*c^4*d^3*n^3 - 18*(d*x + c)^n*B*a*b^2*c^4*
d^3*n^3 - 6*(d*x + c)^n*A*b^3*c^4*d^3*n^3 + 44*(d*x + c)^n*C*a^3*c^3*d^4*n^
3 + 132*(d*x + c)^n*B*a^2*b*c^3*d^4*n^3 + 132*(d*x + c)^n*A*a*b^2*c^3*d^4*n
^3 - 245*(d*x + c)^n*B*a^3*c^2*d^5*n^3 - 735*(d*x + c)^n*A*a^2*b*c^2*d^5*n^
3 + 1665*(d*x + c)^n*A*a^3*c*d^6*n^3 + 360*(d*x + c)^n*D*a*b^2*c^5*d^2*n^2*
x + 120*(d*x + c)^n*C*b^3*c^5*d^2*n^2*x - 936*(d*x + c)^n*D*a^2*b*c^4*d^3*n
^2*x - 936*(d*x + c)^n*C*a*b^2*c^4*d^3*n^2*x - 312*(d*x + c)^n*B*b^3*c^4*d^
3*n^2*x + 642*(d*x + c)^n*D*a^3*c^3*d^4*n^2*x + 1926*(d*x + c)^n*C*a^2*b*c^
3*d^4*n^2*x + 1926*(d*x + c)^n*B*a*b^2*c^3*d^4*n^2*x + 642*(d*x + c)^n*A*b^
3*c^3*d^4*n^2*x - 1276*(d*x + c)^n*C*a^3*c^2*d^5*n^2*x - 3828*(d*x + c)^n*B
*a^2*b*c^2*d^5*n^2*x - 3828*(d*x + c)^n*A*a*b^2*c^2*d^5*n^2*x + 2754*(d*x +
 c)^n*B*a^3*c*d^6*n^2*x + 8262*(d*x + c)^n*A*a^2*b*c*d^6*n^2*x + 5104*(d*x
+ c)^n*A*a^3*d^7*n^2*x + 360*(d*x + c)^n*D*b^3*c^5*d^2*n*x^2 - 1260*(d*x +
c)^n*D*a*b^2*c^4*d^3*n*x^2 - 420*(d*x + c)^n*C*b^3*c^4*d^3*n*x^2 + 1512*(d*
x + c)^n*D*a^2*b*c^3*d^4*n*x^2 + 1512*(d*x + c)^n*C*a*b^2*c^3*d^4*n*x^2 + 5
```

```
04*(d*x + c)^n*B*b^3*c^3*d^4*n*x^2 - 630*(d*x + c)^n*D*a^3*c^2*d^5*n*x^2 -
1890*(d*x + c)^n*C*a^2*b*c^2*d^5*n*x^2 - 1890*(d*x + c)^n*B*a*b^2*c^2*d^5*n
*x^2 - 630*(d*x + c)^n*A*b^3*c^2*d^5*n*x^2 + 840*(d*x + c)^n*C*a^3*c*d^6*n*
x^2 + 2520*(d*x + c)^n*B*a^2*b*c*d^6*n*x^2 + 2520*(d*x + c)^n*A*a*b^2*c*d^6
*n*x^2 + 5274*(d*x + c)^n*B*a^3*d^7*n*x^2 + 15822*(d*x + c)^n*A*a^2*b*d^7*n
*x^2 + 1680*(d*x + c)^n*C*a^3*d^7*x^3 + 5040*(d*x + c)^n*B*a^2*b*d^7*x^3 +
5040*(d*x + c)^n*A*a*b^2*d^7*x^3 + 72*(d*x + c)^n*D*a^2*b*c^5*d^2*n^2 + 72*
(d*x + c)^n*C*a*b^2*c^5*d^2*n^2 + 24*(d*x + c)^n*B*b^3*c^5*d^2*n^2 - 108*(d
*x + c)^n*D*a^3*c^4*d^3*n^2 - 324*(d*x + c)^n*C*a^2*b*c^4*d^3*n^2 - 324*(d*
x + c)^n*B*a*b^2*c^4*d^3*n^2 - 108*(d*x + c)^n*A*b^3*c^4*d^3*n^2 + 358*(d*x
 + c)^n*C*a^3*c^3*d^4*n^2 + 1074*(d*x + c)^n*B*a^2*b*c^3*d^4*n^2 + 1074*(d*
x + c)^n*A*a*b^2*c^3*d^4*n^2 - 1175*(d*x + c)^n*B*a^3*c^2*d^5*n^2 - 3525*(d
*x + c)^n*A*a^2*b*c^2*d^5*n^2 + 5104*(d*x + c)^n*A*a^3*c*d^6*n^2 - 720*(d*x
 + c)^n*D*b^3*c^6*d*n*x + 2520*(d*x + c)^n*D*a*b^2*c^5*d^2*n*x + 840*(d*x +
 c)^n*C*b^3*c^5*d^2*n*x - 3024*(d*x + c)^n*D*a^2*b*c^4*d^3*n*x - 3024*(d*x
+ c)^n*C*a*b^2*c^4*d^3*n*x - 1008*(d*x + c)^n*B*b^3*c^4*d^3*n*x + 1260*(d*x
 + c)^n*D*a^3*c^3*d^4*n*x + 3780*(d*x + c)^n*C*a^2*b*c^3*d^4*n*x + 3780*(d*
x + c)^n*B*a*b^2*c^3*d^4*n*x + 1260*(d*x + c)^n*A*b^3*c^3*d^4*n*x - 1680*(d
*x + c)^n*C*a^3*c^2*d^5*n*x - 5040*(d*x + c)^n*B*a^2*b*c^2*d^5*n*x - 5040*(
d*x + c)^n*A*a*b^2*c^2*d^5*n*x + 2520*(d*x + c)^n*B*a^3*c*d^6*n*x + 7560*(d
*x + c)^n*A*a^2*b*c*d^6*n*x + 8028*(d*x + c)^n*A*a^3*d^7*n*x + 2520*(d*x +
c)^n*B*a^3*d^7*x^2 + 7560*(d*x + c)^n*A*a^2*b*d^7*x^2 - 360*(d*x + c)^n*D*a
*b^2*c^6*d*n - 120*(d*x + c)^n*C*b^3*c^6*d*n + 936*(d*x + c)^n*D*a^2*b*c^5*
d^2*n + 936*(d*x + c)^n*C*a*b^2*c^5*d^2*n + 312*(d*x + c)^n*B*b^3*c^5*d^2*n
 - 642*(d*x + c)^n*D*a^3*c^4*d^3*n - 1926*(d*x + c)^n*C*a^2*b*c^4*d^3*n - 1
926*(d*x + c)^n*B*a*b^2*c^4*d^3*n - 642*(d*x + c)^n*A*b^3*c^4*d^3*n + 1276*
(d*x + c)^n*C*a^3*c^3*d^4*n + 3828*(d*x + c)^n*B*a^2*b*c^3*d^4*n + 3828*(d*
x + c)^n*A*a*b^2*c^3*d^4*n - 2754*(d*x + c)^n*B*a^3*c^2*d^5*n - 8262*(d*x +
 c)^n*A*a^2*b*c^2*d^5*n + 8028*(d*x + c)^n*A*a^3*c*d^6*n + 5040*(d*x + c)^n
*A*a^3*d^7*x + 720*(d*x + c)^n*D*b^3*c^7 - 2520*(d*x + c)^n*D*a*b^2*c^6*d -
 840*(d*x + c)^n*C*b^3*c^6*d + 3024*(d*x + c)^n*D*a^2*b*c^5*d^2 + 3024*(d*x
 + c)^n*C*a*b^2*c^5*d^2 + 1008*(d*x + c)^n*B*b^3*c^5*d^2 - 1260*(d*x + c)^n
*D*a^3*c^4*d^3 - 3780*(d*x + c)^n*C*a^2*b*c^4*d^3 - 3780*(d*x + c)^n*B*a*b^
2*c^4*d^3 - 1260*(d*x + c)^n*A*b^3*c^4*d^3 + 1680*(d*x + c)^n*C*a^3*c^3*d^4
 + 5040*(d*x + c)^n*B*a^2*b*c^3*d^4 + 5040*(d*x + c)^n*A*a*b^2*c^3*d^4 - 25
20*(d*x + c)^n*B*a^3*c^2*d^5 - 7560*(d*x + c)^n*A*a^2*b*c^2*d^5 + 5040*(d*x
 + c)^n*A*a^3*c*d^6)/(d^7*n^7 + 28*d^7*n^6 + 322*d^7*n^5 + 1960*d^7*n^4 + 6
769*d^7*n^3 + 13132*d^7*n^2 + 13068*d^7*n + 5040*d^7)
```

Mupad [F(-1)]

Timed out.

$$\int (a+bx)^3(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) \, dx \\ = \int {\left(a+b\,x\right)}^3\,{\left(c+d\,x\right)}^n\,\left(A+B\,x+C\,x^2+x^3\,D\right) \,dx$$

```
[In] int((a + b*x)^3*(c + d*x)^n*(A + B*x + C*x^2 + x^3*D),x)
[Out] int((a + b*x)^3*(c + d*x)^n*(A + B*x + C*x^2 + x^3*D), x)
```

3.26 $\int (a+bx)^2(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) \, dx$

Optimal result

Integrand size = 30, antiderivative size = 338

$$\begin{aligned}
&\int (a+bx)^2(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) \, dx \\
&= \frac{(bc-ad)^2 \left(c^2Cd-Bcd^2+Ad^3-c^3D\right) (c+dx)^{1+n}}{d^6(1+n)} \\
&\quad + \frac{(bc-ad) \left(ad(2cCd-Bd^2-3c^2D)-b(4c^2Cd-3Bcd^2+2Ad^3-5c^3D)\right) (c+dx)^{2+n}}{d^6(2+n)} \\
&\quad + \frac{\left(a^2d^2(Cd-3cD)-2abd(3cCd-Bd^2-6c^2D)+b^2(6c^2Cd-3Bcd^2+Ad^3-10c^3D)\right) (c+dx)^{3+n}}{d^6(3+n)} \\
&\quad + \frac{\left(a^2d^2D+2abd(Cd-4cD)-b^2(4cCd-Bd^2-10c^2D)\right) (c+dx)^{4+n}}{d^6(4+n)} \\
&\quad + \frac{b(bCd-5bcD+2adD)(c+dx)^{5+n}}{d^6(5+n)} + \frac{b^2D(c+dx)^{6+n}}{d^6(6+n)}
\end{aligned}$$

```
[Out] (-a*d+b*c)^2*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)*(d*x+c)^(1+n)/d^6/(1+n)+(-a*d+b*
c)*(a*d*(-B*d^2+2*C*c*d-3*D*c^2)-b*(2*A*d^3-3*B*c*d^2+4*C*c^2*d-5*D*c^3))*(
d*x+c)^(2+n)/d^6/(2+n)+(a^2*d^2*(C*d-3*D*c)-2*a*b*d*(-B*d^2+3*C*c*d-6*D*c^2
)+b^2*(A*d^3-3*B*c*d^2+6*C*c^2*d-10*D*c^3))*(d*x+c)^(3+n)/d^6/(3+n)+(a^2*d^
2*D+2*a*b*d*(C*d-4*D*c)-b^2*(-B*d^2+4*C*c*d-10*D*c^2))*(d*x+c)^(4+n)/d^6/(4
+n)+b*(C*b*d+2*D*a*d-5*D*b*c)*(d*x+c)^(5+n)/d^6/(5+n)+b^2*D*(d*x+c)^(6+n)/d
^6/(6+n)
```

Rubi [A] (verified)

Time = 0.19 (sec) , antiderivative size = 338, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.033, Rules used = {1634}

$$\begin{aligned}
&\int (a+bx)^2(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right)\,dx \\
&= \frac{(c+dx)^{n+3}\left(a^2d^2(Cd-3cD)-2abd(-Bd^2-6c^2D+3cCd)+b^2(Ad^3-3Bcd^2-10c^3D+6c^2Cd)\right)}{d^6(n+3)} \\
&\quad+\frac{(c+dx)^{n+4}\left(a^2d^2D+2abd(Cd-4cD)-(b^2(-Bd^2-10c^2D+4cCd))\right)}{d^6(n+4)} \\
&\quad+\frac{(bc-ad)^2(c+dx)^{n+1}\left(Ad^3-Bcd^2+c^3(-D)+c^2Cd\right)}{d^6(n+1)} \\
&\quad+\frac{(bc-ad)(c+dx)^{n+2}\left(ad(-Bd^2-3c^2D+2cCd)-b(2Ad^3-3Bcd^2-5c^3D+4c^2Cd)\right)}{d^6(n+2)} \\
&\quad+\frac{b(c+dx)^{n+5}(2adD-5bcD+bCd)}{d^6(n+5)}+\frac{b^2D(c+dx)^{n+6}}{d^6(n+6)}
\end{aligned}$$

```
[In] Int[(a + b*x)^2*(c + d*x)^n*(A + B*x + C*x^2 + D*x^3),x]
```

```
[Out] ((b*c - a*d)^2*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D)*(c + d*x)^(1 + n))/(d^6*
(1 + n)) + ((b*c - a*d)*(a*d*(2*c*C*d - B*d^2 - 3*c^2*D) - b*(4*c^2*C*d - 3
*B*c*d^2 + 2*A*d^3 - 5*c^3*D))*(c + d*x)^(2 + n))/(d^6*(2 + n)) + ((a^2*d^2
*(C*d - 3*c*D) - 2*a*b*d*(3*c*C*d - B*d^2 - 6*c^2*D) + b^2*(6*c^2*C*d - 3*B
*c*d^2 + A*d^3 - 10*c^3*D))*(c + d*x)^(3 + n))/(d^6*(3 + n)) + ((a^2*d^2*D
+ 2*a*b*d*(C*d - 4*c*D) - b^2*(4*c*C*d - B*d^2 - 10*c^2*D))*(c + d*x)^(4 +
n))/(d^6*(4 + n)) + (b*(b*C*d - 5*b*c*D + 2*a*d*D)*(c + d*x)^(5 + n))/(d^6*
(5 + n)) + (b^2*D*(c + d*x)^(6 + n))/(d^6*(6 + n))
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

$$\text{integral} = \int \left(\frac{(-bc+ad)^2\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)(c+dx)^n}{d^5} + \frac{(bc-ad)\left(ad(2cCd-Bd^2-3c^2D)-b(4c^2Cd-3Bcd^2+2Ad^3-5c^3D)\right)(c+dx)^{1+n}}{d^5} + \frac{\left(a^2d^2(Cd-3cD)-2abd(3cCd-Bd^2-6c^2D)+b^2(6c^2Cd-3Bcd^2+Ad^3-10c^3D)\right)(c+dx)^{2+n}}{d^5} + \frac{\left(a^2d^2D+2abd(Cd-4cD)-b^2(4cCd-Bd^2-10c^2D)\right)(c+dx)^{3+n}}{d^5} + \frac{b(bCd-5bcD+2adD)(c+dx)^{4+n}}{d^5} + \frac{b^2D(c+dx)^{5+n}}{d^5} \right) dx$$

$$= \frac{(bc-ad)^2\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)(c+dx)^{1+n}}{d^6(1+n)} + \frac{(bc-ad)\left(ad(2cCd-Bd^2-3c^2D)-b(4c^2Cd-3Bcd^2+2Ad^3-5c^3D)\right)(c+dx)^{2+n}}{d^6(2+n)} + \frac{\left(a^2d^2(Cd-3cD)-2abd(3cCd-Bd^2-6c^2D)+b^2(6c^2Cd-3Bcd^2+Ad^3-10c^3D)\right)(c+dx)^{3}}{d^6(3+n)} + \frac{\left(a^2d^2D+2abd(Cd-4cD)-b^2(4cCd-Bd^2-10c^2D)\right)(c+dx)^{4+n}}{d^6(4+n)} + \frac{b(bCd-5bcD+2adD)(c+dx)^{5+n}}{d^6(5+n)} + \frac{b^2D(c+dx)^{6+n}}{d^6(6+n)}$$

Mathematica [A] (verified)

Time = 0.77 (sec) , antiderivative size = 308, normalized size of antiderivative = 0.91

$$\int (a+bx)^2(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx$$

$$= \frac{(c+dx)^{1+n}\left(\frac{(bc-ad)^2\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)}{1+n} + \frac{(bc-ad)\left(-ad\left(-2cCd+Bd^2+3c^2D\right)+b\left(-4c^2Cd+3Bcd^2-2Ad^3+5c^3D\right)\right)(c+dx)}{2+n} + (a\right.}{}$$

```
[In] Integrate[(a + b*x)^2*(c + d*x)^n*(A + B*x + C*x^2 + D*x^3),x]
```

```
[Out] ((c + d*x)^(1 + n)*(((b*c - a*d)^2*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D))/(1
+ n) + ((b*c - a*d)*(-(a*d*(-2*c*C*d + B*d^2 + 3*c^2*D)) + b*(-4*c^2*C*d +
3*B*c*d^2 - 2*A*d^3 + 5*c^3*D))*(c + d*x))/(2 + n) + ((a^2*d^2*(C*d - 3*c*D
) + 2*a*b*d*(-3*c*C*d + B*d^2 + 6*c^2*D) + b^2*(6*c^2*C*d - 3*B*c*d^2 + A*d
^3 - 10*c^3*D))*(c + d*x)^2)/(3 + n) + ((a^2*d^2*D + 2*a*b*d*(C*d - 4*c*D)
+ b^2*(-4*c*C*d + B*d^2 + 10*c^2*D))*(c + d*x)^3)/(4 + n) + (b*(b*C*d - 5*b
*c*D + 2*a*d*D)*(c + d*x)^4)/(5 + n) + (b^2*D*(c + d*x)^5)/(6 + n)))/d^6
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 2174 vs. 2(338) = 676.

Time = 1.69 (sec) , antiderivative size = 2175, normalized size of antiderivative = 6.43

method	result	size
norman	Expression too large to display	2175
gosper	Expression too large to display	2588
parallelrisch	Expression too large to display	5150

```
[In] int((b*x+a)^2*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x,method=_RETURNVERBOSE)
```

```
[Out] D*b^2/(6+n)*x^6*exp(n*ln(d*x+c))+c*(A*a^2*d^5*n^5+20*A*a^2*d^5*n^4-2*A*a*b*
c*d^4*n^4-B*a^2*c*d^4*n^4+155*A*a^2*d^5*n^3-36*A*a*b*c*d^4*n^3+2*A*b^2*c^2*
d^3*n^3-18*B*a^2*c*d^4*n^3+4*B*a*b*c^2*d^3*n^3+2*C*a^2*c^2*d^3*n^3+580*A*a^
2*d^5*n^2-238*A*a*b*c*d^4*n^2+30*A*b^2*c^2*d^3*n^2-119*B*a^2*c*d^4*n^2+60*B
*a*b*c^2*d^3*n^2-6*B*b^2*c^3*d^2*n^2+30*C*a^2*c^2*d^3*n^2-12*C*a*b*c^3*d^2*
n^2-6*D*a^2*c^3*d^2*n^2+1044*A*a^2*d^5*n-684*A*a*b*c*d^4*n+148*A*b^2*c^2*d^
3*n-342*B*a^2*c*d^4*n+296*B*a*b*c^2*d^3*n-66*B*b^2*c^3*d^2*n+148*C*a^2*c^2*
d^3*n-132*C*a*b*c^3*d^2*n+24*C*b^2*c^4*d*n-66*D*a^2*c^3*d^2*n+48*D*a*b*c^4*
d*n+720*A*a^2*d^5-720*A*a*b*c*d^4+240*A*b^2*c^2*d^3-360*B*a^2*c*d^4+480*B*a
*b*c^2*d^3-180*B*b^2*c^3*d^2+240*C*a^2*c^2*d^3-360*C*a*b*c^3*d^2+144*C*b^2*
c^4*d-180*D*a^2*c^3*d^2+288*D*a*b*c^4*d-120*D*b^2*c^5)/d^6/(n^6+21*n^5+175*
n^4+735*n^3+1624*n^2+1764*n+720)*exp(n*ln(d*x+c))+(B*b^2*d^2*n^2+2*C*a*b*d^
2*n^2+C*b^2*c*d*n^2+D*a^2*d^2*n^2+2*D*a*b*c*d*n^2+11*B*b^2*d^2*n+22*C*a*b*d
^2*n+6*C*b^2*c*d*n+11*D*a^2*d^2*n+12*D*a*b*c*d*n-5*D*b^2*c^2*n+30*B*b^2*d^2
+60*C*a*b*d^2+30*D*a^2*d^2)/d^2/(n^3+15*n^2+74*n+120)*x^4*exp(n*ln(d*x+c))+
(A*b^2*d^3*n^3+2*B*a*b*d^3*n^3+B*b^2*c*d^2*n^3+C*a^2*d^3*n^3+2*C*a*b*c*d^2*
n^3+D*a^2*c*d^2*n^3+15*A*b^2*d^3*n^2+30*B*a*b*d^3*n^2+11*B*b^2*c*d^2*n^2+15
*C*a^2*d^3*n^2+22*C*a*b*c*d^2*n^2-4*C*b^2*c^2*d*n^2+11*D*a^2*c*d^2*n^2-8*D*
a*b*c^2*d*n^2+74*A*b^2*d^3*n+148*B*a*b*d^3*n+30*B*b^2*c*d^2*n+74*C*a^2*d^3*
n+60*C*a*b*c*d^2*n-24*C*b^2*c^2*d*n+30*D*a^2*c*d^2*n-48*D*a*b*c^2*d*n+20*D*
b^2*c^3*n+120*A*b^2*d^3+240*B*a*b*d^3+120*C*a^2*d^3)/d^3/(n^4+18*n^3+119*n^
2+342*n+360)*x^3*exp(n*ln(d*x+c))+(2*A*a*b*d^4*n^4+A*b^2*c*d^3*n^4+B*a^2*d^
4*n^4+2*B*a*b*c*d^3*n^4+C*a^2*c*d^3*n^4+36*A*a*b*d^4*n^3+15*A*b^2*c*d^3*n^3
+18*B*a^2*d^4*n^3+30*B*a*b*c*d^3*n^3-3*B*b^2*c^2*d^2*n^3+15*C*a^2*c*d^3*n^3
-6*C*a*b*c^2*d^2*n^3-3*D*a^2*c^2*d^2*n^3+238*A*a*b*d^4*n^2+74*A*b^2*c*d^3*n
^2+119*B*a^2*d^4*n^2+148*B*a*b*c*d^3*n^2-33*B*b^2*c^2*d^2*n^2+74*C*a^2*c*d^
3*n^2-66*C*a*b*c^2*d^2*n^2+12*C*b^2*c^3*d*n^2-33*D*a^2*c^2*d^2*n^2+24*D*a*b
*c^3*d*n^2+684*A*a*b*d^4*n+120*A*b^2*c*d^3*n+342*B*a^2*d^4*n+240*B*a*b*c*d^
3*n-90*B*b^2*c^2*d^2*n+120*C*a^2*c*d^3*n-180*C*a*b*c^2*d^2*n+72*C*b^2*c^3*d
*n-90*D*a^2*c^2*d^2*n+144*D*a*b*c^3*d*n-60*D*b^2*c^4*n+720*A*a*b*d^4+360*B*
a^2*d^4)/d^4/(n^5+20*n^4+155*n^3+580*n^2+1044*n+720)*x^2*exp(n*ln(d*x+c))+(
A*a^2*d^5*n^5+2*A*a*b*c*d^4*n^5+B*a^2*c*d^4*n^5+20*A*a^2*d^5*n^4+36*A*a*b*c
*d^4*n^4-2*A*b^2*c^2*d^3*n^4+18*B*a^2*c*d^4*n^4-4*B*a*b*c^2*d^3*n^4-2*C*a^2
```

```
*c^2*d^3*n^4+155*A*a^2*d^5*n^3+238*A*a*b*c*d^4*n^3-30*A*b^2*c^2*d^3*n^3+119
*B*a^2*c*d^4*n^3-60*B*a*b*c^2*d^3*n^3+6*B*b^2*c^3*d^2*n^3-30*C*a^2*c^2*d^3*
n^3+12*C*a*b*c^3*d^2*n^3+6*D*a^2*c^3*d^2*n^3+580*A*a^2*d^5*n^2+684*A*a*b*c*
d^4*n^2-148*A*b^2*c^2*d^3*n^2+342*B*a^2*c*d^4*n^2-296*B*a*b*c^2*d^3*n^2+66*
B*b^2*c^3*d^2*n^2-148*C*a^2*c^2*d^3*n^2+132*C*a*b*c^3*d^2*n^2-24*C*b^2*c^4*
d*n^2+66*D*a^2*c^3*d^2*n^2-48*D*a*b*c^4*d*n^2+1044*A*a^2*d^5*n+720*A*a*b*c*
d^4*n-240*A*b^2*c^2*d^3*n+360*B*a^2*c*d^4*n-480*B*a*b*c^2*d^3*n+180*B*b^2*c
^3*d^2*n-240*C*a^2*c^2*d^3*n+360*C*a*b*c^3*d^2*n-144*C*b^2*c^4*d*n+180*D*a^
2*c^3*d^2*n-288*D*a*b*c^4*d*n+120*D*b^2*c^5*n+720*A*a^2*d^5)/d^5/(n^6+21*n^
5+175*n^4+735*n^3+1624*n^2+1764*n+720)*x*exp(n*ln(d*x+c))+(C*b*d*n+2*D*a*d*
n+D*b*c*n+6*C*b*d+12*D*a*d)*b/d/(n^2+11*n+30)*x^5*exp(n*ln(d*x+c))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2258 vs. 2(342) = 684.

Time = 0.32 (sec) , antiderivative size = 2258, normalized size of antiderivative = 6.68

$$\int (a+bx)^2(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) \, dx = \text{Too large to display}$$

```
[In] integrate((b*x+a)^2*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="fricas")
```

```
[Out] (A*a^2*c*d^5*n^5 - 120*D*b^2*c^6 + 720*A*a^2*c*d^5 + 240*(C*a^2 + 2*B*a*b +
 A*b^2)*c^3*d^3 - 360*(B*a^2 + 2*A*a*b)*c^2*d^4 + (D*b^2*d^6*n^5 + 15*D*b^2
*d^6*n^4 + 85*D*b^2*d^6*n^3 + 225*D*b^2*d^6*n^2 + 274*D*b^2*d^6*n + 120*D*b
^2*d^6)*x^6 + (144*(2*D*a*b + C*b^2)*d^6 + (D*b^2*c*d^5 + (2*D*a*b + C*b^2)
*d^6)*n^5 + 2*(5*D*b^2*c*d^5 + 8*(2*D*a*b + C*b^2)*d^6)*n^4 + 5*(7*D*b^2*c*
d^5 + 19*(2*D*a*b + C*b^2)*d^6)*n^3 + 10*(5*D*b^2*c*d^5 + 26*(2*D*a*b + C*b
^2)*d^6)*n^2 + 12*(2*D*b^2*c*d^5 + 27*(2*D*a*b + C*b^2)*d^6)*n)*x^5 + (20*A
*a^2*c*d^5 - (B*a^2 + 2*A*a*b)*c^2*d^4)*n^4 + (180*(D*a^2 + 2*C*a*b + B*b^2
)*d^6 + ((D*a^2 + 2*C*a*b + B*b^2)*d^6 + (2*D*a*b*c + C*b^2*c)*d^5)*n^5 - (
5*D*b^2*c^2*d^4 - 17*(D*a^2 + 2*C*a*b + B*b^2)*d^6 - 12*(2*D*a*b*c + C*b^2*
c)*d^5)*n^4 - (30*D*b^2*c^2*d^4 - 107*(D*a^2 + 2*C*a*b + B*b^2)*d^6 - 47*(2
*D*a*b*c + C*b^2*c)*d^5)*n^3 - (55*D*b^2*c^2*d^4 - 307*(D*a^2 + 2*C*a*b + B
*b^2)*d^6 - 72*(2*D*a*b*c + C*b^2*c)*d^5)*n^2 - 6*(5*D*b^2*c^2*d^4 - 66*(D*
a^2 + 2*C*a*b + B*b^2)*d^6 - 6*(2*D*a*b*c + C*b^2*c)*d^5)*n)*x^4 + (155*A*a
^2*c*d^5 + 2*(C*a^2 + 2*B*a*b + A*b^2)*c^3*d^3 - 18*(B*a^2 + 2*A*a*b)*c^2*d
^4)*n^3 + (240*(C*a^2 + 2*B*a*b + A*b^2)*d^6 + ((C*a^2 + 2*B*a*b + A*b^2)*d
^6 + (D*a^2*c + (2*C*a*b + B*b^2)*c)*d^5)*n^5 + 2*(9*(C*a^2 + 2*B*a*b + A*b
^2)*d^6 + 7*(D*a^2*c + (2*C*a*b + B*b^2)*c)*d^5 - 2*(2*D*a*b*c^2 + C*b^2*c^
2)*d^4)*n^4 + (20*D*b^2*c^3*d^3 + 121*(C*a^2 + 2*B*a*b + A*b^2)*d^6 + 65*(D
*a^2*c + (2*C*a*b + B*b^2)*c)*d^5 - 36*(2*D*a*b*c^2 + C*b^2*c^2)*d^4)*n^3 +
 4*(15*D*b^2*c^3*d^3 + 93*(C*a^2 + 2*B*a*b + A*b^2)*d^6 + 28*(D*a^2*c + (2*
C*a*b + B*b^2)*c)*d^5 - 20*(2*D*a*b*c^2 + C*b^2*c^2)*d^4)*n^2 + 4*(10*D*b^2
*c^3*d^3 + 127*(C*a^2 + 2*B*a*b + A*b^2)*d^6 + 15*(D*a^2*c + (2*C*a*b + B*b
```

```
^2)*c)*d^5 - 12*(2*D*a*b*c^2 + C*b^2*c^2)*d^4)*n)*x^3 - 180*(D*a^2*c^4 + (2
*C*a*b + B*b^2)*c^4)*d^2 + (580*A*a^2*c*d^5 + 30*(C*a^2 + 2*B*a*b + A*b^2)*
c^3*d^3 - 119*(B*a^2 + 2*A*a*b)*c^2*d^4 - 6*(D*a^2*c^4 + (2*C*a*b + B*b^2)*
c^4)*d^2)*n^2 + (360*(B*a^2 + 2*A*a*b)*d^6 + ((C*a^2 + 2*B*a*b + A*b^2)*c*d
^5 + (B*a^2 + 2*A*a*b)*d^6)*n^5 + (16*(C*a^2 + 2*B*a*b + A*b^2)*c*d^5 + 19*
(B*a^2 + 2*A*a*b)*d^6 - 3*(D*a^2*c^2 + (2*C*a*b + B*b^2)*c^2)*d^4)*n^4 + (8
9*(C*a^2 + 2*B*a*b + A*b^2)*c*d^5 + 137*(B*a^2 + 2*A*a*b)*d^6 - 36*(D*a^2*c
^2 + (2*C*a*b + B*b^2)*c^2)*d^4 + 12*(2*D*a*b*c^3 + C*b^2*c^3)*d^3)*n^3 - (
60*D*b^2*c^4*d^2 - 194*(C*a^2 + 2*B*a*b + A*b^2)*c*d^5 - 461*(B*a^2 + 2*A*a
*b)*d^6 + 123*(D*a^2*c^2 + (2*C*a*b + B*b^2)*c^2)*d^4 - 84*(2*D*a*b*c^3 + C
*b^2*c^3)*d^3)*n^2 - 6*(10*D*b^2*c^4*d^2 - 20*(C*a^2 + 2*B*a*b + A*b^2)*c*d
^5 - 117*(B*a^2 + 2*A*a*b)*d^6 + 15*(D*a^2*c^2 + (2*C*a*b + B*b^2)*c^2)*d^4
 - 12*(2*D*a*b*c^3 + C*b^2*c^3)*d^3)*n)*x^2 + 144*(2*D*a*b*c^5 + C*b^2*c^5)
*d + 2*(522*A*a^2*c*d^5 + 74*(C*a^2 + 2*B*a*b + A*b^2)*c^3*d^3 - 171*(B*a^2
 + 2*A*a*b)*c^2*d^4 - 33*(D*a^2*c^4 + (2*C*a*b + B*b^2)*c^4)*d^2 + 12*(2*D*
a*b*c^5 + C*b^2*c^5)*d)*n + (720*A*a^2*d^6 + (A*a^2*d^6 + (B*a^2 + 2*A*a*b)
*c*d^5)*n^5 + 2*(10*A*a^2*d^6 - (C*a^2 + 2*B*a*b + A*b^2)*c^2*d^4 + 9*(B*a^
2 + 2*A*a*b)*c*d^5)*n^4 + (155*A*a^2*d^6 - 30*(C*a^2 + 2*B*a*b + A*b^2)*c^2
*d^4 + 119*(B*a^2 + 2*A*a*b)*c*d^5 + 6*(D*a^2*c^3 + (2*C*a*b + B*b^2)*c^3)*
d^3)*n^3 + 2*(290*A*a^2*d^6 - 74*(C*a^2 + 2*B*a*b + A*b^2)*c^2*d^4 + 171*(B
*a^2 + 2*A*a*b)*c*d^5 + 33*(D*a^2*c^3 + (2*C*a*b + B*b^2)*c^3)*d^3 - 12*(2*
D*a*b*c^4 + C*b^2*c^4)*d^2)*n^2 + 12*(10*D*b^2*c^5*d + 87*A*a^2*d^6 - 20*(C
*a^2 + 2*B*a*b + A*b^2)*c^2*d^4 + 30*(B*a^2 + 2*A*a*b)*c*d^5 + 15*(D*a^2*c^
3 + (2*C*a*b + B*b^2)*c^3)*d^3 - 12*(2*D*a*b*c^4 + C*b^2*c^4)*d^2)*n)*x)*(d
*x + c)^n/(d^6*n^6 + 21*d^6*n^5 + 175*d^6*n^4 + 735*d^6*n^3 + 1624*d^6*n^2
+ 1764*d^6*n + 720*d^6)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 32849 vs. 2(328) = 656.

Time = 6.73 (sec) , antiderivative size = 32849, normalized size of antiderivative = 97.19

$$\int (a+bx)^2(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right)\,dx = \text{Too large to display}$$

```
[In] integrate((b*x+a)**2*(d*x+c)**n*(D*x**3+C*x**2+B*x+A),x)
```

```
[Out] Piecewise((c**n*(A*a**2*x + A*a*b*x**2 + A*b**2*x**3/3 + B*a**2*x**2/2 + 2*
B*a*b*x**3/3 + B*b**2*x**4/4 + C*a**2*x**3/3 + C*a*b*x**4/2 + C*b**2*x**5/5
 + D*a**2*x**4/4 + 2*D*a*b*x**5/5 + D*b**2*x**6/6), Eq(d, 0)), (-12*A*a**2*
d**5/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x
**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 6*A*a*b*c*d**4/(60*c**5*d**6 + 30
0*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4
+ 60*d**11*x**5) - 30*A*a*b*d**5*x/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c*
*3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 2*A
```

```
*b**2*c**2*d**3/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*
c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 10*A*b**2*c*d**4*x/(60
*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 30
0*c*d**10*x**4 + 60*d**11*x**5) - 20*A*b**2*d**5*x**2/(60*c**5*d**6 + 300*c
**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 6
0*d**11*x**5) - 3*B*a**2*c*d**4/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*
d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 15*B*a
**2*d**5*x/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*
d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 4*B*a*b*c**2*d**3/(60*c**5*
d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d*
*10*x**4 + 60*d**11*x**5) - 20*B*a*b*c*d**4*x/(60*c**5*d**6 + 300*c**4*d**7
*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*
x**5) - 40*B*a*b*d**5*x**2/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*
x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 3*B*b**2*c*
*3*d**2/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**
9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 15*B*b**2*c**2*d**3*x/(60*c**5
*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d
**10*x**4 + 60*d**11*x**5) - 30*B*b**2*c*d**4*x**2/(60*c**5*d**6 + 300*c**4
*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d
**11*x**5) - 30*B*b**2*d**5*x**3/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3
*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 2*C*a
**2*c**2*d**3/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c*
*2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 10*C*a**2*c*d**4*x/(60*c
**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*
c*d**10*x**4 + 60*d**11*x**5) - 20*C*a**2*d**5*x**2/(60*c**5*d**6 + 300*c**
4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*
d**11*x**5) - 6*C*a*b*c**3*d**2/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*
d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 30*C*a
*b*c**2*d**3*x/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c
**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 60*C*a*b*c*d**4*x**2/(6
0*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 3
00*c*d**10*x**4 + 60*d**11*x**5) - 60*C*a*b*d**5*x**3/(60*c**5*d**6 + 300*c
**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 6
0*d**11*x**5) - 12*C*b**2*c**4*d/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3
*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 60*C*
b**2*c**3*d**2*x/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600
*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 120*C*b**2*c**2*d**3*
x**2/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x
**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 120*C*b**2*c*d**4*x**3/(60*c**5*d
**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**
10*x**4 + 60*d**11*x**5) - 60*C*b**2*d**5*x**4/(60*c**5*d**6 + 300*c**4*d**
7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11
*x**5) - 3*D*a**2*c**3*d**2/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8
*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 15*D*a**2*
c**2*d**3*x/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2
```

```
*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 30*D*a**2*c*d**4*x**2/(60*
c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300
*c*d**10*x**4 + 60*d**11*x**5) - 30*D*a**2*d**5*x**3/(60*c**5*d**6 + 300*c*
*4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60
*d**11*x**5) - 24*D*a*b*c**4*d/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d
**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 120*D*a
*b*c**3*d**2*x/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c
**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) - 240*D*a*b*c**2*d**3*x**
2/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3
 + 300*c*d**10*x**4 + 60*d**11*x**5) - 240*D*a*b*c*d**4*x**3/(60*c**5*d**6
+ 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x
**4 + 60*d**11*x**5) - 120*D*a*b*d**5*x**4/(60*c**5*d**6 + 300*c**4*d**7*x
+ 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**
5) + 60*D*b**2*c**5*log(c/d + x)/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3
*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) + 137*D
*b**2*c**5/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*
d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) + 300*D*b**2*c**4*d*x*log(c/d
 + x)/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*
x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) + 625*D*b**2*c**4*d*x/(60*c**5*d**
6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10
*x**4 + 60*d**11*x**5) + 600*D*b**2*c**3*d**2*x**2*log(c/d + x)/(60*c**5*d*
*6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**1
0*x**4 + 60*d**11*x**5) + 1100*D*b**2*c**3*d**2*x**2/(60*c**5*d**6 + 300*c*
*4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60
*d**11*x**5) + 600*D*b**2*c**2*d**3*x**3*log(c/d + x)/(60*c**5*d**6 + 300*c
**4*d**7*x + 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 6
0*d**11*x**5) + 900*D*b**2*c**2*d**3*x**3/(60*c**5*d**6 + 300*c**4*d**7*x +
 600*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5
) + 300*D*b**2*c*d**4*x**4*log(c/d + x)/(60*c**5*d**6 + 300*c**4*d**7*x + 6
00*c**3*d**8*x**2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5)
+ 300*D*b**2*c*d**4*x**4/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x*
*2 + 600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5) + 60*D*b**2*d**
5*x**5*log(c/d + x)/(60*c**5*d**6 + 300*c**4*d**7*x + 600*c**3*d**8*x**2 +
600*c**2*d**9*x**3 + 300*c*d**10*x**4 + 60*d**11*x**5), Eq(n, -6)), (-3*A*a
**2*d**5/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**
3 + 12*d**10*x**4) - 2*A*a*b*c*d**4/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**
2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 8*A*a*b*d**5*x/(12*c**4*d**
6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) -
A*b**2*c**2*d**3/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*
d**9*x**3 + 12*d**10*x**4) - 4*A*b**2*c*d**4*x/(12*c**4*d**6 + 48*c**3*d**7
*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 6*A*b**2*d**5*x*
*2/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12
*d**10*x**4) - B*a**2*c*d**4/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*
x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 4*B*a**2*d**5*x/(12*c**4*d**6 + 48
*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 2*B*a*
```

```
b*c**2*d**3/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*
x**3 + 12*d**10*x**4) - 8*B*a*b*c*d**4*x/(12*c**4*d**6 + 48*c**3*d**7*x + 7
2*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 12*B*a*b*d**5*x**2/(12
*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10
*x**4) - 3*B*b**2*c**3*d**2/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x
**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 12*B*b**2*c**2*d**3*x/(12*c**4*d**6
 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 1
8*B*b**2*c*d**4*x**2/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 4
8*c*d**9*x**3 + 12*d**10*x**4) - 12*B*b**2*d**5*x**3/(12*c**4*d**6 + 48*c**
3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - C*a**2*c**
2*d**3/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3
 + 12*d**10*x**4) - 4*C*a**2*c*d**4*x/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c*
*2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 6*C*a**2*d**5*x**2/(12*c**
4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**
4) - 6*C*a*b*c**3*d**2/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 +
 48*c*d**9*x**3 + 12*d**10*x**4) - 24*C*a*b*c**2*d**3*x/(12*c**4*d**6 + 48*
c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 36*C*a*
b*c*d**4*x**2/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**
9*x**3 + 12*d**10*x**4) - 24*C*a*b*d**5*x**3/(12*c**4*d**6 + 48*c**3*d**7*x
 + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) + 12*C*b**2*c**4*d*l
og(c/d + x)/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*
x**3 + 12*d**10*x**4) + 25*C*b**2*c**4*d/(12*c**4*d**6 + 48*c**3*d**7*x + 7
2*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) + 48*C*b**2*c**3*d**2*x*
log(c/d + x)/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9
*x**3 + 12*d**10*x**4) + 88*C*b**2*c**3*d**2*x/(12*c**4*d**6 + 48*c**3*d**7
*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) + 72*C*b**2*c**2*d
**3*x**2*log(c/d + x)/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 +
48*c*d**9*x**3 + 12*d**10*x**4) + 108*C*b**2*c**2*d**3*x**2/(12*c**4*d**6 +
 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) + 48*
C*b**2*c*d**4*x**3*log(c/d + x)/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d*
*8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) + 48*C*b**2*c*d**4*x**3/(12*c**4*
d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4)
 + 12*C*b**2*d**5*x**4*log(c/d + x)/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**
2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 3*D*a**2*c**3*d**2/(12*c**4
*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4
) - 12*D*a**2*c**2*d**3*x/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**
2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 18*D*a**2*c*d**4*x**2/(12*c**4*d**6 +
 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 12*
D*a**2*d**5*x**3/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*
d**9*x**3 + 12*d**10*x**4) + 24*D*a*b*c**4*d*log(c/d + x)/(12*c**4*d**6 + 4
8*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) + 50*D*
a*b*c**4*d/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x
**3 + 12*d**10*x**4) + 96*D*a*b*c**3*d**2*x*log(c/d + x)/(12*c**4*d**6 + 48
*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) + 176*D*
a*b*c**3*d**2*x/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d
```

```
**9*x**3 + 12*d**10*x**4) + 144*D*a*b*c**2*d**3*x**2*log(c/d + x)/(12*c**4*
d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4)
 + 216*D*a*b*c**2*d**3*x**2/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x
**2 + 48*c*d**9*x**3 + 12*d**10*x**4) + 96*D*a*b*c*d**4*x**3*log(c/d + x)/(
12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**
10*x**4) + 96*D*a*b*c*d**4*x**3/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d*
*8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) + 24*D*a*b*d**5*x**4*log(c/d + x)
/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d
**10*x**4) - 60*D*b**2*c**5*log(c/d + x)/(12*c**4*d**6 + 48*c**3*d**7*x + 7
2*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 125*D*b**2*c**5/(12*c*
*4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x*
*4) - 240*D*b**2*c**4*d*x*log(c/d + x)/(12*c**4*d**6 + 48*c**3*d**7*x + 72*
c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 440*D*b**2*c**4*d*x/(12*
c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*
x**4) - 360*D*b**2*c**3*d**2*x**2*log(c/d + x)/(12*c**4*d**6 + 48*c**3*d**7
*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 540*D*b**2*c**3*
d**2*x**2/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x*
*3 + 12*d**10*x**4) - 240*D*b**2*c**2*d**3*x**3*log(c/d + x)/(12*c**4*d**6
+ 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*x**4) - 24
0*D*b**2*c**2*d**3*x**3/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2
+ 48*c*d**9*x**3 + 12*d**10*x**4) - 60*D*b**2*c*d**4*x**4*log(c/d + x)/(12*
c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x**2 + 48*c*d**9*x**3 + 12*d**10*
x**4) + 12*D*b**2*d**5*x**5/(12*c**4*d**6 + 48*c**3*d**7*x + 72*c**2*d**8*x
**2 + 48*c*d**9*x**3 + 12*d**10*x**4), Eq(n, -5)), (-2*A*a**2*d**5/(6*c**3*
d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) - 2*A*a*b*c*d**4/(6*c
**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) - 6*A*a*b*d**5*x/
(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) - 2*A*b**2*c*
*2*d**3/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) - 6*A
*b**2*c*d**4*x/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3
) - 6*A*b**2*d**5*x**2/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d
**9*x**3) - B*a**2*c*d**4/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 +
6*d**9*x**3) - 3*B*a**2*d**5*x/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x*
*2 + 6*d**9*x**3) - 4*B*a*b*c**2*d**3/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*
d**8*x**2 + 6*d**9*x**3) - 12*B*a*b*c*d**4*x/(6*c**3*d**6 + 18*c**2*d**7*x
+ 18*c*d**8*x**2 + 6*d**9*x**3) - 12*B*a*b*d**5*x**2/(6*c**3*d**6 + 18*c**2
*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 6*B*b**2*c**3*d**2*log(c/d + x)/(
6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 11*B*b**2*c*
*3*d**2/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 18*
B*b**2*c**2*d**3*x*log(c/d + x)/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x
**2 + 6*d**9*x**3) + 27*B*b**2*c**2*d**3*x/(6*c**3*d**6 + 18*c**2*d**7*x +
18*c*d**8*x**2 + 6*d**9*x**3) + 18*B*b**2*c*d**4*x**2*log(c/d + x)/(6*c**3*
d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 18*B*b**2*c*d**4*x*
*2/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 6*B*b**2
*d**5*x**3*log(c/d + x)/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*
d**9*x**3) - 2*C*a**2*c**2*d**3/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x
```

```
**2 + 6*d**9*x**3) - 6*C*a**2*c*d**4*x/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c
*d**8*x**2 + 6*d**9*x**3) - 6*C*a**2*d**5*x**2/(6*c**3*d**6 + 18*c**2*d**7*
x + 18*c*d**8*x**2 + 6*d**9*x**3) + 12*C*a*b*c**3*d**2*log(c/d + x)/(6*c**3
*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 22*C*a*b*c**3*d**2
/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 36*C*a*b*c
**2*d**3*x*log(c/d + x)/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*
d**9*x**3) + 54*C*a*b*c**2*d**3*x/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8
*x**2 + 6*d**9*x**3) + 36*C*a*b*c*d**4*x**2*log(c/d + x)/(6*c**3*d**6 + 18*
c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 36*C*a*b*c*d**4*x**2/(6*c**3*
d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 12*C*a*b*d**5*x**3*
log(c/d + x)/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3)
- 24*C*b**2*c**4*d*log(c/d + x)/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x
**2 + 6*d**9*x**3) - 44*C*b**2*c**4*d/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*
d**8*x**2 + 6*d**9*x**3) - 72*C*b**2*c**3*d**2*x*log(c/d + x)/(6*c**3*d**6
+ 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) - 108*C*b**2*c**3*d**2*x/(
6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) - 72*C*b**2*c*
*2*d**3*x**2*log(c/d + x)/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 +
6*d**9*x**3) - 72*C*b**2*c**2*d**3*x**2/(6*c**3*d**6 + 18*c**2*d**7*x + 18*
c*d**8*x**2 + 6*d**9*x**3) - 24*C*b**2*c*d**4*x**3*log(c/d + x)/(6*c**3*d**
6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 6*C*b**2*d**5*x**4/(6*
c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 6*D*a**2*c**3*
d**2*log(c/d + x)/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x
**3) + 11*D*a**2*c**3*d**2/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 +
 6*d**9*x**3) + 18*D*a**2*c**2*d**3*x*log(c/d + x)/(6*c**3*d**6 + 18*c**2*d
**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 27*D*a**2*c**2*d**3*x/(6*c**3*d**6
+ 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 18*D*a**2*c*d**4*x**2*lo
g(c/d + x)/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) +
18*D*a**2*c*d**4*x**2/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d*
*9*x**3) + 6*D*a**2*d**5*x**3*log(c/d + x)/(6*c**3*d**6 + 18*c**2*d**7*x +
18*c*d**8*x**2 + 6*d**9*x**3) - 48*D*a*b*c**4*d*log(c/d + x)/(6*c**3*d**6 +
 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) - 88*D*a*b*c**4*d/(6*c**3*d
**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) - 144*D*a*b*c**3*d**2*
x*log(c/d + x)/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3
) - 216*D*a*b*c**3*d**2*x/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 +
6*d**9*x**3) - 144*D*a*b*c**2*d**3*x**2*log(c/d + x)/(6*c**3*d**6 + 18*c**2
*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) - 144*D*a*b*c**2*d**3*x**2/(6*c**3*
d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) - 48*D*a*b*c*d**4*x**
3*log(c/d + x)/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3
) + 12*D*a*b*d**5*x**4/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d
**9*x**3) + 60*D*b**2*c**5*log(c/d + x)/(6*c**3*d**6 + 18*c**2*d**7*x + 18*
c*d**8*x**2 + 6*d**9*x**3) + 110*D*b**2*c**5/(6*c**3*d**6 + 18*c**2*d**7*x
+ 18*c*d**8*x**2 + 6*d**9*x**3) + 180*D*b**2*c**4*d*x*log(c/d + x)/(6*c**3*
d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 270*D*b**2*c**4*d*x
/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 180*D*b**2
*c**3*d**2*x**2*log(c/d + x)/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2
```

```
 + 6*d**9*x**3) + 180*D*b**2*c**3*d**2*x**2/(6*c**3*d**6 + 18*c**2*d**7*x +
 18*c*d**8*x**2 + 6*d**9*x**3) + 60*D*b**2*c**2*d**3*x**3*log(c/d + x)/(6*c
**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) - 15*D*b**2*c*d**
4*x**4/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3) + 3*D*
b**2*d**5*x**5/(6*c**3*d**6 + 18*c**2*d**7*x + 18*c*d**8*x**2 + 6*d**9*x**3
), Eq(n, -4)), (-3*A*a**2*d**5/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) -
6*A*a*b*c*d**4/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) - 12*A*a*b*d**5*x/
(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 6*A*b**2*c**2*d**3*log(c/d + x)
/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 9*A*b**2*c**2*d**3/(6*c**2*d**
6 + 12*c*d**7*x + 6*d**8*x**2) + 12*A*b**2*c*d**4*x*log(c/d + x)/(6*c**2*d*
*6 + 12*c*d**7*x + 6*d**8*x**2) + 12*A*b**2*c*d**4*x/(6*c**2*d**6 + 12*c*d*
*7*x + 6*d**8*x**2) + 6*A*b**2*d**5*x**2*log(c/d + x)/(6*c**2*d**6 + 12*c*d
**7*x + 6*d**8*x**2) - 3*B*a**2*c*d**4/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*
x**2) - 6*B*a**2*d**5*x/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 12*B*a*
b*c**2*d**3*log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 18*B*a
*b*c**2*d**3/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 24*B*a*b*c*d**4*x*
log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 24*B*a*b*c*d**4*x/
(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 12*B*a*b*d**5*x**2*log(c/d + x)
/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) - 18*B*b**2*c**3*d**2*log(c/d +
x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) - 27*B*b**2*c**3*d**2/(6*c**2*
d**6 + 12*c*d**7*x + 6*d**8*x**2) - 36*B*b**2*c**2*d**3*x*log(c/d + x)/(6*c
**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) - 36*B*b**2*c**2*d**3*x/(6*c**2*d**6
+ 12*c*d**7*x + 6*d**8*x**2) - 18*B*b**2*c*d**4*x**2*log(c/d + x)/(6*c**2*d
**6 + 12*c*d**7*x + 6*d**8*x**2) + 6*B*b**2*d**5*x**3/(6*c**2*d**6 + 12*c*d
**7*x + 6*d**8*x**2) + 6*C*a**2*c**2*d**3*log(c/d + x)/(6*c**2*d**6 + 12*c*
d**7*x + 6*d**8*x**2) + 9*C*a**2*c**2*d**3/(6*c**2*d**6 + 12*c*d**7*x + 6*d
**8*x**2) + 12*C*a**2*c*d**4*x*log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6*
d**8*x**2) + 12*C*a**2*c*d**4*x/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) +
 6*C*a**2*d**5*x**2*log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2)
- 36*C*a*b*c**3*d**2*log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2)
 - 54*C*a*b*c**3*d**2/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) - 72*C*a*b*
c**2*d**3*x*log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) - 72*C*a
*b*c**2*d**3*x/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) - 36*C*a*b*c*d**4*
x**2*log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 12*C*a*b*d**5
*x**3/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 36*C*b**2*c**4*d*log(c/d
+ x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 54*C*b**2*c**4*d/(6*c**2*d
**6 + 12*c*d**7*x + 6*d**8*x**2) + 72*C*b**2*c**3*d**2*x*log(c/d + x)/(6*c*
*2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 72*C*b**2*c**3*d**2*x/(6*c**2*d**6 +
 12*c*d**7*x + 6*d**8*x**2) + 36*C*b**2*c**2*d**3*x**2*log(c/d + x)/(6*c**2
*d**6 + 12*c*d**7*x + 6*d**8*x**2) - 12*C*b**2*c*d**4*x**3/(6*c**2*d**6 + 1
2*c*d**7*x + 6*d**8*x**2) + 3*C*b**2*d**5*x**4/(6*c**2*d**6 + 12*c*d**7*x +
 6*d**8*x**2) - 18*D*a**2*c**3*d**2*log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x
 + 6*d**8*x**2) - 27*D*a**2*c**3*d**2/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x
**2) - 36*D*a**2*c**2*d**3*x*log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d*
*8*x**2) - 36*D*a**2*c**2*d**3*x/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2)
```

```
- 18*D*a**2*c*d**4*x**2*log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x*
*2) + 6*D*a**2*d**5*x**3/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 72*D*a
*b*c**4*d*log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 108*D*a*
b*c**4*d/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 144*D*a*b*c**3*d**2*x*
log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 144*D*a*b*c**3*d**
2*x/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 72*D*a*b*c**2*d**3*x**2*log
(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) - 24*D*a*b*c*d**4*x**3/
(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 6*D*a*b*d**5*x**4/(6*c**2*d**6
+ 12*c*d**7*x + 6*d**8*x**2) - 60*D*b**2*c**5*log(c/d + x)/(6*c**2*d**6 + 1
2*c*d**7*x + 6*d**8*x**2) - 90*D*b**2*c**5/(6*c**2*d**6 + 12*c*d**7*x + 6*d
**8*x**2) - 120*D*b**2*c**4*d*x*log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6
*d**8*x**2) - 120*D*b**2*c**4*d*x/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2)
 - 60*D*b**2*c**3*d**2*x**2*log(c/d + x)/(6*c**2*d**6 + 12*c*d**7*x + 6*d**
8*x**2) + 20*D*b**2*c**2*d**3*x**3/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2
) - 5*D*b**2*c*d**4*x**4/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2) + 2*D*b*
*2*d**5*x**5/(6*c**2*d**6 + 12*c*d**7*x + 6*d**8*x**2), Eq(n, -3)), (-12*A*
a**2*d**5/(12*c*d**6 + 12*d**7*x) + 24*A*a*b*c*d**4*log(c/d + x)/(12*c*d**6
 + 12*d**7*x) + 24*A*a*b*c*d**4/(12*c*d**6 + 12*d**7*x) + 24*A*a*b*d**5*x*l
og(c/d + x)/(12*c*d**6 + 12*d**7*x) - 24*A*b**2*c**2*d**3*log(c/d + x)/(12*
c*d**6 + 12*d**7*x) - 24*A*b**2*c**2*d**3/(12*c*d**6 + 12*d**7*x) - 24*A*b*
*2*c*d**4*x*log(c/d + x)/(12*c*d**6 + 12*d**7*x) + 12*A*b**2*d**5*x**2/(12*
c*d**6 + 12*d**7*x) + 12*B*a**2*c*d**4*log(c/d + x)/(12*c*d**6 + 12*d**7*x)
 + 12*B*a**2*c*d**4/(12*c*d**6 + 12*d**7*x) + 12*B*a**2*d**5*x*log(c/d + x)
/(12*c*d**6 + 12*d**7*x) - 48*B*a*b*c**2*d**3*log(c/d + x)/(12*c*d**6 + 12*
d**7*x) - 48*B*a*b*c**2*d**3/(12*c*d**6 + 12*d**7*x) - 48*B*a*b*c*d**4*x*lo
g(c/d + x)/(12*c*d**6 + 12*d**7*x) + 24*B*a*b*d**5*x**2/(12*c*d**6 + 12*d**
7*x) + 36*B*b**2*c**3*d**2*log(c/d + x)/(12*c*d**6 + 12*d**7*x) + 36*B*b**2
*c**3*d**2/(12*c*d**6 + 12*d**7*x) + 36*B*b**2*c**2*d**3*x*log(c/d + x)/(12
*c*d**6 + 12*d**7*x) - 18*B*b**2*c*d**4*x**2/(12*c*d**6 + 12*d**7*x) + 6*B*
b**2*d**5*x**3/(12*c*d**6 + 12*d**7*x) - 24*C*a**2*c**2*d**3*log(c/d + x)/(
12*c*d**6 + 12*d**7*x) - 24*C*a**2*c**2*d**3/(12*c*d**6 + 12*d**7*x) - 24*C
*a**2*c*d**4*x*log(c/d + x)/(12*c*d**6 + 12*d**7*x) + 12*C*a**2*d**5*x**2/(
12*c*d**6 + 12*d**7*x) + 72*C*a*b*c**3*d**2*log(c/d + x)/(12*c*d**6 + 12*d*
*7*x) + 72*C*a*b*c**3*d**2/(12*c*d**6 + 12*d**7*x) + 72*C*a*b*c**2*d**3*x*l
og(c/d + x)/(12*c*d**6 + 12*d**7*x) - 36*C*a*b*c*d**4*x**2/(12*c*d**6 + 12*
d**7*x) + 12*C*a*b*d**5*x**3/(12*c*d**6 + 12*d**7*x) - 48*C*b**2*c**4*d*log
(c/d + x)/(12*c*d**6 + 12*d**7*x) - 48*C*b**2*c**4*d/(12*c*d**6 + 12*d**7*x
) - 48*C*b**2*c**3*d**2*x*log(c/d + x)/(12*c*d**6 + 12*d**7*x) + 24*C*b**2*
c**2*d**3*x**2/(12*c*d**6 + 12*d**7*x) - 8*C*b**2*c*d**4*x**3/(12*c*d**6 +
12*d**7*x) + 4*C*b**2*d**5*x**4/(12*c*d**6 + 12*d**7*x) + 36*D*a**2*c**3*d*
*2*log(c/d + x)/(12*c*d**6 + 12*d**7*x) + 36*D*a**2*c**3*d**2/(12*c*d**6 +
12*d**7*x) + 36*D*a**2*c**2*d**3*x*log(c/d + x)/(12*c*d**6 + 12*d**7*x) - 1
8*D*a**2*c*d**4*x**2/(12*c*d**6 + 12*d**7*x) + 6*D*a**2*d**5*x**3/(12*c*d**
6 + 12*d**7*x) - 96*D*a*b*c**4*d*log(c/d + x)/(12*c*d**6 + 12*d**7*x) - 96*
D*a*b*c**4*d/(12*c*d**6 + 12*d**7*x) - 96*D*a*b*c**3*d**2*x*log(c/d + x)/(1
```

```
2*c*d**6 + 12*d**7*x) + 48*D*a*b*c**2*d**3*x**2/(12*c*d**6 + 12*d**7*x) - 1
6*D*a*b*c*d**4*x**3/(12*c*d**6 + 12*d**7*x) + 8*D*a*b*d**5*x**4/(12*c*d**6
+ 12*d**7*x) + 60*D*b**2*c**5*log(c/d + x)/(12*c*d**6 + 12*d**7*x) + 60*D*b
**2*c**5/(12*c*d**6 + 12*d**7*x) + 60*D*b**2*c**4*d*x*log(c/d + x)/(12*c*d*
*6 + 12*d**7*x) - 30*D*b**2*c**3*d**2*x**2/(12*c*d**6 + 12*d**7*x) + 10*D*b
**2*c**2*d**3*x**3/(12*c*d**6 + 12*d**7*x) - 5*D*b**2*c*d**4*x**4/(12*c*d**
6 + 12*d**7*x) + 3*D*b**2*d**5*x**5/(12*c*d**6 + 12*d**7*x), Eq(n, -2)), (A
*a**2*log(c/d + x)/d - 2*A*a*b*c*log(c/d + x)/d**2 + 2*A*a*b*x/d + A*b**2*c
**2*log(c/d + x)/d**3 - A*b**2*c*x/d**2 + A*b**2*x**2/(2*d) - B*a**2*c*log(
c/d + x)/d**2 + B*a**2*x/d + 2*B*a*b*c**2*log(c/d + x)/d**3 - 2*B*a*b*c*x/d
**2 + B*a*b*x**2/d - B*b**2*c**3*log(c/d + x)/d**4 + B*b**2*c**2*x/d**3 - B
*b**2*c*x**2/(2*d**2) + B*b**2*x**3/(3*d) + C*a**2*c**2*log(c/d + x)/d**3 -
 C*a**2*c*x/d**2 + C*a**2*x**2/(2*d) - 2*C*a*b*c**3*log(c/d + x)/d**4 + 2*C
*a*b*c**2*x/d**3 - C*a*b*c*x**2/d**2 + 2*C*a*b*x**3/(3*d) + C*b**2*c**4*log
(c/d + x)/d**5 - C*b**2*c**3*x/d**4 + C*b**2*c**2*x**2/(2*d**3) - C*b**2*c*
x**3/(3*d**2) + C*b**2*x**4/(4*d) - D*a**2*c**3*log(c/d + x)/d**4 + D*a**2*
c**2*x/d**3 - D*a**2*c*x**2/(2*d**2) + D*a**2*x**3/(3*d) + 2*D*a*b*c**4*log
(c/d + x)/d**5 - 2*D*a*b*c**3*x/d**4 + D*a*b*c**2*x**2/d**3 - 2*D*a*b*c*x**
3/(3*d**2) + D*a*b*x**4/(2*d) - D*b**2*c**5*log(c/d + x)/d**6 + D*b**2*c**4
*x/d**5 - D*b**2*c**3*x**2/(2*d**4) + D*b**2*c**2*x**3/(3*d**3) - D*b**2*c*
x**4/(4*d**2) + D*b**2*x**5/(5*d), Eq(n, -1)), (A*a**2*c*d**5*n**5*(c + d*x
)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*
n**2 + 1764*d**6*n + 720*d**6) + 20*A*a**2*c*d**5*n**4*(c + d*x)**n/(d**6*n
**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*
d**6*n + 720*d**6) + 155*A*a**2*c*d**5*n**3*(c + d*x)**n/(d**6*n**6 + 21*d*
*6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 72
0*d**6) + 580*A*a**2*c*d**5*n**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 1
75*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 1
044*A*a**2*c*d**5*n*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4
+ 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 720*A*a**2*c*d
**5*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3
+ 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + A*a**2*d**6*n**5*x*(c + d*x)**
n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**
2 + 1764*d**6*n + 720*d**6) + 20*A*a**2*d**6*n**4*x*(c + d*x)**n/(d**6*n**6
 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**
6*n + 720*d**6) + 155*A*a**2*d**6*n**3*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*
n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d
**6) + 580*A*a**2*d**6*n**2*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*
d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 1044
*A*a**2*d**6*n*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 7
35*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 720*A*a**2*d**6*x
*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1
624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 2*A*a*b*c**2*d**4*n**4*(c + d*x)*
*n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n*
*2 + 1764*d**6*n + 720*d**6) - 36*A*a*b*c**2*d**4*n**3*(c + d*x)**n/(d**6*n
```

```
**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*
d**6*n + 720*d**6) - 238*A*a*b*c**2*d**4*n**2*(c + d*x)**n/(d**6*n**6 + 21*
d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n +
720*d**6) - 684*A*a*b*c**2*d**4*n*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 +
175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) -
720*A*a*b*c**2*d**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4
+ 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 2*A*a*b*c*d**5
*n**5*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n
**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 36*A*a*b*c*d**5*n**4*x*(c
+ d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*
d**6*n**2 + 1764*d**6*n + 720*d**6) + 238*A*a*b*c*d**5*n**3*x*(c + d*x)**n/
(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2
+ 1764*d**6*n + 720*d**6) + 684*A*a*b*c*d**5*n**2*x*(c + d*x)**n/(d**6*n**6
 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**
6*n + 720*d**6) + 720*A*a*b*c*d**5*n*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n*
*5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**
6) + 2*A*a*b*d**6*n**5*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d*
*6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 38*A*a
*b*d**6*n**4*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 +
735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 274*A*a*b*d**6*n
**3*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*
n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 922*A*a*b*d**6*n**2*x**2*
(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 16
24*d**6*n**2 + 1764*d**6*n + 720*d**6) + 1404*A*a*b*d**6*n*x**2*(c + d*x)**
n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**
2 + 1764*d**6*n + 720*d**6) + 720*A*a*b*d**6*x**2*(c + d*x)**n/(d**6*n**6 +
 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*
n + 720*d**6) + 2*A*b**2*c**3*d**3*n**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n
**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d*
*6) + 30*A*b**2*c**3*d**3*n**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175
*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 148
*A*b**2*c**3*d**3*n*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4
+ 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 240*A*b**2*c**
3*d**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n*
*3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 2*A*b**2*c**2*d**4*n**4*x*(
c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 162
4*d**6*n**2 + 1764*d**6*n + 720*d**6) - 30*A*b**2*c**2*d**4*n**3*x*(c + d*x
)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*
n**2 + 1764*d**6*n + 720*d**6) - 148*A*b**2*c**2*d**4*n**2*x*(c + d*x)**n/(
d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 +
 1764*d**6*n + 720*d**6) - 240*A*b**2*c**2*d**4*n*x*(c + d*x)**n/(d**6*n**6
 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**
6*n + 720*d**6) + A*b**2*c*d**5*n**5*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6
*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*
d**6) + 16*A*b**2*c*d**5*n**4*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 +
```

```
 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) +
 89*A*b**2*c*d**5*n**3*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d*
*6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 194*A*
b**2*c*d**5*n**2*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**
4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 120*A*b**2*c
*d**5*n*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d
**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + A*b**2*d**6*n**5*x**3
*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1
624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 18*A*b**2*d**6*n**4*x**3*(c + d*x
)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*
n**2 + 1764*d**6*n + 720*d**6) + 121*A*b**2*d**6*n**3*x**3*(c + d*x)**n/(d*
*6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1
764*d**6*n + 720*d**6) + 372*A*b**2*d**6*n**2*x**3*(c + d*x)**n/(d**6*n**6
+ 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6
*n + 720*d**6) + 508*A*b**2*d**6*n*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n
**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d*
*6) + 240*A*b**2*d**6*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**
6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - B*a**2*
c**2*d**4*n**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735
*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 18*B*a**2*c**2*d**4
*n**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**
3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 119*B*a**2*c**2*d**4*n**2*(c
 + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624
*d**6*n**2 + 1764*d**6*n + 720*d**6) - 342*B*a**2*c**2*d**4*n*(c + d*x)**n/
(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2
+ 1764*d**6*n + 720*d**6) - 360*B*a**2*c**2*d**4*(c + d*x)**n/(d**6*n**6 +
21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n
 + 720*d**6) + B*a**2*c*d**5*n**5*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5
+ 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6)
+ 18*B*a**2*c*d**5*n**4*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6
*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 119*B*a*
*2*c*d**5*n**3*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 7
35*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 342*B*a**2*c*d**5
*n**2*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n
**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 360*B*a**2*c*d**5*n*x*(c +
 d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d
**6*n**2 + 1764*d**6*n + 720*d**6) + B*a**2*d**6*n**5*x**2*(c + d*x)**n/(d*
*6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1
764*d**6*n + 720*d**6) + 19*B*a**2*d**6*n**4*x**2*(c + d*x)**n/(d**6*n**6 +
 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*
n + 720*d**6) + 137*B*a**2*d**6*n**3*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6
*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*
d**6) + 461*B*a**2*d**6*n**2*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 +
175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) +
702*B*a**2*d**6*n*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n*
```

```
*4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 360*B*a**2*
d**6*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6
*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 4*B*a*b*c**3*d**3*n**3*(
c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 162
4*d**6*n**2 + 1764*d**6*n + 720*d**6) + 60*B*a*b*c**3*d**3*n**2*(c + d*x)**
n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**
2 + 1764*d**6*n + 720*d**6) + 296*B*a*b*c**3*d**3*n*(c + d*x)**n/(d**6*n**6
 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**
6*n + 720*d**6) + 480*B*a*b*c**3*d**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**
5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6
) - 4*B*a*b*c**2*d**4*n**4*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d
**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 60*B*
a*b*c**2*d**4*n**3*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4
 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 296*B*a*b*c**
2*d**4*n**2*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*
d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 480*B*a*b*c**2*d**4*
n*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3
 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 2*B*a*b*c*d**5*n**5*x**2*(c +
d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d*
*6*n**2 + 1764*d**6*n + 720*d**6) + 32*B*a*b*c*d**5*n**4*x**2*(c + d*x)**n/
(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2
 + 1764*d**6*n + 720*d**6) + 178*B*a*b*c*d**5*n**3*x**2*(c + d*x)**n/(d**6*n
**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*
d**6*n + 720*d**6) + 388*B*a*b*c*d**5*n**2*x**2*(c + d*x)**n/(d**6*n**6 + 2
1*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n
 + 720*d**6) + 240*B*a*b*c*d**5*n*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**
5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6
) + 2*B*a*b*d**6*n**5*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**
6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 36*B*a*
b*d**6*n**4*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 7
35*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 242*B*a*b*d**6*n*
*3*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n
**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 744*B*a*b*d**6*n**2*x**3*(
c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 162
4*d**6*n**2 + 1764*d**6*n + 720*d**6) + 1016*B*a*b*d**6*n*x**3*(c + d*x)**n
/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2
 + 1764*d**6*n + 720*d**6) + 480*B*a*b*d**6*x**3*(c + d*x)**n/(d**6*n**6 +
21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n
 + 720*d**6) - 6*B*b**2*c**4*d**2*n**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n*
*5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**
6) - 66*B*b**2*c**4*d**2*n*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**
6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 180*B*b
**2*c**4*d**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*
d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 6*B*b**2*c**3*d**3*n
**3*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**
```

```
3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 66*B*b**2*c**3*d**3*n**2*x*(
c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 162
4*d**6*n**2 + 1764*d**6*n + 720*d**6) + 180*B*b**2*c**3*d**3*n*x*(c + d*x)*
*n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n*
*2 + 1764*d**6*n + 720*d**6) - 3*B*b**2*c**2*d**4*n**4*x**2*(c + d*x)**n/(d
**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 +
1764*d**6*n + 720*d**6) - 36*B*b**2*c**2*d**4*n**3*x**2*(c + d*x)**n/(d**6*
n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764
*d**6*n + 720*d**6) - 123*B*b**2*c**2*d**4*n**2*x**2*(c + d*x)**n/(d**6*n**
6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d*
*6*n + 720*d**6) - 90*B*b**2*c**2*d**4*n*x**2*(c + d*x)**n/(d**6*n**6 + 21*
d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n +
720*d**6) + B*b**2*c*d**5*n**5*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5
+ 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6)
+ 14*B*b**2*c*d**5*n**4*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d
**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 65*B*
b**2*c*d**5*n**3*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**
4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 112*B*b**2*c
*d**5*n**2*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 73
5*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 60*B*b**2*c*d**5*n
*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**
3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + B*b**2*d**6*n**5*x**4*(c + d
*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**
6*n**2 + 1764*d**6*n + 720*d**6) + 17*B*b**2*d**6*n**4*x**4*(c + d*x)**n/(d
**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 +
1764*d**6*n + 720*d**6) + 107*B*b**2*d**6*n**3*x**4*(c + d*x)**n/(d**6*n**6
 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**
6*n + 720*d**6) + 307*B*b**2*d**6*n**2*x**4*(c + d*x)**n/(d**6*n**6 + 21*d*
*6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 72
0*d**6) + 396*B*b**2*d**6*n*x**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 1
75*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 1
80*B*b**2*d**6*x**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4
+ 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 2*C*a**2*c**3*
d**3*n**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6
*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 30*C*a**2*c**3*d**3*n**2
*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1
624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 148*C*a**2*c**3*d**3*n*(c + d*x)*
*n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n*
*2 + 1764*d**6*n + 720*d**6) + 240*C*a**2*c**3*d**3*(c + d*x)**n/(d**6*n**6
 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**
6*n + 720*d**6) - 2*C*a**2*c**2*d**4*n**4*x*(c + d*x)**n/(d**6*n**6 + 21*d*
*6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 72
0*d**6) - 30*C*a**2*c**2*d**4*n**3*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5
 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6)
 - 148*C*a**2*c**2*d**4*n**2*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175
```

```
*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 240
*C*a**2*c**2*d**4*n*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**
4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + C*a**2*c*d**
5*n**5*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d*
*6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 16*C*a**2*c*d**5*n**4*
x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3
 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 89*C*a**2*c*d**5*n**3*x**2*(c
 + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624
*d**6*n**2 + 1764*d**6*n + 720*d**6) + 194*C*a**2*c*d**5*n**2*x**2*(c + d*x
)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*
n**2 + 1764*d**6*n + 720*d**6) + 120*C*a**2*c*d**5*n*x**2*(c + d*x)**n/(d**
6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 17
64*d**6*n + 720*d**6) + C*a**2*d**6*n**5*x**3*(c + d*x)**n/(d**6*n**6 + 21*
d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n +
720*d**6) + 18*C*a**2*d**6*n**4*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5
 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6)
 + 121*C*a**2*d**6*n**3*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d
**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 372*C
*a**2*d**6*n**2*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4
 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 508*C*a**2*d*
*6*n*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6
*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 240*C*a**2*d**6*x**3*(c
+ d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*
d**6*n**2 + 1764*d**6*n + 720*d**6) - 12*C*a*b*c**4*d**2*n**2*(c + d*x)**n/
(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2
+ 1764*d**6*n + 720*d**6) - 132*C*a*b*c**4*d**2*n*(c + d*x)**n/(d**6*n**6 +
 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*
n + 720*d**6) - 360*C*a*b*c**4*d**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5
+ 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6)
+ 12*C*a*b*c**3*d**3*n**3*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d*
*6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 132*C*
a*b*c**3*d**3*n**2*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4
 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 360*C*a*b*c**
3*d**3*n*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**
6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 6*C*a*b*c**2*d**4*n**4*
x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3
 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 72*C*a*b*c**2*d**4*n**3*x**2*
(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 16
24*d**6*n**2 + 1764*d**6*n + 720*d**6) - 246*C*a*b*c**2*d**4*n**2*x**2*(c +
 d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d
**6*n**2 + 1764*d**6*n + 720*d**6) - 180*C*a*b*c**2*d**4*n*x**2*(c + d*x)**
n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**
2 + 1764*d**6*n + 720*d**6) + 2*C*a*b*c*d**5*n**5*x**3*(c + d*x)**n/(d**6*n
**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*
d**6*n + 720*d**6) + 28*C*a*b*c*d**5*n**4*x**3*(c + d*x)**n/(d**6*n**6 + 21
```

```
*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n +
 720*d**6) + 130*C*a*b*c*d**5*n**3*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n
**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d*
*6) + 224*C*a*b*c*d**5*n**2*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 1
75*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 1
20*C*a*b*c*d**5*n*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n*
*4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 2*C*a*b*d**
6*n**5*x**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d*
*6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 34*C*a*b*d**6*n**4*x**
4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 +
1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 214*C*a*b*d**6*n**3*x**4*(c + d*
x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6
*n**2 + 1764*d**6*n + 720*d**6) + 614*C*a*b*d**6*n**2*x**4*(c + d*x)**n/(d*
*6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1
764*d**6*n + 720*d**6) + 792*C*a*b*d**6*n*x**4*(c + d*x)**n/(d**6*n**6 + 21
*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n +
 720*d**6) + 360*C*a*b*d**6*x**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 1
75*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 2
4*C*b**2*c**5*d*n*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 +
735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 144*C*b**2*c**5*
d*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 +
1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 24*C*b**2*c**4*d**2*n**2*x*(c +
d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d*
*6*n**2 + 1764*d**6*n + 720*d**6) - 144*C*b**2*c**4*d**2*n*x*(c + d*x)**n/(
d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 +
 1764*d**6*n + 720*d**6) + 12*C*b**2*c**3*d**3*n**3*x**2*(c + d*x)**n/(d**6
*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 176
4*d**6*n + 720*d**6) + 84*C*b**2*c**3*d**3*n**2*x**2*(c + d*x)**n/(d**6*n**
6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d*
*6*n + 720*d**6) + 72*C*b**2*c**3*d**3*n*x**2*(c + d*x)**n/(d**6*n**6 + 21*
d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n +
720*d**6) - 4*C*b**2*c**2*d**4*n**4*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*
n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d
**6) - 36*C*b**2*c**2*d**4*n**3*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5
 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6)
 - 80*C*b**2*c**2*d**4*n**2*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 1
75*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 4
8*C*b**2*c**2*d**4*n*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6
*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + C*b**2*c
*d**5*n**5*x**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 73
5*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 12*C*b**2*c*d**5*n
**4*x**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*
n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 47*C*b**2*c*d**5*n**3*x**
4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 +
1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 72*C*b**2*c*d**5*n**2*x**4*(c +
```

```
d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d*
*6*n**2 + 1764*d**6*n + 720*d**6) + 36*C*b**2*c*d**5*n*x**4*(c + d*x)**n/(d
**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 +
1764*d**6*n + 720*d**6) + C*b**2*d**6*n**5*x**5*(c + d*x)**n/(d**6*n**6 + 2
1*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n
 + 720*d**6) + 16*C*b**2*d**6*n**4*x**5*(c + d*x)**n/(d**6*n**6 + 21*d**6*n*
*5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**
6) + 95*C*b**2*d**6*n**3*x**5*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*
d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 260*
C*b**2*d**6*n**2*x**5*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**
4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 324*C*b**2*d
**6*n*x**5*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**
6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 144*C*b**2*d**6*x**5*(c
 + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624
*d**6*n**2 + 1764*d**6*n + 720*d**6) - 6*D*a**2*c**4*d**2*n**2*(c + d*x)**n
/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2
 + 1764*d**6*n + 720*d**6) - 66*D*a**2*c**4*d**2*n*(c + d*x)**n/(d**6*n**6
+ 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6
*n + 720*d**6) - 180*D*a**2*c**4*d**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**
5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6
) + 6*D*a**2*c**3*d**3*n**3*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*
d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 66*D
*a**2*c**3*d**3*n**2*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n*
*4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 180*D*a**2*
c**3*d**3*n*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*
d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 3*D*a**2*c**2*d**4*n
**4*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*
n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 36*D*a**2*c**2*d**4*n**3*
x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3
 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 123*D*a**2*c**2*d**4*n**2*x**
2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 +
1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 90*D*a**2*c**2*d**4*n*x**2*(c +
d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d*
*6*n**2 + 1764*d**6*n + 720*d**6) + D*a**2*c*d**5*n**5*x**3*(c + d*x)**n/(d
**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 +
1764*d**6*n + 720*d**6) + 14*D*a**2*c*d**5*n**4*x**3*(c + d*x)**n/(d**6*n**
6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d*
*6*n + 720*d**6) + 65*D*a**2*c*d**5*n**3*x**3*(c + d*x)**n/(d**6*n**6 + 21*
d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n +
720*d**6) + 112*D*a**2*c*d**5*n**2*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n
**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d*
*6) + 60*D*a**2*c*d**5*n*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*
d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + D*a*
*2*d**6*n**5*x**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 +
735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 17*D*a**2*d**6*n
```

```
**4*x**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*
n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 107*D*a**2*d**6*n**3*x**4
*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1
624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 307*D*a**2*d**6*n**2*x**4*(c + d*
x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6
*n**2 + 1764*d**6*n + 720*d**6) + 396*D*a**2*d**6*n*x**4*(c + d*x)**n/(d**6
*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 176
4*d**6*n + 720*d**6) + 180*D*a**2*d**6*x**4*(c + d*x)**n/(d**6*n**6 + 21*d*
*6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 72
0*d**6) + 48*D*a*b*c**5*d*n*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d*
*6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 288*D*
a*b*c**5*d*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**
6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 48*D*a*b*c**4*d**2*n**2
*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 +
 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 288*D*a*b*c**4*d**2*n*x*(c + d*
x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6
*n**2 + 1764*d**6*n + 720*d**6) + 24*D*a*b*c**3*d**3*n**3*x**2*(c + d*x)**n
/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2
 + 1764*d**6*n + 720*d**6) + 168*D*a*b*c**3*d**3*n**2*x**2*(c + d*x)**n/(d*
*6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1
764*d**6*n + 720*d**6) + 144*D*a*b*c**3*d**3*n*x**2*(c + d*x)**n/(d**6*n**6
 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**
6*n + 720*d**6) - 8*D*a*b*c**2*d**4*n**4*x**3*(c + d*x)**n/(d**6*n**6 + 21*
d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n +
720*d**6) - 72*D*a*b*c**2*d**4*n**3*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*
n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d
**6) - 160*D*a*b*c**2*d**4*n**2*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5
 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6)
 - 96*D*a*b*c**2*d**4*n*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d
**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 2*D*a
*b*c*d**5*n**5*x**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4
+ 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 24*D*a*b*c*d**
5*n**4*x**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d*
*6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 94*D*a*b*c*d**5*n**3*x
**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3
+ 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 144*D*a*b*c*d**5*n**2*x**4*(c
+ d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*
d**6*n**2 + 1764*d**6*n + 720*d**6) + 72*D*a*b*c*d**5*n*x**4*(c + d*x)**n/(
d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 +
 1764*d**6*n + 720*d**6) + 2*D*a*b*d**6*n**5*x**5*(c + d*x)**n/(d**6*n**6 +
 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*
n + 720*d**6) + 32*D*a*b*d**6*n**4*x**5*(c + d*x)**n/(d**6*n**6 + 21*d**6*n
**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d*
*6) + 190*D*a*b*d**6*n**3*x**5*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175
*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 520
```

```
*D*a*b*d**6*n**2*x**5*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**
4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 648*D*a*b*d*
*6*n*x**5*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6
*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 288*D*a*b*d**6*x**5*(c +
 d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d
**6*n**2 + 1764*d**6*n + 720*d**6) - 120*D*b**2*c**6*(c + d*x)**n/(d**6*n**
6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d*
*6*n + 720*d**6) + 120*D*b**2*c**5*d*n*x*(c + d*x)**n/(d**6*n**6 + 21*d**6*
n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d
**6) - 60*D*b**2*c**4*d**2*n**2*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5
 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6)
 - 60*D*b**2*c**4*d**2*n*x**2*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*
d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 20*D
*b**2*c**3*d**3*n**3*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6
*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 60*D*b**
2*c**3*d**3*n**2*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**
4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 40*D*b**2*c*
*3*d**3*n*x**3*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735
*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 5*D*b**2*c**2*d**4*
n**4*x**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6
*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 30*D*b**2*c**2*d**4*n**3
*x**4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**
3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 55*D*b**2*c**2*d**4*n**2*x**
4*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 +
1624*d**6*n**2 + 1764*d**6*n + 720*d**6) - 30*D*b**2*c**2*d**4*n*x**4*(c +
d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d*
*6*n**2 + 1764*d**6*n + 720*d**6) + D*b**2*c*d**5*n**5*x**5*(c + d*x)**n/(d
**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 +
1764*d**6*n + 720*d**6) + 10*D*b**2*c*d**5*n**4*x**5*(c + d*x)**n/(d**6*n**
6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d*
*6*n + 720*d**6) + 35*D*b**2*c*d**5*n**3*x**5*(c + d*x)**n/(d**6*n**6 + 21*
d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n +
720*d**6) + 50*D*b**2*c*d**5*n**2*x**5*(c + d*x)**n/(d**6*n**6 + 21*d**6*n*
*5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**
6) + 24*D*b**2*c*d**5*n*x**5*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d
**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + D*b**
2*d**6*n**5*x**6*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 7
35*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 15*D*b**2*d**6*n*
*4*x**6*(c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n
**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*d**6) + 85*D*b**2*d**6*n**3*x**6*(
c + d*x)**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 162
4*d**6*n**2 + 1764*d**6*n + 720*d**6) + 225*D*b**2*d**6*n**2*x**6*(c + d*x)
**n/(d**6*n**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n
**2 + 1764*d**6*n + 720*d**6) + 274*D*b**2*d**6*n*x**6*(c + d*x)**n/(d**6*n
**6 + 21*d**6*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*
```

```
d**6*n + 720*d**6) + 120*D*b**2*d**6*x**6*(c + d*x)**n/(d**6*n**6 + 21*d**6
*n**5 + 175*d**6*n**4 + 735*d**6*n**3 + 1624*d**6*n**2 + 1764*d**6*n + 720*
d**6), True))
```

Maxima [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1118 vs. $2(342) = 684$.

Time = 0.24 (sec) , antiderivative size = 1118, normalized size of antiderivative = 3.31

$$\int (a+bx)^2(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right)\, dx = \text{Too large to display}$$

```
[In] integrate((b*x+a)^2*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="maxima")
```

```
[Out] (d^2*(n + 1)*x^2 + c*d*n*x - c^2)*(d*x + c)^n*B*a^2/((n^2 + 3*n + 2)*d^2) +
 2*(d^2*(n + 1)*x^2 + c*d*n*x - c^2)*(d*x + c)^n*A*a*b/((n^2 + 3*n + 2)*d^2
) + (d*x + c)^(n + 1)*A*a^2/(d*(n + 1)) + ((n^2 + 3*n + 2)*d^3*x^3 + (n^2 +
 n)*c*d^2*x^2 - 2*c^2*d*n*x + 2*c^3)*(d*x + c)^n*C*a^2/((n^3 + 6*n^2 + 11*n
 + 6)*d^3) + 2*((n^2 + 3*n + 2)*d^3*x^3 + (n^2 + n)*c*d^2*x^2 - 2*c^2*d*n*x
 + 2*c^3)*(d*x + c)^n*B*a*b/((n^3 + 6*n^2 + 11*n + 6)*d^3) + ((n^2 + 3*n +
2)*d^3*x^3 + (n^2 + n)*c*d^2*x^2 - 2*c^2*d*n*x + 2*c^3)*(d*x + c)^n*A*b^2/(
(n^3 + 6*n^2 + 11*n + 6)*d^3) + ((n^3 + 6*n^2 + 11*n + 6)*d^4*x^4 + (n^3 +
3*n^2 + 2*n)*c*d^3*x^3 - 3*(n^2 + n)*c^2*d^2*x^2 + 6*c^3*d*n*x - 6*c^4)*(d*
x + c)^n*D*a^2/((n^4 + 10*n^3 + 35*n^2 + 50*n + 24)*d^4) + 2*((n^3 + 6*n^2
+ 11*n + 6)*d^4*x^4 + (n^3 + 3*n^2 + 2*n)*c*d^3*x^3 - 3*(n^2 + n)*c^2*d^2*x
^2 + 6*c^3*d*n*x - 6*c^4)*(d*x + c)^n*C*a*b/((n^4 + 10*n^3 + 35*n^2 + 50*n
+ 24)*d^4) + ((n^3 + 6*n^2 + 11*n + 6)*d^4*x^4 + (n^3 + 3*n^2 + 2*n)*c*d^3*
x^3 - 3*(n^2 + n)*c^2*d^2*x^2 + 6*c^3*d*n*x - 6*c^4)*(d*x + c)^n*B*b^2/((n^
4 + 10*n^3 + 35*n^2 + 50*n + 24)*d^4) + 2*((n^4 + 10*n^3 + 35*n^2 + 50*n +
24)*d^5*x^5 + (n^4 + 6*n^3 + 11*n^2 + 6*n)*c*d^4*x^4 - 4*(n^3 + 3*n^2 + 2*n
)*c^2*d^3*x^3 + 12*(n^2 + n)*c^3*d^2*x^2 - 24*c^4*d*n*x + 24*c^5)*(d*x + c)
^n*D*a*b/((n^5 + 15*n^4 + 85*n^3 + 225*n^2 + 274*n + 120)*d^5) + ((n^4 + 10
*n^3 + 35*n^2 + 50*n + 24)*d^5*x^5 + (n^4 + 6*n^3 + 11*n^2 + 6*n)*c*d^4*x^4
 - 4*(n^3 + 3*n^2 + 2*n)*c^2*d^3*x^3 + 12*(n^2 + n)*c^3*d^2*x^2 - 24*c^4*d*
n*x + 24*c^5)*(d*x + c)^n*C*b^2/((n^5 + 15*n^4 + 85*n^3 + 225*n^2 + 274*n +
 120)*d^5) + ((n^5 + 15*n^4 + 85*n^3 + 225*n^2 + 274*n + 120)*d^6*x^6 + (n^
5 + 10*n^4 + 35*n^3 + 50*n^2 + 24*n)*c*d^5*x^5 - 5*(n^4 + 6*n^3 + 11*n^2 +
6*n)*c^2*d^4*x^4 + 20*(n^3 + 3*n^2 + 2*n)*c^3*d^3*x^3 - 60*(n^2 + n)*c^4*d^
2*x^2 + 120*c^5*d*n*x - 120*c^6)*(d*x + c)^n*D*b^2/((n^6 + 21*n^5 + 175*n^4
 + 735*n^3 + 1624*n^2 + 1764*n + 720)*d^6)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 4972 vs. $2(342) = 684$.

Time = 0.34 (sec) , antiderivative size = 4972, normalized size of antiderivative = 14.71

$$\int (a+bx)^2(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right)\,dx = \text{Too large to display}$$

```
[In] integrate((b*x+a)^2*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="giac")
```

```
[Out] ((d*x + c)^n*D*b^2*d^6*n^5*x^6 + (d*x + c)^n*D*b^2*c*d^5*n^5*x^5 + 2*(d*x +
 c)^n*D*a*b*d^6*n^5*x^5 + (d*x + c)^n*C*b^2*d^6*n^5*x^5 + 15*(d*x + c)^n*D*
b^2*d^6*n^4*x^6 + 2*(d*x + c)^n*D*a*b*c*d^5*n^5*x^4 + (d*x + c)^n*C*b^2*c*d
^5*n^5*x^4 + (d*x + c)^n*D*a^2*d^6*n^5*x^4 + 2*(d*x + c)^n*C*a*b*d^6*n^5*x^
4 + (d*x + c)^n*B*b^2*d^6*n^5*x^4 + 10*(d*x + c)^n*D*b^2*c*d^5*n^4*x^5 + 32
*(d*x + c)^n*D*a*b*d^6*n^4*x^5 + 16*(d*x + c)^n*C*b^2*d^6*n^4*x^5 + 85*(d*x
 + c)^n*D*b^2*d^6*n^3*x^6 + (d*x + c)^n*D*a^2*c*d^5*n^5*x^3 + 2*(d*x + c)^n
*C*a*b*c*d^5*n^5*x^3 + (d*x + c)^n*B*b^2*c*d^5*n^5*x^3 + (d*x + c)^n*C*a^2*
d^6*n^5*x^3 + 2*(d*x + c)^n*B*a*b*d^6*n^5*x^3 + (d*x + c)^n*A*b^2*d^6*n^5*x
^3 - 5*(d*x + c)^n*D*b^2*c^2*d^4*n^4*x^4 + 24*(d*x + c)^n*D*a*b*c*d^5*n^4*x
^4 + 12*(d*x + c)^n*C*b^2*c*d^5*n^4*x^4 + 17*(d*x + c)^n*D*a^2*d^6*n^4*x^4
+ 34*(d*x + c)^n*C*a*b*d^6*n^4*x^4 + 17*(d*x + c)^n*B*b^2*d^6*n^4*x^4 + 35*
(d*x + c)^n*D*b^2*c*d^5*n^3*x^5 + 190*(d*x + c)^n*D*a*b*d^6*n^3*x^5 + 95*(d
*x + c)^n*C*b^2*d^6*n^3*x^5 + 225*(d*x + c)^n*D*b^2*d^6*n^2*x^6 + (d*x + c)
^n*C*a^2*c*d^5*n^5*x^2 + 2*(d*x + c)^n*B*a*b*c*d^5*n^5*x^2 + (d*x + c)^n*A*
b^2*c*d^5*n^5*x^2 + (d*x + c)^n*B*a^2*d^6*n^5*x^2 + 2*(d*x + c)^n*A*a*b*d^6
*n^5*x^2 - 8*(d*x + c)^n*D*a*b*c^2*d^4*n^4*x^3 - 4*(d*x + c)^n*C*b^2*c^2*d^
4*n^4*x^3 + 14*(d*x + c)^n*D*a^2*c*d^5*n^4*x^3 + 28*(d*x + c)^n*C*a*b*c*d^5
*n^4*x^3 + 14*(d*x + c)^n*B*b^2*c*d^5*n^4*x^3 + 18*(d*x + c)^n*C*a^2*d^6*n^
4*x^3 + 36*(d*x + c)^n*B*a*b*d^6*n^4*x^3 + 18*(d*x + c)^n*A*b^2*d^6*n^4*x^3
 - 30*(d*x + c)^n*D*b^2*c^2*d^4*n^3*x^4 + 94*(d*x + c)^n*D*a*b*c*d^5*n^3*x^
4 + 47*(d*x + c)^n*C*b^2*c*d^5*n^3*x^4 + 107*(d*x + c)^n*D*a^2*d^6*n^3*x^4
+ 214*(d*x + c)^n*C*a*b*d^6*n^3*x^4 + 107*(d*x + c)^n*B*b^2*d^6*n^3*x^4 + 5
0*(d*x + c)^n*D*b^2*c*d^5*n^2*x^5 + 520*(d*x + c)^n*D*a*b*d^6*n^2*x^5 + 260
*(d*x + c)^n*C*b^2*d^6*n^2*x^5 + 274*(d*x + c)^n*D*b^2*d^6*n*x^6 + (d*x + c
)^n*B*a^2*c*d^5*n^5*x + 2*(d*x + c)^n*A*a*b*c*d^5*n^5*x + (d*x + c)^n*A*a^2
*d^6*n^5*x - 3*(d*x + c)^n*D*a^2*c^2*d^4*n^4*x^2 - 6*(d*x + c)^n*C*a*b*c^2*
d^4*n^4*x^2 - 3*(d*x + c)^n*B*b^2*c^2*d^4*n^4*x^2 + 16*(d*x + c)^n*C*a^2*c*
d^5*n^4*x^2 + 32*(d*x + c)^n*B*a*b*c*d^5*n^4*x^2 + 16*(d*x + c)^n*A*b^2*c*d
^5*n^4*x^2 + 19*(d*x + c)^n*B*a^2*d^6*n^4*x^2 + 38*(d*x + c)^n*A*a*b*d^6*n^
4*x^2 + 20*(d*x + c)^n*D*b^2*c^3*d^3*n^3*x^3 - 72*(d*x + c)^n*D*a*b*c^2*d^4
*n^3*x^3 - 36*(d*x + c)^n*C*b^2*c^2*d^4*n^3*x^3 + 65*(d*x + c)^n*D*a^2*c*d^
5*n^3*x^3 + 130*(d*x + c)^n*C*a*b*c*d^5*n^3*x^3 + 65*(d*x + c)^n*B*b^2*c*d^
5*n^3*x^3 + 121*(d*x + c)^n*C*a^2*d^6*n^3*x^3 + 242*(d*x + c)^n*B*a*b*d^6*n
^3*x^3 + 121*(d*x + c)^n*A*b^2*d^6*n^3*x^3 - 55*(d*x + c)^n*D*b^2*c^2*d^4*n
```

```
^2*x^4 + 144*(d*x + c)^n*D*a*b*c*d^5*n^2*x^4 + 72*(d*x + c)^n*C*b^2*c*d^5*n
^2*x^4 + 307*(d*x + c)^n*D*a^2*d^6*n^2*x^4 + 614*(d*x + c)^n*C*a*b*d^6*n^2*
x^4 + 307*(d*x + c)^n*B*b^2*d^6*n^2*x^4 + 24*(d*x + c)^n*D*b^2*c*d^5*n*x^5
 + 648*(d*x + c)^n*D*a*b*d^6*n*x^5 + 324*(d*x + c)^n*C*b^2*d^6*n*x^5 + 120*(
d*x + c)^n*D*b^2*d^6*x^6 + (d*x + c)^n*A*a^2*c*d^5*n^5 - 2*(d*x + c)^n*C*a^
2*c^2*d^4*n^4*x - 4*(d*x + c)^n*B*a*b*c^2*d^4*n^4*x - 2*(d*x + c)^n*A*b^2*c
^2*d^4*n^4*x + 18*(d*x + c)^n*B*a^2*c*d^5*n^4*x + 36*(d*x + c)^n*A*a*b*c*d^
5*n^4*x + 20*(d*x + c)^n*A*a^2*d^6*n^4*x + 24*(d*x + c)^n*D*a*b*c^3*d^3*n^3
*x^2 + 12*(d*x + c)^n*C*b^2*c^3*d^3*n^3*x^2 - 36*(d*x + c)^n*D*a^2*c^2*d^4*
n^3*x^2 - 72*(d*x + c)^n*C*a*b*c^2*d^4*n^3*x^2 - 36*(d*x + c)^n*B*b^2*c^2*d
^4*n^3*x^2 + 89*(d*x + c)^n*C*a^2*c*d^5*n^3*x^2 + 178*(d*x + c)^n*B*a*b*c*d
^5*n^3*x^2 + 89*(d*x + c)^n*A*b^2*c*d^5*n^3*x^2 + 137*(d*x + c)^n*B*a^2*d^6
*n^3*x^2 + 274*(d*x + c)^n*A*a*b*d^6*n^3*x^2 + 60*(d*x + c)^n*D*b^2*c^3*d^3
*n^2*x^3 - 160*(d*x + c)^n*D*a*b*c^2*d^4*n^2*x^3 - 80*(d*x + c)^n*C*b^2*c^2
*d^4*n^2*x^3 + 112*(d*x + c)^n*D*a^2*c*d^5*n^2*x^3 + 224*(d*x + c)^n*C*a*b*
c*d^5*n^2*x^3 + 112*(d*x + c)^n*B*b^2*c*d^5*n^2*x^3 + 372*(d*x + c)^n*C*a^2
*d^6*n^2*x^3 + 744*(d*x + c)^n*B*a*b*d^6*n^2*x^3 + 372*(d*x + c)^n*A*b^2*d^
6*n^2*x^3 - 30*(d*x + c)^n*D*b^2*c^2*d^4*n*x^4 + 72*(d*x + c)^n*D*a*b*c*d^5
*n*x^4 + 36*(d*x + c)^n*C*b^2*c*d^5*n*x^4 + 396*(d*x + c)^n*D*a^2*d^6*n*x^4
 + 792*(d*x + c)^n*C*a*b*d^6*n*x^4 + 396*(d*x + c)^n*B*b^2*d^6*n*x^4 + 288*
(d*x + c)^n*D*a*b*d^6*x^5 + 144*(d*x + c)^n*C*b^2*d^6*x^5 - (d*x + c)^n*B*a
^2*c^2*d^4*n^4 - 2*(d*x + c)^n*A*a*b*c^2*d^4*n^4 + 20*(d*x + c)^n*A*a^2*c*d
^5*n^4 + 6*(d*x + c)^n*D*a^2*c^3*d^3*n^3*x + 12*(d*x + c)^n*C*a*b*c^3*d^3*n
^3*x + 6*(d*x + c)^n*B*b^2*c^3*d^3*n^3*x - 30*(d*x + c)^n*C*a^2*c^2*d^4*n^3
*x - 60*(d*x + c)^n*B*a*b*c^2*d^4*n^3*x - 30*(d*x + c)^n*A*b^2*c^2*d^4*n^3*
x + 119*(d*x + c)^n*B*a^2*c*d^5*n^3*x + 238*(d*x + c)^n*A*a*b*c*d^5*n^3*x +
 155*(d*x + c)^n*A*a^2*d^6*n^3*x - 60*(d*x + c)^n*D*b^2*c^4*d^2*n^2*x^2 + 1
68*(d*x + c)^n*D*a*b*c^3*d^3*n^2*x^2 + 84*(d*x + c)^n*C*b^2*c^3*d^3*n^2*x^2
 - 123*(d*x + c)^n*D*a^2*c^2*d^4*n^2*x^2 - 246*(d*x + c)^n*C*a*b*c^2*d^4*n^
2*x^2 - 123*(d*x + c)^n*B*b^2*c^2*d^4*n^2*x^2 + 194*(d*x + c)^n*C*a^2*c*d^5
*n^2*x^2 + 388*(d*x + c)^n*B*a*b*c*d^5*n^2*x^2 + 194*(d*x + c)^n*A*b^2*c*d^
5*n^2*x^2 + 461*(d*x + c)^n*B*a^2*d^6*n^2*x^2 + 922*(d*x + c)^n*A*a*b*d^6*n
^2*x^2 + 40*(d*x + c)^n*D*b^2*c^3*d^3*n*x^3 - 96*(d*x + c)^n*D*a*b*c^2*d^4*
n*x^3 - 48*(d*x + c)^n*C*b^2*c^2*d^4*n*x^3 + 60*(d*x + c)^n*D*a^2*c*d^5*n*x
^3 + 120*(d*x + c)^n*C*a*b*c*d^5*n*x^3 + 60*(d*x + c)^n*B*b^2*c*d^5*n*x^3 +
 508*(d*x + c)^n*C*a^2*d^6*n*x^3 + 1016*(d*x + c)^n*B*a*b*d^6*n*x^3 + 508*(
d*x + c)^n*A*b^2*d^6*n*x^3 + 180*(d*x + c)^n*D*a^2*d^6*x^4 + 360*(d*x + c)^
n*C*a*b*d^6*x^4 + 180*(d*x + c)^n*B*b^2*d^6*x^4 + 2*(d*x + c)^n*C*a^2*c^3*d
^3*n^3 + 4*(d*x + c)^n*B*a*b*c^3*d^3*n^3 + 2*(d*x + c)^n*A*b^2*c^3*d^3*n^3
 - 18*(d*x + c)^n*B*a^2*c^2*d^4*n^3 - 36*(d*x + c)^n*A*a*b*c^2*d^4*n^3 + 155
*(d*x + c)^n*A*a^2*c*d^5*n^3 - 48*(d*x + c)^n*D*a*b*c^4*d^2*n^2*x - 24*(d*x
 + c)^n*C*b^2*c^4*d^2*n^2*x + 66*(d*x + c)^n*D*a^2*c^3*d^3*n^2*x + 132*(d*x
 + c)^n*C*a*b*c^3*d^3*n^2*x + 66*(d*x + c)^n*B*b^2*c^3*d^3*n^2*x - 148*(d*x
 + c)^n*C*a^2*c^2*d^4*n^2*x - 296*(d*x + c)^n*B*a*b*c^2*d^4*n^2*x - 148*(d*
x + c)^n*A*b^2*c^2*d^4*n^2*x + 342*(d*x + c)^n*B*a^2*c*d^5*n^2*x + 684*(d*x
```

```
 + c)^n*A*a*b*c*d^5*n^2*x + 580*(d*x + c)^n*A*a^2*d^6*n^2*x - 60*(d*x + c)^
n*D*b^2*c^4*d^2*n*x^2 + 144*(d*x + c)^n*D*a*b*c^3*d^3*n*x^2 + 72*(d*x + c)^
n*C*b^2*c^3*d^3*n*x^2 - 90*(d*x + c)^n*D*a^2*c^2*d^4*n*x^2 - 180*(d*x + c)^
n*C*a*b*c^2*d^4*n*x^2 - 90*(d*x + c)^n*B*b^2*c^2*d^4*n*x^2 + 120*(d*x + c)^
n*C*a^2*c*d^5*n*x^2 + 240*(d*x + c)^n*B*a*b*c*d^5*n*x^2 + 120*(d*x + c)^n*A
*b^2*c*d^5*n*x^2 + 702*(d*x + c)^n*B*a^2*d^6*n*x^2 + 1404*(d*x + c)^n*A*a*b
*d^6*n*x^2 + 240*(d*x + c)^n*C*a^2*d^6*x^3 + 480*(d*x + c)^n*B*a*b*d^6*x^3
+ 240*(d*x + c)^n*A*b^2*d^6*x^3 - 6*(d*x + c)^n*D*a^2*c^4*d^2*n^2 - 12*(d*x
 + c)^n*C*a*b*c^4*d^2*n^2 - 6*(d*x + c)^n*B*b^2*c^4*d^2*n^2 + 30*(d*x + c)^
n*C*a^2*c^3*d^3*n^2 + 60*(d*x + c)^n*B*a*b*c^3*d^3*n^2 + 30*(d*x + c)^n*A*b
^2*c^3*d^3*n^2 - 119*(d*x + c)^n*B*a^2*c^2*d^4*n^2 - 238*(d*x + c)^n*A*a*b*
c^2*d^4*n^2 + 580*(d*x + c)^n*A*a^2*c*d^5*n^2 + 120*(d*x + c)^n*D*b^2*c^5*d
*n*x - 288*(d*x + c)^n*D*a*b*c^4*d^2*n*x - 144*(d*x + c)^n*C*b^2*c^4*d^2*n*
x + 180*(d*x + c)^n*D*a^2*c^3*d^3*n*x + 360*(d*x + c)^n*C*a*b*c^3*d^3*n*x +
 180*(d*x + c)^n*B*b^2*c^3*d^3*n*x - 240*(d*x + c)^n*C*a^2*c^2*d^4*n*x - 48
0*(d*x + c)^n*B*a*b*c^2*d^4*n*x - 240*(d*x + c)^n*A*b^2*c^2*d^4*n*x + 360*(
d*x + c)^n*B*a^2*c*d^5*n*x + 720*(d*x + c)^n*A*a*b*c*d^5*n*x + 1044*(d*x +
c)^n*A*a^2*d^6*n*x + 360*(d*x + c)^n*B*a^2*d^6*x^2 + 720*(d*x + c)^n*A*a*b*
d^6*x^2 + 48*(d*x + c)^n*D*a*b*c^5*d*n + 24*(d*x + c)^n*C*b^2*c^5*d*n - 66*
(d*x + c)^n*D*a^2*c^4*d^2*n - 132*(d*x + c)^n*C*a*b*c^4*d^2*n - 66*(d*x + c
)^n*B*b^2*c^4*d^2*n + 148*(d*x + c)^n*C*a^2*c^3*d^3*n + 296*(d*x + c)^n*B*a
*b*c^3*d^3*n + 148*(d*x + c)^n*A*b^2*c^3*d^3*n - 342*(d*x + c)^n*B*a^2*c^2*
d^4*n - 684*(d*x + c)^n*A*a*b*c^2*d^4*n + 1044*(d*x + c)^n*A*a^2*c*d^5*n +
720*(d*x + c)^n*A*a^2*d^6*x - 120*(d*x + c)^n*D*b^2*c^6 + 288*(d*x + c)^n*D
*a*b*c^5*d + 144*(d*x + c)^n*C*b^2*c^5*d - 180*(d*x + c)^n*D*a^2*c^4*d^2 -
360*(d*x + c)^n*C*a*b*c^4*d^2 - 180*(d*x + c)^n*B*b^2*c^4*d^2 + 240*(d*x +
c)^n*C*a^2*c^3*d^3 + 480*(d*x + c)^n*B*a*b*c^3*d^3 + 240*(d*x + c)^n*A*b^2*
c^3*d^3 - 360*(d*x + c)^n*B*a^2*c^2*d^4 - 720*(d*x + c)^n*A*a*b*c^2*d^4 + 7
20*(d*x + c)^n*A*a^2*c*d^5)/(d^6*n^6 + 21*d^6*n^5 + 175*d^6*n^4 + 735*d^6*n
^3 + 1624*d^6*n^2 + 1764*d^6*n + 720*d^6)
```

Mupad [F(-1)]

Timed out.

$$\int (a+bx)^2(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx$$
$$=\int (a+b\,x)^2\,(c+d\,x)^n\,\left(A+B\,x+C\,x^2+x^3\,D\right)\,dx$$

```
[In] int((a + b*x)^2*(c + d*x)^n*(A + B*x + C*x^2 + x^3*D),x)
[Out] int((a + b*x)^2*(c + d*x)^n*(A + B*x + C*x^2 + x^3*D), x)
```

3.27 $\int (a+bx)(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) dx$

Optimal result

Integrand size = 28, antiderivative size = 226

$$\begin{aligned}
\int (a+bx)(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) dx
&= -\frac{(bc-ad)\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)(c+dx)^{1+n}}{d^5(1+n)} \\
&\quad - \frac{\left(ad(2cCd-Bd^2-3c^2D)-b(3c^2Cd-2Bcd^2+Ad^3-4c^3D)\right)(c+dx)^{2+n}}{d^5(2+n)} \\
&\quad + \frac{\left(ad(Cd-3cD)-b(3cCd-Bd^2-6c^2D)\right)(c+dx)^{3+n}}{d^5(3+n)} \\
&\quad + \frac{(bCd-4bcD+adD)(c+dx)^{4+n}}{d^5(4+n)} + \frac{bD(c+dx)^{5+n}}{d^5(5+n)}
\end{aligned}$$

```
[Out] -(-a*d+b*c)*(A*d^3-B*c*d^2+C*c^2*d-D*c^3)*(d*x+c)^(1+n)/d^5/(1+n)-(a*d*(-B*
d^2+2*C*c*d-3*D*c^2)-b*(A*d^3-2*B*c*d^2+3*C*c^2*d-4*D*c^3))*(d*x+c)^(2+n)/d
^5/(2+n)+(a*d*(C*d-3*D*c)-b*(-B*d^2+3*C*c*d-6*D*c^2))*(d*x+c)^(3+n)/d^5/(3+
n)+(C*b*d+D*a*d-4*D*b*c)*(d*x+c)^(4+n)/d^5/(4+n)+b*D*(d*x+c)^(5+n)/d^5/(5+n
)
```

Rubi [A] (verified)

Time = 0.11 (sec) , antiderivative size = 226, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.036, Rules used

$= \{1634\}$

$$\begin{aligned}\int (a+bx)(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx &= -\frac{(bc-ad)(c+dx)^{n+1}\left(Ad^3-Bcd^2+c^3(-D)+c^2Cd\right)}{d^5(n+1)}\\ &\quad-\frac{(c+dx)^{n+2}\left(ad(-Bd^2-3c^2D+2cCd)-b(Ad^3-2Bcd^2-4c^3D+3c^2Cd)\right)}{d^5(n+2)}\\ &\quad+\frac{(c+dx)^{n+3}\left(ad(Cd-3cD)-b(-Bd^2-6c^2D+3cCd)\right)}{d^5(n+3)}\\ &\quad+\frac{(c+dx)^{n+4}(adD-4bcD+bCd)}{d^5(n+4)}+\frac{bD(c+dx)^{n+5}}{d^5(n+5)}\end{aligned}$$

```
[In] Int[(a + b*x)*(c + d*x)^n*(A + B*x + C*x^2 + D*x^3),x]
```

```
[Out] -(((b*c - a*d)*(c^2*C*d - B*c*d^2 + A*d^3 - c^3*D)*(c + d*x)^(1 + n))/(d^5*
(1 + n))) - ((a*d*(2*c*C*d - B*d^2 - 3*c^2*D) - b*(3*c^2*C*d - 2*B*c*d^2 +
A*d^3 - 4*c^3*D))*(c + d*x)^(2 + n))/(d^5*(2 + n)) + ((a*d*(C*d - 3*c*D) -
b*(3*c*C*d - B*d^2 - 6*c^2*D))*(c + d*x)^(3 + n))/(d^5*(3 + n)) + ((b*C*d -
 4*b*c*D + a*d*D)*(c + d*x)^(4 + n))/(d^5*(4 + n)) + (b*D*(c + d*x)^(5 + n)
)/(d^5*(5 + n))
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

$$\begin{aligned}\text{integral} = \int \Bigg(&\frac{(-bc+ad)\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)(c+dx)^n}{d^4}\\ &+\frac{(-ad(2cCd-Bd^2-3c^2D)+b(3c^2Cd-2Bcd^2+Ad^3-4c^3D))\,(c+dx)^{1+n}}{d^4}\\ &+\frac{(ad(Cd-3cD)-b(3cCd-Bd^2-6c^2D))\,(c+dx)^{2+n}}{d^4}\\ &+\frac{(bCd-4bcD+adD)(c+dx)^{3+n}}{d^4}+\frac{bD(c+dx)^{4+n}}{d^4}\Bigg)\,dx\end{aligned}$$

$$
\begin{aligned}
&= -\frac{(bc-ad)\left(c^2Cd-Bcd^2+Ad^3-c^3D\right)(c+dx)^{1+n}}{d^5(1+n)} \\
&\quad - \frac{\left(ad(2cCd-Bd^2-3c^2D)-b(3c^2Cd-2Bcd^2+Ad^3-4c^3D)\right)(c+dx)^{2+n}}{d^5(2+n)} \\
&\quad + \frac{\left(ad(Cd-3cD)-b(3cCd-Bd^2-6c^2D)\right)(c+dx)^{3+n}}{d^5(3+n)} \\
&\quad + \frac{(bCd-4bcD+adD)(c+dx)^{4+n}}{d^5(4+n)} + \frac{bD(c+dx)^{5+n}}{d^5(5+n)}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.44 (sec) , antiderivative size = 199, normalized size of antiderivative = 0.88

$$
\int (a+bx)(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx
$$

$$
= \frac{(c+dx)^{1+n}\left(\frac{(bc-ad)(-c^2Cd+Bcd^2-Ad^3+c^3D)}{1+n}+\frac{(ad(-2cCd+Bd^2+3c^2D)+b(3c^2Cd-2Bcd^2+Ad^3-4c^3D))(c+dx)}{2+n}+ (ad(Cd-3cL\right.}{d^5}
$$

```
[In] Integrate[(a + b*x)*(c + d*x)^n*(A + B*x + C*x^2 + D*x^3),x]
```

```
[Out] ((c + d*x)^(1 + n)*(((b*c - a*d)*(-(c^2*C*d) + B*c*d^2 - A*d^3 + c^3*D))/(1
 + n) + ((a*d*(-2*c*C*d + B*d^2 + 3*c^2*D) + b*(3*c^2*C*d - 2*B*c*d^2 + A*d
^3 - 4*c^3*D))*(c + d*x))/(2 + n) + ((a*d*(C*d - 3*c*D) + b*(-3*c*C*d + B*d
^2 + 6*c^2*D))*(c + d*x)^2)/(3 + n) + ((b*C*d - 4*b*c*D + a*d*D)*(c + d*x)^
3)/(4 + n) + (b*D*(c + d*x)^4)/(5 + n)))/d^5
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 963 vs. 2(226) = 452.

Time = 1.66 (sec) , antiderivative size = 964, normalized size of antiderivative = 4.27

method	result
norman	$\frac{bDx^5 e^{n\ln(dx+c)}}{5+n}+\frac{c(Aa\,d^4n^4+14Aa\,d^4n^3-Abc\,d^3n^3-Bac\,d^3n^3+71Aa\,d^4n^2-12Abc\,d^3n^2-12Bac\,d^3n^2+2Bb\,c^2d^2n^2+2Ca}{}$
gosper	Expression too large to display
parallelrisch	Expression too large to display

```
[In] int((b*x+a)*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x,method=_RETURNVERBOSE)
```

```
[Out] b*D/(5+n)*x^5*exp(n*ln(d*x+c))+c*(A*a*d^4*n^4+14*A*a*d^4*n^3-A*b*c*d^3*n^3-
B*a*c*d^3*n^3+71*A*a*d^4*n^2-12*A*b*c*d^3*n^2-12*B*a*c*d^3*n^2+2*B*b*c^2*d^
2*n^2+2*C*a*c^2*d^2*n^2+154*A*a*d^4*n-47*A*b*c*d^3*n-47*B*a*c*d^3*n+18*B*b*
c^2*d^2*n+18*C*a*c^2*d^2*n-6*C*b*c^3*d*n-6*D*a*c^3*d*n+120*A*a*d^4-60*A*b*c
```

```
*d^3-60*B*a*c*d^3+40*B*b*c^2*d^2+40*C*a*c^2*d^2-30*C*b*c^3*d-30*D*a*c^3*d+2
4*D*b*c^4)/d^5/(n^5+15*n^4+85*n^3+225*n^2+274*n+120)*exp(n*ln(d*x+c))+(C*b*
d*n+D*a*d*n+D*b*c*n+5*C*b*d+5*D*a*d)/d/(n^2+9*n+20)*x^4*exp(n*ln(d*x+c))+(B
*b*d^2*n^2+C*a*d^2*n^2+C*b*c*d*n^2+D*a*c*d*n^2+9*B*b*d^2*n+9*C*a*d^2*n+5*C*
b*c*d*n+5*D*a*c*d*n-4*D*b*c^2*n+20*B*b*d^2+20*C*a*d^2)/d^2/(n^3+12*n^2+47*n
+60)*x^3*exp(n*ln(d*x+c))+(A*b*d^3*n^3+B*a*d^3*n^3+B*b*c*d^2*n^3+C*a*c*d^2*
n^3+12*A*b*d^3*n^2+12*B*a*d^3*n^2+9*B*b*c*d^2*n^2+9*C*a*c*d^2*n^2-3*C*b*c^2
*d*n^2-3*D*a*c^2*d*n^2+47*A*b*d^3*n+47*B*a*d^3*n+20*B*b*c*d^2*n+20*C*a*c*d^
2*n-15*C*b*c^2*d*n-15*D*a*c^2*d*n+12*D*b*c^3*n+60*A*b*d^3+60*B*a*d^3)/d^3/(
n^4+14*n^3+71*n^2+154*n+120)*x^2*exp(n*ln(d*x+c))+(A*a*d^4*n^4+A*b*c*d^3*n^
4+B*a*c*d^3*n^4+14*A*a*d^4*n^3+12*A*b*c*d^3*n^3+12*B*a*c*d^3*n^3-2*B*b*c^2*
d^2*n^3-2*C*a*c^2*d^2*n^3+71*A*a*d^4*n^2+47*A*b*c*d^3*n^2+47*B*a*c*d^3*n^2-
18*B*b*c^2*d^2*n^2-18*C*a*c^2*d^2*n^2+6*C*b*c^3*d*n^2+6*D*a*c^3*d*n^2+154*A
*a*d^4*n+60*A*b*c*d^3*n+60*B*a*c*d^3*n-40*B*b*c^2*d^2*n-40*C*a*c^2*d^2*n+30
*C*b*c^3*d*n+30*D*a*c^3*d*n-24*D*b*c^4*n+120*A*a*d^4)/d^4/(n^5+15*n^4+85*n^
3+225*n^2+274*n+120)*x*exp(n*ln(d*x+c))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 988 vs. 2(228) = 456.

Time = 0.27 (sec) , antiderivative size = 988, normalized size of antiderivative = 4.37

$$\int (a+bx)(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right)\,dx$$
$$= \frac{(Aacd^4n^4+24\,Dbc^5+120\,Aacd^4+40\,(Ca+Bb)c^3d^2-60\,(Ba+Ab)c^2d^3+(Dbd^5n^4+10\,Dbd^5n^3+35\,D}{}$$

```
[In] integrate((b*x+a)*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="fricas")
```

```
[Out] (A*a*c*d^4*n^4 + 24*D*b*c^5 + 120*A*a*c*d^4 + 40*(C*a + B*b)*c^3*d^2 - 60*(
B*a + A*b)*c^2*d^3 + (D*b*d^5*n^4 + 10*D*b*d^5*n^3 + 35*D*b*d^5*n^2 + 50*D*
b*d^5*n + 24*D*b*d^5)*x^5 + (30*(D*a + C*b)*d^5 + (D*b*c*d^4 + (D*a + C*b)*
d^5)*n^4 + (6*D*b*c*d^4 + 11*(D*a + C*b)*d^5)*n^3 + (11*D*b*c*d^4 + 41*(D*a
 + C*b)*d^5)*n^2 + (6*D*b*c*d^4 + 61*(D*a + C*b)*d^5)*n)*x^4 + (14*A*a*c*d^
4 - (B*a + A*b)*c^2*d^3)*n^3 + (40*(C*a + B*b)*d^5 + ((C*a + B*b)*d^5 + (D*
a*c + C*b*c)*d^4)*n^4 - 4*(D*b*c^2*d^3 - 3*(C*a + B*b)*d^5 - 2*(D*a*c + C*b
*c)*d^4)*n^3 - (12*D*b*c^2*d^3 - 49*(C*a + B*b)*d^5 - 17*(D*a*c + C*b*c)*d^
4)*n^2 - 2*(4*D*b*c^2*d^3 - 39*(C*a + B*b)*d^5 - 5*(D*a*c + C*b*c)*d^4)*n)*
x^3 + (71*A*a*c*d^4 + 2*(C*a + B*b)*c^3*d^2 - 12*(B*a + A*b)*c^2*d^3)*n^2 +
 (60*(B*a + A*b)*d^5 + ((C*a + B*b)*c*d^4 + (B*a + A*b)*d^5)*n^4 + (10*(C*a
 + B*b)*c*d^4 + 13*(B*a + A*b)*d^5 - 3*(D*a*c^2 + C*b*c^2)*d^3)*n^3 + (12*D
*b*c^3*d^2 + 29*(C*a + B*b)*c*d^4 + 59*(B*a + A*b)*d^5 - 18*(D*a*c^2 + C*b*
c^2)*d^3)*n^2 + (12*D*b*c^3*d^2 + 20*(C*a + B*b)*c*d^4 + 107*(B*a + A*b)*d^
5 - 15*(D*a*c^2 + C*b*c^2)*d^3)*n)*x^2 - 30*(D*a*c^4 + C*b*c^4)*d + (154*A*
a*c*d^4 + 18*(C*a + B*b)*c^3*d^2 - 47*(B*a + A*b)*c^2*d^3 - 6*(D*a*c^4 + C*
```

```
b*c^4)*d)*n + (120*A*a*d^5 + (A*a*d^5 + (B*a + A*b)*c*d^4)*n^4 + 2*(7*A*a*d
^5 - (C*a + B*b)*c^2*d^3 + 6*(B*a + A*b)*c*d^4)*n^3 + (71*A*a*d^5 - 18*(C*a
 + B*b)*c^2*d^3 + 47*(B*a + A*b)*c*d^4 + 6*(D*a*c^3 + C*b*c^3)*d^2)*n^2 - 2
*(12*D*b*c^4*d - 77*A*a*d^5 + 20*(C*a + B*b)*c^2*d^3 - 30*(B*a + A*b)*c*d^4
 - 15*(D*a*c^3 + C*b*c^3)*d^2)*n)*x)*(d*x + c)^n/(d^5*n^5 + 15*d^5*n^4 + 85
*d^5*n^3 + 225*d^5*n^2 + 274*d^5*n + 120*d^5)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 13522 vs. $2(211) = 422$.

Time = 2.68 (sec) , antiderivative size = 13522, normalized size of antiderivative = 59.83

$$\int (a+bx)(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right)\,dx = \text{Too large to display}$$

```
[In] integrate((b*x+a)*(d*x+c)**n*(D*x**3+C*x**2+B*x+A),x)
```

```
[Out] Piecewise((c**n*(A*a*x + A*b*x**2/2 + B*a*x**2/2 + B*b*x**3/3 + C*a*x**3/3
+ C*b*x**4/4 + D*a*x**4/4 + D*b*x**5/5), Eq(d, 0)), (-3*A*a*d**4/(12*c**4*d
**5 + 48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) -
 A*b*c*d**3/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*
x**3 + 12*d**9*x**4) - 4*A*b*d**4*x/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**
2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) - B*a*c*d**3/(12*c**4*d**5 + 4
8*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) - 4*B*a*
d**4*x/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3
+ 12*d**9*x**4) - B*b*c**2*d**2/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**2*d*
*7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) - 4*B*b*c*d**3*x/(12*c**4*d**5 + 4
8*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) - 6*B*b*
d**4*x**2/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x*
*3 + 12*d**9*x**4) - C*a*c**2*d**2/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**2
*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) - 4*C*a*c*d**3*x/(12*c**4*d**5
+ 48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) - 6*C
*a*d**4*x**2/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8
*x**3 + 12*d**9*x**4) - 3*C*b*c**3*d/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c*
*2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) - 12*C*b*c**2*d**2*x/(12*c**4
*d**5 + 48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4)
 - 18*C*b*c*d**3*x**2/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**2*d**7*x**2 +
48*c*d**8*x**3 + 12*d**9*x**4) - 12*C*b*d**4*x**3/(12*c**4*d**5 + 48*c**3*d
**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) - 3*D*a*c**3*d/(
12*c**4*d**5 + 48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**
9*x**4) - 12*D*a*c**2*d**2*x/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**2*d**7*
x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) - 18*D*a*c*d**3*x**2/(12*c**4*d**5 +
48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) - 12*D*
a*d**4*x**3/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*
x**3 + 12*d**9*x**4) + 12*D*b*c**4*log(c/d + x)/(12*c**4*d**5 + 48*c**3*d**
```

```
6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) + 25*D*b*c**4/(12*
c**4*d**5 + 48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x
**4) + 48*D*b*c**3*d*x*log(c/d + x)/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**
2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) + 88*D*b*c**3*d*x/(12*c**4*d**
5 + 48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) + 7
2*D*b*c**2*d**2*x**2*log(c/d + x)/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**2*
d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) + 108*D*b*c**2*d**2*x**2/(12*c**
4*d**5 + 48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4
) + 48*D*b*c*d**3*x**3*log(c/d + x)/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**
2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4) + 48*D*b*c*d**3*x**3/(12*c**4*
d**5 + 48*c**3*d**6*x + 72*c**2*d**7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4)
+ 12*D*b*d**4*x**4*log(c/d + x)/(12*c**4*d**5 + 48*c**3*d**6*x + 72*c**2*d*
*7*x**2 + 48*c*d**8*x**3 + 12*d**9*x**4), Eq(n, -5)), (-2*A*a*d**4/(6*c**3*
d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) - A*b*c*d**3/(6*c**3*
d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) - 3*A*b*d**4*x/(6*c**
3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) - B*a*c*d**3/(6*c**
3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) - 3*B*a*d**4*x/(6*c
**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) - 2*B*b*c**2*d**2
/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) - 6*B*b*c*d*
*3*x/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) - 6*B*b*
d**4*x**2/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) - 2
*C*a*c**2*d**2/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3
) - 6*C*a*c*d**3*x/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*
x**3) - 6*C*a*d**4*x**2/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*
d**8*x**3) + 6*C*b*c**3*d*log(c/d + x)/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c
*d**7*x**2 + 6*d**8*x**3) + 11*C*b*c**3*d/(6*c**3*d**5 + 18*c**2*d**6*x + 1
8*c*d**7*x**2 + 6*d**8*x**3) + 18*C*b*c**2*d**2*x*log(c/d + x)/(6*c**3*d**5
 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) + 27*C*b*c**2*d**2*x/(6*c
**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) + 18*C*b*c*d**3*x
**2*log(c/d + x)/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x*
*3) + 18*C*b*c*d**3*x**2/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6
*d**8*x**3) + 6*C*b*d**4*x**3*log(c/d + x)/(6*c**3*d**5 + 18*c**2*d**6*x +
18*c*d**7*x**2 + 6*d**8*x**3) + 6*D*a*c**3*d*log(c/d + x)/(6*c**3*d**5 + 18
*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) + 11*D*a*c**3*d/(6*c**3*d**5 +
 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) + 18*D*a*c**2*d**2*x*log(c/
d + x)/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) + 27*D
*a*c**2*d**2*x/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3
) + 18*D*a*c*d**3*x**2*log(c/d + x)/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d*
*7*x**2 + 6*d**8*x**3) + 18*D*a*c*d**3*x**2/(6*c**3*d**5 + 18*c**2*d**6*x +
 18*c*d**7*x**2 + 6*d**8*x**3) + 6*D*a*d**4*x**3*log(c/d + x)/(6*c**3*d**5
+ 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) - 24*D*b*c**4*log(c/d + x)
/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) - 44*D*b*c**
4/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) - 72*D*b*c*
*3*d*x*log(c/d + x)/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8
*x**3) - 108*D*b*c**3*d*x/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 +
```

```
6*d**8*x**3) - 72*D*b*c**2*d**2*x**2*log(c/d + x)/(6*c**3*d**5 + 18*c**2*d*
*6*x + 18*c*d**7*x**2 + 6*d**8*x**3) - 72*D*b*c**2*d**2*x**2/(6*c**3*d**5 +
 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) - 24*D*b*c*d**3*x**3*log(c/
d + x)/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3) + 6*D*
b*d**4*x**4/(6*c**3*d**5 + 18*c**2*d**6*x + 18*c*d**7*x**2 + 6*d**8*x**3),
Eq(n, -4)), (-A*a*d**4/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) - A*b*c*d**
3/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) - 2*A*b*d**4*x/(2*c**2*d**5 + 4*
c*d**6*x + 2*d**7*x**2) - B*a*c*d**3/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**
2) - 2*B*a*d**4*x/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) + 2*B*b*c**2*d**
2*log(c/d + x)/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) + 3*B*b*c**2*d**2/(
2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) + 4*B*b*c*d**3*x*log(c/d + x)/(2*c*
*2*d**5 + 4*c*d**6*x + 2*d**7*x**2) + 4*B*b*c*d**3*x/(2*c**2*d**5 + 4*c*d**
6*x + 2*d**7*x**2) + 2*B*b*d**4*x**2*log(c/d + x)/(2*c**2*d**5 + 4*c*d**6*x
 + 2*d**7*x**2) + 2*C*a*c**2*d**2*log(c/d + x)/(2*c**2*d**5 + 4*c*d**6*x +
2*d**7*x**2) + 3*C*a*c**2*d**2/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) + 4
*C*a*c*d**3*x*log(c/d + x)/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) + 4*C*a
*c*d**3*x/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) + 2*C*a*d**4*x**2*log(c/
d + x)/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) - 6*C*b*c**3*d*log(c/d + x)
/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) - 9*C*b*c**3*d/(2*c**2*d**5 + 4*c
*d**6*x + 2*d**7*x**2) - 12*C*b*c**2*d**2*x*log(c/d + x)/(2*c**2*d**5 + 4*c
*d**6*x + 2*d**7*x**2) - 12*C*b*c**2*d**2*x/(2*c**2*d**5 + 4*c*d**6*x + 2*d
**7*x**2) - 6*C*b*c*d**3*x**2*log(c/d + x)/(2*c**2*d**5 + 4*c*d**6*x + 2*d*
*7*x**2) + 2*C*b*d**4*x**3/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) - 6*D*a
*c**3*d*log(c/d + x)/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) - 9*D*a*c**3*
d/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) - 12*D*a*c**2*d**2*x*log(c/d + x
)/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) - 12*D*a*c**2*d**2*x/(2*c**2*d**
5 + 4*c*d**6*x + 2*d**7*x**2) - 6*D*a*c*d**3*x**2*log(c/d + x)/(2*c**2*d**5
 + 4*c*d**6*x + 2*d**7*x**2) + 2*D*a*d**4*x**3/(2*c**2*d**5 + 4*c*d**6*x +
2*d**7*x**2) + 12*D*b*c**4*log(c/d + x)/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*
x**2) + 18*D*b*c**4/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) + 24*D*b*c**3*
d*x*log(c/d + x)/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) + 24*D*b*c**3*d*x
/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) + 12*D*b*c**2*d**2*x**2*log(c/d +
 x)/(2*c**2*d**5 + 4*c*d**6*x + 2*d**7*x**2) - 4*D*b*c*d**3*x**3/(2*c**2*d*
*5 + 4*c*d**6*x + 2*d**7*x**2) + D*b*d**4*x**4/(2*c**2*d**5 + 4*c*d**6*x +
2*d**7*x**2), Eq(n, -3)), (-6*A*a*d**4/(6*c*d**5 + 6*d**6*x) + 6*A*b*c*d**3
*log(c/d + x)/(6*c*d**5 + 6*d**6*x) + 6*A*b*c*d**3/(6*c*d**5 + 6*d**6*x) +
6*A*b*d**4*x*log(c/d + x)/(6*c*d**5 + 6*d**6*x) + 6*B*a*c*d**3*log(c/d + x)
/(6*c*d**5 + 6*d**6*x) + 6*B*a*c*d**3/(6*c*d**5 + 6*d**6*x) + 6*B*a*d**4*x*
log(c/d + x)/(6*c*d**5 + 6*d**6*x) - 12*B*b*c**2*d**2*log(c/d + x)/(6*c*d**
5 + 6*d**6*x) - 12*B*b*c**2*d**2/(6*c*d**5 + 6*d**6*x) - 12*B*b*c*d**3*x*lo
g(c/d + x)/(6*c*d**5 + 6*d**6*x) + 6*B*b*d**4*x**2/(6*c*d**5 + 6*d**6*x) -
12*C*a*c**2*d**2*log(c/d + x)/(6*c*d**5 + 6*d**6*x) - 12*C*a*c**2*d**2/(6*c
*d**5 + 6*d**6*x) - 12*C*a*c*d**3*x*log(c/d + x)/(6*c*d**5 + 6*d**6*x) + 6*
C*a*d**4*x**2/(6*c*d**5 + 6*d**6*x) + 18*C*b*c**3*d*log(c/d + x)/(6*c*d**5
+ 6*d**6*x) + 18*C*b*c**3*d/(6*c*d**5 + 6*d**6*x) + 18*C*b*c**2*d**2*x*log(
```

```
c/d + x)/(6*c*d**5 + 6*d**6*x) - 9*C*b*c*d**3*x**2/(6*c*d**5 + 6*d**6*x) +
3*C*b*d**4*x**3/(6*c*d**5 + 6*d**6*x) + 18*D*a*c**3*d*log(c/d + x)/(6*c*d**
5 + 6*d**6*x) + 18*D*a*c**3*d/(6*c*d**5 + 6*d**6*x) + 18*D*a*c**2*d**2*x*lo
g(c/d + x)/(6*c*d**5 + 6*d**6*x) - 9*D*a*c*d**3*x**2/(6*c*d**5 + 6*d**6*x)
+ 3*D*a*d**4*x**3/(6*c*d**5 + 6*d**6*x) - 24*D*b*c**4*log(c/d + x)/(6*c*d**
5 + 6*d**6*x) - 24*D*b*c**4/(6*c*d**5 + 6*d**6*x) - 24*D*b*c**3*d*x*log(c/d
 + x)/(6*c*d**5 + 6*d**6*x) + 12*D*b*c**2*d**2*x**2/(6*c*d**5 + 6*d**6*x) -
 4*D*b*c*d**3*x**3/(6*c*d**5 + 6*d**6*x) + 2*D*b*d**4*x**4/(6*c*d**5 + 6*d*
*6*x), Eq(n, -2)), (A*a*log(c/d + x)/d - A*b*c*log(c/d + x)/d**2 + A*b*x/d
- B*a*c*log(c/d + x)/d**2 + B*a*x/d + B*b*c**2*log(c/d + x)/d**3 - B*b*c*x/
d**2 + B*b*x**2/(2*d) + C*a*c**2*log(c/d + x)/d**3 - C*a*c*x/d**2 + C*a*x**
2/(2*d) - C*b*c**3*log(c/d + x)/d**4 + C*b*c**2*x/d**3 - C*b*c*x**2/(2*d**2
) + C*b*x**3/(3*d) - D*a*c**3*log(c/d + x)/d**4 + D*a*c**2*x/d**3 - D*a*c*x
**2/(2*d**2) + D*a*x**3/(3*d) + D*b*c**4*log(c/d + x)/d**5 - D*b*c**3*x/d**
4 + D*b*c**2*x**2/(2*d**3) - D*b*c*x**3/(3*d**2) + D*b*x**4/(4*d), Eq(n, -1
)), (A*a*c*d**4*n**4*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3
+ 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 14*A*a*c*d**4*n**3*(c + d*x)**n/
(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120
*d**5) + 71*A*a*c*d**4*n**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**
5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 154*A*a*c*d**4*n*(c + d*x
)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n
+ 120*d**5) + 120*A*a*c*d**4*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d*
*5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + A*a*d**5*n**4*x*(c + d*x
)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n
+ 120*d**5) + 14*A*a*d**5*n**3*x*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 8
5*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 71*A*a*d**5*n**2*x*(
c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*
d**5*n + 120*d**5) + 154*A*a*d**5*n*x*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**
4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 120*A*a*d**5*x*
(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274
*d**5*n + 120*d**5) - A*b*c**2*d**3*n**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*
n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) - 12*A*b*c**2*
d**3*n**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*
n**2 + 274*d**5*n + 120*d**5) - 47*A*b*c**2*d**3*n*(c + d*x)**n/(d**5*n**5
+ 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) - 60
*A*b*c**2*d**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*
d**5*n**2 + 274*d**5*n + 120*d**5) + A*b*c*d**4*n**4*x*(c + d*x)**n/(d**5*n
**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5)
+ 12*A*b*c*d**4*n**3*x*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**
3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 47*A*b*c*d**4*n**2*x*(c + d*x)
**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n +
 120*d**5) + 60*A*b*c*d**4*n*x*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*
d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + A*b*d**5*n**4*x**2*(c
+ d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d*
*5*n + 120*d**5) + 13*A*b*d**5*n**3*x**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*
```

```
n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 59*A*b*d**5*
n**2*x**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*
n**2 + 274*d**5*n + 120*d**5) + 107*A*b*d**5*n*x**2*(c + d*x)**n/(d**5*n**5
 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 6
0*A*b*d**5*x**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225
*d**5*n**2 + 274*d**5*n + 120*d**5) - B*a*c**2*d**3*n**3*(c + d*x)**n/(d**5
*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5
) - 12*B*a*c**2*d**3*n**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*
n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) - 47*B*a*c**2*d**3*n*(c + d*x
)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n
 + 120*d**5) - 60*B*a*c**2*d**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*
d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + B*a*c*d**4*n**4*x*(c +
 d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**
5*n + 120*d**5) + 12*B*a*c*d**4*n**3*x*(c + d*x)**n/(d**5*n**5 + 15*d**5*n*
*4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 47*B*a*c*d**4*
n**2*x*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**
2 + 274*d**5*n + 120*d**5) + 60*B*a*c*d**4*n*x*(c + d*x)**n/(d**5*n**5 + 15
*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + B*a*d*
*5*n**4*x**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d*
*5*n**2 + 274*d**5*n + 120*d**5) + 13*B*a*d**5*n**3*x**2*(c + d*x)**n/(d**5
*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5
) + 59*B*a*d**5*n**2*x**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*
n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 107*B*a*d**5*n*x**2*(c + d*
x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n
 + 120*d**5) + 60*B*a*d**5*x**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85
*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 2*B*b*c**3*d**2*n**2*
(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274
*d**5*n + 120*d**5) + 18*B*b*c**3*d**2*n*(c + d*x)**n/(d**5*n**5 + 15*d**5*
n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 40*B*b*c**3*
d**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2
 + 274*d**5*n + 120*d**5) - 2*B*b*c**2*d**3*n**3*x*(c + d*x)**n/(d**5*n**5 +
 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) - 18*
B*b*c**2*d**3*n**2*x*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3
 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) - 40*B*b*c**2*d**3*n*x*(c + d*x)**
n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 1
20*d**5) + B*b*c*d**4*n**4*x**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85
*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 10*B*b*c*d**4*n**3*x*
*2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 +
274*d**5*n + 120*d**5) + 29*B*b*c*d**4*n**2*x**2*(c + d*x)**n/(d**5*n**5 +
15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 20*B
*b*c*d**4*n*x**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 22
5*d**5*n**2 + 274*d**5*n + 120*d**5) + B*b*d**5*n**4*x**3*(c + d*x)**n/(d**
5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**
5) + 12*B*b*d**5*n**3*x**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5
*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 49*B*b*d**5*n**2*x**3*(c +
```

```
 d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**
5*n + 120*d**5) + 78*B*b*d**5*n*x**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4
 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 40*B*b*d**5*x**3
*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 27
4*d**5*n + 120*d**5) + 2*C*a*c**3*d**2*n**2*(c + d*x)**n/(d**5*n**5 + 15*d*
*5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 18*C*a*c*
*3*d**2*n*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*
n**2 + 274*d**5*n + 120*d**5) + 40*C*a*c**3*d**2*(c + d*x)**n/(d**5*n**5 +
15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) - 2*C*
a*c**2*d**3*n**3*x*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 +
225*d**5*n**2 + 274*d**5*n + 120*d**5) - 18*C*a*c**2*d**3*n**2*x*(c + d*x)*
*n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n +
120*d**5) - 40*C*a*c**2*d**3*n*x*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 8
5*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + C*a*c*d**4*n**4*x**2
*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 27
4*d**5*n + 120*d**5) + 10*C*a*c*d**4*n**3*x**2*(c + d*x)**n/(d**5*n**5 + 15
*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 29*C*a
*c*d**4*n**2*x**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 2
25*d**5*n**2 + 274*d**5*n + 120*d**5) + 20*C*a*c*d**4*n*x**2*(c + d*x)**n/(
d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*
d**5) + C*a*d**5*n**4*x**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5
*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 12*C*a*d**5*n**3*x**3*(c +
 d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**
5*n + 120*d**5) + 49*C*a*d**5*n**2*x**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*n
**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 78*C*a*d**5*n
*x**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2
 + 274*d**5*n + 120*d**5) + 40*C*a*d**5*x**3*(c + d*x)**n/(d**5*n**5 + 15*d
**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) - 6*C*b*c*
*4*d*n*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**
2 + 274*d**5*n + 120*d**5) - 30*C*b*c**4*d*(c + d*x)**n/(d**5*n**5 + 15*d**
5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 6*C*b*c**3
*d**2*n**2*x*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d*
*5*n**2 + 274*d**5*n + 120*d**5) + 30*C*b*c**3*d**2*n*x*(c + d*x)**n/(d**5*
n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5)
 - 3*C*b*c**2*d**3*n**3*x**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d*
*5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) - 18*C*b*c**2*d**3*n**2*x*
*2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 +
274*d**5*n + 120*d**5) - 15*C*b*c**2*d**3*n*x**2*(c + d*x)**n/(d**5*n**5 +
15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + C*b*
c*d**4*n**4*x**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 22
5*d**5*n**2 + 274*d**5*n + 120*d**5) + 8*C*b*c*d**4*n**3*x**3*(c + d*x)**n/
(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120
*d**5) + 17*C*b*c*d**4*n**2*x**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 8
5*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 10*C*b*c*d**4*n*x**3
*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 27
```

```
4*d**5*n + 120*d**5) + C*b*d**5*n**4*x**4*(c + d*x)**n/(d**5*n**5 + 15*d**5
*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 11*C*b*d**5
*n**3*x**4*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5
*n**2 + 274*d**5*n + 120*d**5) + 41*C*b*d**5*n**2*x**4*(c + d*x)**n/(d**5*n
**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5)
+ 61*C*b*d**5*n*x**4*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3
+ 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 30*C*b*d**5*x**4*(c + d*x)**n/(d
**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d
**5) - 6*D*a*c**4*d*n*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3
 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) - 30*D*a*c**4*d*(c + d*x)**n/(d**
5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**
5) + 6*D*a*c**3*d**2*n**2*x*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**
5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 30*D*a*c**3*d**2*n*x*(c +
 d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**
5*n + 120*d**5) - 3*D*a*c**2*d**3*n**3*x**2*(c + d*x)**n/(d**5*n**5 + 15*d*
*5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) - 18*D*a*c*
*2*d**3*n**2*x**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 2
25*d**5*n**2 + 274*d**5*n + 120*d**5) - 15*D*a*c**2*d**3*n*x**2*(c + d*x)**
n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 1
20*d**5) + D*a*c*d**4*n**4*x**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85
*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 8*D*a*c*d**4*n**3*x**
3*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 2
74*d**5*n + 120*d**5) + 17*D*a*c*d**4*n**2*x**3*(c + d*x)**n/(d**5*n**5 + 1
5*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 10*D*
a*c*d**4*n*x**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225
*d**5*n**2 + 274*d**5*n + 120*d**5) + D*a*d**5*n**4*x**4*(c + d*x)**n/(d**5
*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5
) + 11*D*a*d**5*n**3*x**4*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*
n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 41*D*a*d**5*n**2*x**4*(c +
d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5
*n + 120*d**5) + 61*D*a*d**5*n*x**4*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4
+ 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 30*D*a*d**5*x**4*
(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274
*d**5*n + 120*d**5) + 24*D*b*c**5*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 +
85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) - 24*D*b*c**4*d*n*x*(
c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*
d**5*n + 120*d**5) + 12*D*b*c**3*d**2*n**2*x**2*(c + d*x)**n/(d**5*n**5 + 1
5*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 12*D*
b*c**3*d**2*n*x**2*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 +
225*d**5*n**2 + 274*d**5*n + 120*d**5) - 4*D*b*c**2*d**3*n**3*x**3*(c + d*x
)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n
+ 120*d**5) - 12*D*b*c**2*d**3*n**2*x**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*
n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) - 8*D*b*c**2*d
**3*n*x**3*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5
*n**2 + 274*d**5*n + 120*d**5) + D*b*c*d**4*n**4*x**4*(c + d*x)**n/(d**5*n*
```

```
*5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) +
 6*D*b*c*d**4*n**3*x**4*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n*
*3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 11*D*b*c*d**4*n**2*x**4*(c +
d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5
*n + 120*d**5) + 6*D*b*c*d**4*n*x**4*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4
 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + D*b*d**5*n**4*x*
*5*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 +
274*d**5*n + 120*d**5) + 10*D*b*d**5*n**3*x**5*(c + d*x)**n/(d**5*n**5 + 15
*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5) + 35*D*b
*d**5*n**2*x**5*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225
*d**5*n**2 + 274*d**5*n + 120*d**5) + 50*D*b*d**5*n*x**5*(c + d*x)**n/(d**5
*n**5 + 15*d**5*n**4 + 85*d**5*n**3 + 225*d**5*n**2 + 274*d**5*n + 120*d**5
) + 24*D*b*d**5*x**5*(c + d*x)**n/(d**5*n**5 + 15*d**5*n**4 + 85*d**5*n**3
+ 225*d**5*n**2 + 274*d**5*n + 120*d**5), True))
```

Maxima [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 596 vs. 2(228) = 456.

Time = 0.22 (sec) , antiderivative size = 596, normalized size of antiderivative = 2.64

$$\int (a+bx)(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right)\,dx$$
$$= \frac{(d^2(n+1)x^2+cdnx-c^2)(dx+c)^n Ba}{(n^2+3n+2)d^2} + \frac{(d^2(n+1)x^2+cdnx-c^2)(dx+c)^n Ab}{(n^2+3n+2)d^2}$$
$$+ \frac{(dx+c)^{n+1}Aa}{d(n+1)} + \frac{((n^2+3n+2)d^3x^3+(n^2+n)cd^2x^2-2c^2dnx+2c^3)(dx+c)^n Ca}{(n^3+6n^2+11n+6)d^3}$$
$$+ \frac{((n^2+3n+2)d^3x^3+(n^2+n)cd^2x^2-2c^2dnx+2c^3)(dx+c)^n Bb}{(n^3+6n^2+11n+6)d^3}$$
$$+ \frac{((n^3+6n^2+11n+6)d^4x^4+(n^3+3n^2+2n)cd^3x^3-3(n^2+n)c^2d^2x^2+6c^3dnx-6c^4)(dx+c)^n Da}{(n^4+10n^3+35n^2+50n+24)d^4}$$
$$+ \frac{((n^3+6n^2+11n+6)d^4x^4+(n^3+3n^2+2n)cd^3x^3-3(n^2+n)c^2d^2x^2+6c^3dnx-6c^4)(dx+c)^n Cb}{(n^4+10n^3+35n^2+50n+24)d^4}$$
$$+ \frac{((n^4+10n^3+35n^2+50n+24)d^5x^5+(n^4+6n^3+11n^2+6n)cd^4x^4-4(n^3+3n^2+2n)c^2d^3x^3+12}{(n^5+15n^4+85n^3+225n^2+274n+120)d^5}$$

```
[In] integrate((b*x+a)*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="maxima")
```

```
[Out] (d^2*(n + 1)*x^2 + c*d*n*x - c^2)*(d*x + c)^n*B*a/((n^2 + 3*n + 2)*d^2) + (
d^2*(n + 1)*x^2 + c*d*n*x - c^2)*(d*x + c)^n*A*b/((n^2 + 3*n + 2)*d^2) + (d
*x + c)^(n + 1)*A*a/(d*(n + 1)) + ((n^2 + 3*n + 2)*d^3*x^3 + (n^2 + n)*c*d^
2*x^2 - 2*c^2*d*n*x + 2*c^3)*(d*x + c)^n*C*a/((n^3 + 6*n^2 + 11*n + 6)*d^3)
 + ((n^2 + 3*n + 2)*d^3*x^3 + (n^2 + n)*c*d^2*x^2 - 2*c^2*d*n*x + 2*c^3)*(d
*x + c)^n*B*b/((n^3 + 6*n^2 + 11*n + 6)*d^3) + ((n^3 + 6*n^2 + 11*n + 6)*d^
```

```
4*x^4 + (n^3 + 3*n^2 + 2*n)*c*d^3*x^3 - 3*(n^2 + n)*c^2*d^2*x^2 + 6*c^3*d*n
*x - 6*c^4)*(d*x + c)^n*D*a/((n^4 + 10*n^3 + 35*n^2 + 50*n + 24)*d^4) + ((n
^3 + 6*n^2 + 11*n + 6)*d^4*x^4 + (n^3 + 3*n^2 + 2*n)*c*d^3*x^3 - 3*(n^2 + n
)*c^2*d^2*x^2 + 6*c^3*d*n*x - 6*c^4)*(d*x + c)^n*C*b/((n^4 + 10*n^3 + 35*n^
2 + 50*n + 24)*d^4) + ((n^4 + 10*n^3 + 35*n^2 + 50*n + 24)*d^5*x^5 + (n^4 +
 6*n^3 + 11*n^2 + 6*n)*c*d^4*x^4 - 4*(n^3 + 3*n^2 + 2*n)*c^2*d^3*x^3 + 12*(
n^2 + n)*c^3*d^2*x^2 - 24*c^4*d*n*x + 24*c^5)*(d*x + c)^n*D*b/((n^5 + 15*n^
4 + 85*n^3 + 225*n^2 + 274*n + 120)*d^5)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2224 vs. 2(228) = 456.

Time = 0.31 (sec) , antiderivative size = 2224, normalized size of antiderivative = 9.84

$$\int (a+bx)(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx = \text{Too large to display}$$

```
[In] integrate((b*x+a)*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="giac")
```

```
[Out] ((d*x + c)^n*D*b*d^5*n^4*x^5 + (d*x + c)^n*D*b*c*d^4*n^4*x^4 + (d*x + c)^n*
D*a*d^5*n^4*x^4 + (d*x + c)^n*C*b*d^5*n^4*x^4 + 10*(d*x + c)^n*D*b*d^5*n^3*
x^5 + (d*x + c)^n*D*a*c*d^4*n^4*x^3 + (d*x + c)^n*C*b*c*d^4*n^4*x^3 + (d*x
+ c)^n*C*a*d^5*n^4*x^3 + (d*x + c)^n*B*b*d^5*n^4*x^3 + 6*(d*x + c)^n*D*b*c*
d^4*n^3*x^4 + 11*(d*x + c)^n*D*a*d^5*n^3*x^4 + 11*(d*x + c)^n*C*b*d^5*n^3*x
^4 + 35*(d*x + c)^n*D*b*d^5*n^2*x^5 + (d*x + c)^n*C*a*c*d^4*n^4*x^2 + (d*x
+ c)^n*B*b*c*d^4*n^4*x^2 + (d*x + c)^n*B*a*d^5*n^4*x^2 + (d*x + c)^n*A*b*d^
5*n^4*x^2 - 4*(d*x + c)^n*D*b*c^2*d^3*n^3*x^3 + 8*(d*x + c)^n*D*a*c*d^4*n^3
*x^3 + 8*(d*x + c)^n*C*b*c*d^4*n^3*x^3 + 12*(d*x + c)^n*C*a*d^5*n^3*x^3 + 1
2*(d*x + c)^n*B*b*d^5*n^3*x^3 + 11*(d*x + c)^n*D*b*c*d^4*n^2*x^4 + 41*(d*x
+ c)^n*D*a*d^5*n^2*x^4 + 41*(d*x + c)^n*C*b*d^5*n^2*x^4 + 50*(d*x + c)^n*D*
b*d^5*n*x^5 + (d*x + c)^n*B*a*c*d^4*n^4*x + (d*x + c)^n*A*b*c*d^4*n^4*x + (
d*x + c)^n*A*a*d^5*n^4*x - 3*(d*x + c)^n*D*a*c^2*d^3*n^3*x^2 - 3*(d*x + c)^
n*C*b*c^2*d^3*n^3*x^2 + 10*(d*x + c)^n*C*a*c*d^4*n^3*x^2 + 10*(d*x + c)^n*B
*b*c*d^4*n^3*x^2 + 13*(d*x + c)^n*B*a*d^5*n^3*x^2 + 13*(d*x + c)^n*A*b*d^5*
n^3*x^2 - 12*(d*x + c)^n*D*b*c^2*d^3*n^2*x^3 + 17*(d*x + c)^n*D*a*c*d^4*n^2
*x^3 + 17*(d*x + c)^n*C*b*c*d^4*n^2*x^3 + 49*(d*x + c)^n*C*a*d^5*n^2*x^3 +
49*(d*x + c)^n*B*b*d^5*n^2*x^3 + 6*(d*x + c)^n*D*b*c*d^4*n*x^4 + 61*(d*x +
c)^n*D*a*d^5*n*x^4 + 61*(d*x + c)^n*C*b*d^5*n*x^4 + 24*(d*x + c)^n*D*b*d^5*
x^5 + (d*x + c)^n*A*a*c*d^4*n^4 - 2*(d*x + c)^n*C*a*c^2*d^3*n^3*x - 2*(d*x
+ c)^n*B*b*c^2*d^3*n^3*x + 12*(d*x + c)^n*B*a*c*d^4*n^3*x + 12*(d*x + c)^n*
A*b*c*d^4*n^3*x + 14*(d*x + c)^n*A*a*d^5*n^3*x + 12*(d*x + c)^n*D*b*c^3*d^2
*n^2*x^2 - 18*(d*x + c)^n*D*a*c^2*d^3*n^2*x^2 - 18*(d*x + c)^n*C*b*c^2*d^3*
n^2*x^2 + 29*(d*x + c)^n*C*a*c*d^4*n^2*x^2 + 29*(d*x + c)^n*B*b*c*d^4*n^2*x
^2 + 59*(d*x + c)^n*B*a*d^5*n^2*x^2 + 59*(d*x + c)^n*A*b*d^5*n^2*x^2 - 8*(d
*x + c)^n*D*b*c^2*d^3*n*x^3 + 10*(d*x + c)^n*D*a*c*d^4*n*x^3 + 10*(d*x + c)
```

```
^n*C*b*c*d^4*n*x^3 + 78*(d*x + c)^n*C*a*d^5*n*x^3 + 78*(d*x + c)^n*B*b*d^5*
n*x^3 + 30*(d*x + c)^n*D*a*d^5*x^4 + 30*(d*x + c)^n*C*b*d^5*x^4 - (d*x + c)
^n*B*a*c^2*d^3*n^3 - (d*x + c)^n*A*b*c^2*d^3*n^3 + 14*(d*x + c)^n*A*a*c*d^4
*n^3 + 6*(d*x + c)^n*D*a*c^3*d^2*n^2*x + 6*(d*x + c)^n*C*b*c^3*d^2*n^2*x -
18*(d*x + c)^n*C*a*c^2*d^3*n^2*x - 18*(d*x + c)^n*B*b*c^2*d^3*n^2*x + 47*(d
*x + c)^n*B*a*c*d^4*n^2*x + 47*(d*x + c)^n*A*b*c*d^4*n^2*x + 71*(d*x + c)^n
*A*a*d^5*n^2*x + 12*(d*x + c)^n*D*b*c^3*d^2*n*x^2 - 15*(d*x + c)^n*D*a*c^2*
d^3*n*x^2 - 15*(d*x + c)^n*C*b*c^2*d^3*n*x^2 + 20*(d*x + c)^n*C*a*c*d^4*n*x
^2 + 20*(d*x + c)^n*B*b*c*d^4*n*x^2 + 107*(d*x + c)^n*B*a*d^5*n*x^2 + 107*(
d*x + c)^n*A*b*d^5*n*x^2 + 40*(d*x + c)^n*C*a*d^5*x^3 + 40*(d*x + c)^n*B*b*
d^5*x^3 + 2*(d*x + c)^n*C*a*c^3*d^2*n^2 + 2*(d*x + c)^n*B*b*c^3*d^2*n^2 - 1
2*(d*x + c)^n*B*a*c^2*d^3*n^2 - 12*(d*x + c)^n*A*b*c^2*d^3*n^2 + 71*(d*x +
c)^n*A*a*c*d^4*n^2 - 24*(d*x + c)^n*D*b*c^4*d*n*x + 30*(d*x + c)^n*D*a*c^3*
d^2*n*x + 30*(d*x + c)^n*C*b*c^3*d^2*n*x - 40*(d*x + c)^n*C*a*c^2*d^3*n*x -
 40*(d*x + c)^n*B*b*c^2*d^3*n*x + 60*(d*x + c)^n*B*a*c*d^4*n*x + 60*(d*x +
c)^n*A*b*c*d^4*n*x + 154*(d*x + c)^n*A*a*d^5*n*x + 60*(d*x + c)^n*B*a*d^5*x
^2 + 60*(d*x + c)^n*A*b*d^5*x^2 - 6*(d*x + c)^n*D*a*c^4*d*n - 6*(d*x + c)^n
*C*b*c^4*d*n + 18*(d*x + c)^n*C*a*c^3*d^2*n + 18*(d*x + c)^n*B*b*c^3*d^2*n
- 47*(d*x + c)^n*B*a*c^2*d^3*n - 47*(d*x + c)^n*A*b*c^2*d^3*n + 154*(d*x +
c)^n*A*a*c*d^4*n + 120*(d*x + c)^n*A*a*d^5*x + 24*(d*x + c)^n*D*b*c^5 - 30*
(d*x + c)^n*D*a*c^4*d - 30*(d*x + c)^n*C*b*c^4*d + 40*(d*x + c)^n*C*a*c^3*d
^2 + 40*(d*x + c)^n*B*b*c^3*d^2 - 60*(d*x + c)^n*B*a*c^2*d^3 - 60*(d*x + c)
^n*A*b*c^2*d^3 + 120*(d*x + c)^n*A*a*c*d^4)/(d^5*n^5 + 15*d^5*n^4 + 85*d^5*
n^3 + 225*d^5*n^2 + 274*d^5*n + 120*d^5)
```

Mupad [F(-1)]

Timed out.

$$\int (a+bx)(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx$$
$$=\int (a+b\,x)\,\left(c+d\,x\right)^n\,\left(A+B\,x+C\,x^2+x^3\,D\right)\,dx$$

```
[In] int((a + b*x)*(c + d*x)^n*(A + B*x + C*x^2 + x^3*D),x)
[Out] int((a + b*x)*(c + d*x)^n*(A + B*x + C*x^2 + x^3*D), x)
```

3.28 $\int (c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) dx$

Optimal result

Integrand size = 23, antiderivative size = 126

$$\int (c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) dx = \frac{(c^2Cd-Bcd^2+Ad^3-c^3D)(c+dx)^{1+n}}{d^4(1+n)} - \frac{(2cCd-Bd^2-3c^2D)(c+dx)^{2+n}}{d^4(2+n)} + \frac{(Cd-3cD)(c+dx)^{3+n}}{d^4(3+n)} + \frac{D(c+dx)^{4+n}}{d^4(4+n)}$$

```
[Out] (A*d^3-B*c*d^2+C*c^2*d-D*c^3)*(d*x+c)^(1+n)/d^4/(1+n)-(-B*d^2+2*C*c*d-3*D*c
^2)*(d*x+c)^(2+n)/d^4/(2+n)+(C*d-3*D*c)*(d*x+c)^(3+n)/d^4/(3+n)+D*(d*x+c)^(
4+n)/d^4/(4+n)
```

Rubi [A] (verified)

Time = 0.05 (sec) , antiderivative size = 126, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.043, Rules used = {1864}

$$\int (c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) dx = \frac{(c+dx)^{n+1}(Ad^3-Bcd^2+c^3(-D)+c^2Cd)}{d^4(n+1)} - \frac{(c+dx)^{n+2}(-Bd^2-3c^2D+2cCd)}{d^4(n+2)} + \frac{(Cd-3cD)(c+dx)^{n+3}}{d^4(n+3)} + \frac{D(c+dx)^{n+4}}{d^4(n+4)}$$

```
[In] Int[(c + d*x)^n*(A + B*x + C*x^2 + D*x^3),x]
```

```
[Out] ((c^2*C*d - B*c*d^2 + A*d^3 - c^3*D)*(c + d*x)^(1 + n))/(d^4*(1 + n)) - ((2
*c*C*d - B*d^2 - 3*c^2*D)*(c + d*x)^(2 + n))/(d^4*(2 + n)) + ((C*d - 3*c*D)
*(c + d*x)^(3 + n))/(d^4*(3 + n)) + (D*(c + d*x)^(4 + n))/(d^4*(4 + n))
```

Rule 1864

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_.))^(p_.), x_Symbol] :> Int[ExpandIntegrand
[Pq*(a + b*x^n)^p, x], x] /; FreeQ[{a, b, n}, x] && PolyQ[Pq, x] && (IGtQ[p
, 0] || EqQ[n, 1])
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \left(\frac{(c^2Cd - Bcd^2 + Ad^3 - c^3D)\,(c+dx)^n}{d^3} + \frac{(-2cCd + Bd^2 + 3c^2D)\,(c+dx)^{1+n}}{d^3}\right.\\ &\qquad \left. + \frac{(Cd - 3cD)(c+dx)^{2+n}}{d^3} + \frac{D(c+dx)^{3+n}}{d^3}\right) dx\\ &= \frac{(c^2Cd - Bcd^2 + Ad^3 - c^3D)\,(c+dx)^{1+n}}{d^4(1+n)} - \frac{(2cCd - Bd^2 - 3c^2D)\,(c+dx)^{2+n}}{d^4(2+n)}\\ &\quad + \frac{(Cd - 3cD)(c+dx)^{3+n}}{d^4(3+n)} + \frac{D(c+dx)^{4+n}}{d^4(4+n)}\end{aligned}$$

Mathematica [A] (verified)

Time = 0.14 (sec) , antiderivative size = 108, normalized size of antiderivative = 0.86

$$\int (c+dx)^n \left(A + Bx + Cx^2 + Dx^3\right) dx$$

$$= \frac{(c+dx)^{1+n}\left(\frac{c^2Cd - Bcd^2 + Ad^3 - c^3D}{1+n} + \frac{(-2cCd + Bd^2 + 3c^2D)(c+dx)}{2+n} + \frac{(Cd-3cD)(c+dx)^2}{3+n} + \frac{D(c+dx)^3}{4+n}\right)}{d^4}$$

```
[In] Integrate[(c + d*x)^n*(A + B*x + C*x^2 + D*x^3),x]
```

```
[Out] ((c + d*x)^(1 + n)*((c^2*C*d - B*c*d^2 + A*d^3 - c^3*D)/(1 + n) + ((-2*c*C*
d + B*d^2 + 3*c^2*D)*(c + d*x))/(2 + n) + ((C*d - 3*c*D)*(c + d*x)^2)/(3 +
n) + (D*(c + d*x)^3)/(4 + n)))/d^4
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 307 vs. 2(126) = 252.

Time = 1.61 (sec) , antiderivative size = 308, normalized size of antiderivative = 2.44

method	result
gosper	$\frac{(dx+c)^{1+n}(Dd^3n^3x^3+C\,d^3n^3x^2+6Dd^3n^2x^3+B\,d^3n^3x+7C\,d^3n^2x^2-3Dc\,d^2n^2x^2+11Dd^3n\,x^3+A\,d^3n^3+8B\,d^3n^2x-2Cc\,d^2n}{}$
norman	$\frac{Dx^4\mathrm{e}^{n\ln(dx+c)}}{4+n}+\frac{c\left(A\,d^3n^3+9A\,d^3n^2-Bc\,d^2n^2+26A\,d^3n-7Bc\,d^2n+2C\,c^2dn+24A\,d^3-12Bc\,d^2+8C\,c^2d-6Dc^3\right)\mathrm{e}^{n\ln(dx+c)}}{d^4(n^4+10n^3+35n^2+50n+24)}$
parallelrisch	$\frac{B\,x^2(dx+c)^nc\,d^4n^3+Bx(dx+c)^nc^2d^3n^3+14C\,x^3(dx+c)^nc\,d^4n+5C\,x^2(dx+c)^nc^2d^3n^2+2Dx^3(dx+c)^nc^2d^3n-3Dx^2(dx+c)^n}{}$

```
[In] int((d*x+c)^n*(D*x^3+C*x^2+B*x+A),x,method=_RETURNVERBOSE)
```

```
[Out] 1/d^4*(d*x+c)^(1+n)/(n^4+10*n^3+35*n^2+50*n+24)*(D*d^3*n^3*x^3+C*d^3*n^3*x^
2+6*D*d^3*n^2*x^3+B*d^3*n^3*x+7*C*d^3*n^2*x^2-3*D*c*d^2*n^2*x^2+11*D*d^3*n*
x^3+A*d^3*n^3+8*B*d^3*n^2*x-2*C*c*d^2*n^2*x+14*C*d^3*n*x^2-9*D*c*d^2*n*x^2+
6*D*d^3*x^3+9*A*d^3*n^2-B*c*d^2*n^2+19*B*d^3*n*x-10*C*c*d^2*n*x+8*C*d^3*x^2
+6*D*c^2*d*n*x-6*D*c*d^2*x^2+26*A*d^3*n-7*B*c*d^2*n+12*B*d^3*x+2*C*c^2*d*n-
8*C*c*d^2*x+6*D*c^2*d*x+24*A*d^3-12*B*c*d^2+8*C*c^2*d-6*D*c^3)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 394 vs. 2(127) = 254.

Time = 0.27 (sec) , antiderivative size = 394, normalized size of antiderivative = 3.13

$$\int (c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx$$
$$=\frac{(Acd^3n^3-6\,Dc^4+8\,Cc^3d-12\,Bc^2d^2+24\,Acd^3+(Dd^4n^3+6\,Dd^4n^2+11\,Dd^4n+6\,Dd^4)x^4+(8\,Cd^4+}{}$$

```
[In] integrate((d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="fricas")
```

```
[Out] (A*c*d^3*n^3 - 6*D*c^4 + 8*C*c^3*d - 12*B*c^2*d^2 + 24*A*c*d^3 + (D*d^4*n^3
 + 6*D*d^4*n^2 + 11*D*d^4*n + 6*D*d^4)*x^4 + (8*C*d^4 + (D*c*d^3 + C*d^4)*n
^3 + (3*D*c*d^3 + 7*C*d^4)*n^2 + 2*(D*c*d^3 + 7*C*d^4)*n)*x^3 - (B*c^2*d^2
- 9*A*c*d^3)*n^2 + (12*B*d^4 + (C*c*d^3 + B*d^4)*n^3 - (3*D*c^2*d^2 - 5*C*c
*d^3 - 8*B*d^4)*n^2 - (3*D*c^2*d^2 - 4*C*c*d^3 - 19*B*d^4)*n)*x^2 + (2*C*c^
3*d - 7*B*c^2*d^2 + 26*A*c*d^3)*n + (24*A*d^4 + (B*c*d^3 + A*d^4)*n^3 - (2*
C*c^2*d^2 - 7*B*c*d^3 - 9*A*d^4)*n^2 + 2*(3*D*c^3*d - 4*C*c^2*d^2 + 6*B*c*d
^3 + 13*A*d^4)*n)*x)*(d*x + c)^n/(d^4*n^4 + 10*d^4*n^3 + 35*d^4*n^2 + 50*d^
4*n + 24*d^4)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 3798 vs. $2(112) = 224$.

Time = 1.15 (sec) , antiderivative size = 3798, normalized size of antiderivative = 30.14

$$\int (c+dx)^n \left(A+Bx+Cx^2+Dx^3\right)\,dx = \text{Too large to display}$$

```
[In] integrate((d*x+c)**n*(D*x**3+C*x**2+B*x+A),x)
```

```
[Out] Piecewise((c**n*(A*x + B*x**2/2 + C*x**3/3 + D*x**4/4), Eq(d, 0)), (-2*A*d*
*3/(6*c**3*d**4 + 18*c**2*d**5*x + 18*c*d**6*x**2 + 6*d**7*x**3) - B*c*d**2
/(6*c**3*d**4 + 18*c**2*d**5*x + 18*c*d**6*x**2 + 6*d**7*x**3) - 3*B*d**3*x
/(6*c**3*d**4 + 18*c**2*d**5*x + 18*c*d**6*x**2 + 6*d**7*x**3) - 2*C*c**2*d
/(6*c**3*d**4 + 18*c**2*d**5*x + 18*c*d**6*x**2 + 6*d**7*x**3) - 6*C*c*d**2
*x/(6*c**3*d**4 + 18*c**2*d**5*x + 18*c*d**6*x**2 + 6*d**7*x**3) - 6*C*d**3
*x**2/(6*c**3*d**4 + 18*c**2*d**5*x + 18*c*d**6*x**2 + 6*d**7*x**3) + 6*D*c
**3*log(c/d + x)/(6*c**3*d**4 + 18*c**2*d**5*x + 18*c*d**6*x**2 + 6*d**7*x*
*3) + 11*D*c**3/(6*c**3*d**4 + 18*c**2*d**5*x + 18*c*d**6*x**2 + 6*d**7*x**
3) + 18*D*c**2*d*x*log(c/d + x)/(6*c**3*d**4 + 18*c**2*d**5*x + 18*c*d**6*x
**2 + 6*d**7*x**3) + 27*D*c**2*d*x/(6*c**3*d**4 + 18*c**2*d**5*x + 18*c*d**
6*x**2 + 6*d**7*x**3) + 18*D*c*d**2*x**2*log(c/d + x)/(6*c**3*d**4 + 18*c**
2*d**5*x + 18*c*d**6*x**2 + 6*d**7*x**3) + 18*D*c*d**2*x**2/(6*c**3*d**4 +
18*c**2*d**5*x + 18*c*d**6*x**2 + 6*d**7*x**3) + 6*D*d**3*x**3*log(c/d + x)
/(6*c**3*d**4 + 18*c**2*d**5*x + 18*c*d**6*x**2 + 6*d**7*x**3), Eq(n, -4)),
 (-A*d**3/(2*c**2*d**4 + 4*c*d**5*x + 2*d**6*x**2) - B*c*d**2/(2*c**2*d**4
+ 4*c*d**5*x + 2*d**6*x**2) - 2*B*d**3*x/(2*c**2*d**4 + 4*c*d**5*x + 2*d**6
*x**2) + 2*C*c**2*d*log(c/d + x)/(2*c**2*d**4 + 4*c*d**5*x + 2*d**6*x**2) +
 3*C*c**2*d/(2*c**2*d**4 + 4*c*d**5*x + 2*d**6*x**2) + 4*C*c*d**2*x*log(c/d
 + x)/(2*c**2*d**4 + 4*c*d**5*x + 2*d**6*x**2) + 4*C*c*d**2*x/(2*c**2*d**4
+ 4*c*d**5*x + 2*d**6*x**2) + 2*C*d**3*x**2*log(c/d + x)/(2*c**2*d**4 + 4*c
*d**5*x + 2*d**6*x**2) - 6*D*c**3*log(c/d + x)/(2*c**2*d**4 + 4*c*d**5*x +
2*d**6*x**2) - 9*D*c**3/(2*c**2*d**4 + 4*c*d**5*x + 2*d**6*x**2) - 12*D*c**
2*d*x*log(c/d + x)/(2*c**2*d**4 + 4*c*d**5*x + 2*d**6*x**2) - 12*D*c**2*d*x
/(2*c**2*d**4 + 4*c*d**5*x + 2*d**6*x**2) - 6*D*c*d**2*x**2*log(c/d + x)/(2
*c**2*d**4 + 4*c*d**5*x + 2*d**6*x**2) + 2*D*d**3*x**3/(2*c**2*d**4 + 4*c*d
**5*x + 2*d**6*x**2), Eq(n, -3)), (-2*A*d**3/(2*c*d**4 + 2*d**5*x) + 2*B*c*
d**2*log(c/d + x)/(2*c*d**4 + 2*d**5*x) + 2*B*c*d**2/(2*c*d**4 + 2*d**5*x)
+ 2*B*d**3*x*log(c/d + x)/(2*c*d**4 + 2*d**5*x) - 4*C*c**2*d*log(c/d + x)/(
2*c*d**4 + 2*d**5*x) - 4*C*c**2*d/(2*c*d**4 + 2*d**5*x) - 4*C*c*d**2*x*log(
c/d + x)/(2*c*d**4 + 2*d**5*x) + 2*C*d**3*x**2/(2*c*d**4 + 2*d**5*x) + 6*D*
c**3*log(c/d + x)/(2*c*d**4 + 2*d**5*x) + 6*D*c**3/(2*c*d**4 + 2*d**5*x) +
6*D*c**2*d*x*log(c/d + x)/(2*c*d**4 + 2*d**5*x) - 3*D*c*d**2*x**2/(2*c*d**4
 + 2*d**5*x) + D*d**3*x**3/(2*c*d**4 + 2*d**5*x), Eq(n, -2)), (A*log(c/d +
x)/d - B*c*log(c/d + x)/d**2 + B*x/d + C*c**2*log(c/d + x)/d**3 - C*c*x/d**
```

```
2 + C*x**2/(2*d) - D*c**3*log(c/d + x)/d**4 + D*c**2*x/d**3 - D*c*x**2/(2*d
**2) + D*x**3/(3*d), Eq(n, -1)), (A*c*d**3*n**3*(c + d*x)**n/(d**4*n**4 + 1
0*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 9*A*c*d**3*n**2*(c + d*
x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 26*
A*c*d**3*n*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*
n + 24*d**4) + 24*A*c*d**3*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4
*n**2 + 50*d**4*n + 24*d**4) + A*d**4*n**3*x*(c + d*x)**n/(d**4*n**4 + 10*d
**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 9*A*d**4*n**2*x*(c + d*x)*
*n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 26*A*d
**4*n*x*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n +
 24*d**4) + 24*A*d**4*x*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n*
*2 + 50*d**4*n + 24*d**4) - B*c**2*d**2*n**2*(c + d*x)**n/(d**4*n**4 + 10*d
**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) - 7*B*c**2*d**2*n*(c + d*x)*
*n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) - 12*B*c
**2*d**2*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n
+ 24*d**4) + B*c*d**3*n**3*x*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d*
*4*n**2 + 50*d**4*n + 24*d**4) + 7*B*c*d**3*n**2*x*(c + d*x)**n/(d**4*n**4
+ 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 12*B*c*d**3*n*x*(c +
 d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) +
B*d**4*n**3*x**2*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50
*d**4*n + 24*d**4) + 8*B*d**4*n**2*x**2*(c + d*x)**n/(d**4*n**4 + 10*d**4*n
**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 19*B*d**4*n*x**2*(c + d*x)**n/(
d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 12*B*d**4*
x**2*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24
*d**4) + 2*C*c**3*d*n*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2
 + 50*d**4*n + 24*d**4) + 8*C*c**3*d*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3
 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) - 2*C*c**2*d**2*n**2*x*(c + d*x)**n/
(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) - 8*C*c**2*
d**2*n*x*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n
+ 24*d**4) + C*c*d**3*n**3*x**2*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35
*d**4*n**2 + 50*d**4*n + 24*d**4) + 5*C*c*d**3*n**2*x**2*(c + d*x)**n/(d**4
*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 4*C*c*d**3*n*x
**2*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*
d**4) + C*d**4*n**3*x**3*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n
**2 + 50*d**4*n + 24*d**4) + 7*C*d**4*n**2*x**3*(c + d*x)**n/(d**4*n**4 + 1
0*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 14*C*d**4*n*x**3*(c + d
*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 8*
C*d**4*x**3*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4
*n + 24*d**4) - 6*D*c**4*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n
**2 + 50*d**4*n + 24*d**4) + 6*D*c**3*d*n*x*(c + d*x)**n/(d**4*n**4 + 10*d*
*4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) - 3*D*c**2*d**2*n**2*x**2*(c
+ d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) -
 3*D*c**2*d**2*n*x**2*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2
 + 50*d**4*n + 24*d**4) + D*c*d**3*n**3*x**3*(c + d*x)**n/(d**4*n**4 + 10*d
**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 3*D*c*d**3*n**2*x**3*(c +
```

```
d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 2
*D*c*d**3*n*x**3*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50
*d**4*n + 24*d**4) + D*d**4*n**3*x**4*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**
3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 6*D*d**4*n**2*x**4*(c + d*x)**n/(
d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n + 24*d**4) + 11*D*d**4*
n*x**4*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n**2 + 50*d**4*n +
24*d**4) + 6*D*d**4*x**4*(c + d*x)**n/(d**4*n**4 + 10*d**4*n**3 + 35*d**4*n
**2 + 50*d**4*n + 24*d**4), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.22 (sec) , antiderivative size = 234, normalized size of antiderivative = 1.86

$$\int (c+dx)^n \left(A+Bx+Cx^2+Dx^3\right)\,dx = \frac{(d^2(n+1)x^2+cdnx-c^2)(dx+c)^nB}{(n^2+3\,n+2)d^2} + \frac{(dx+c)^{n+1}A}{d(n+1)} + \frac{((n^2+3\,n+2)d^3x^3+(n^2+n)cd^2x^2-2\,c^2dnx+2\,c^3)(dx+c)^nC}{(n^3+6\,n^2+11\,n+6)d^3} + \frac{((n^3+6\,n^2+11\,n+6)d^4x^4+(n^3+3\,n^2+2\,n)cd^3x^3-3\,(n^2+n)c^2d^2x^2+6\,c^3dnx-6\,c^4)(dx+c)^nD}{(n^4+10\,n^3+35\,n^2+50\,n+24)d^4}$$

```
[In] integrate((d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="maxima")
```

```
[Out] (d^2*(n + 1)*x^2 + c*d*n*x - c^2)*(d*x + c)^n*B/((n^2 + 3*n + 2)*d^2) + (d*
x + c)^(n + 1)*A/(d*(n + 1)) + ((n^2 + 3*n + 2)*d^3*x^3 + (n^2 + n)*c*d^2*x
^2 - 2*c^2*d*n*x + 2*c^3)*(d*x + c)^n*C/((n^3 + 6*n^2 + 11*n + 6)*d^3) + ((
n^3 + 6*n^2 + 11*n + 6)*d^4*x^4 + (n^3 + 3*n^2 + 2*n)*c*d^3*x^3 - 3*(n^2 +
n)*c^2*d^2*x^2 + 6*c^3*d*n*x - 6*c^4)*(d*x + c)^n*D/((n^4 + 10*n^3 + 35*n^2
 + 50*n + 24)*d^4)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 728 vs. 2(127) = 254.

Time = 0.28 (sec) , antiderivative size = 728, normalized size of antiderivative = 5.78

$$\int (c+dx)^n \left(A+Bx+Cx^2+Dx^3\right)\,dx = \frac{(dx+c)^nDd^4n^3x^4+(dx+c)^nDcd^3n^3x^3+(dx+c)^nCd^4n^3x^3+6\,(dx+c)^nDd^4n^2x^4+(dx+c)^nCcd^3n^3x^2}{}$$

```
[In] integrate((d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="giac")
```

```
[Out] ((d*x + c)^n*D*d^4*n^3*x^4 + (d*x + c)^n*D*c*d^3*n^3*x^3 + (d*x + c)^n*C*d^
4*n^3*x^3 + 6*(d*x + c)^n*D*d^4*n^2*x^4 + (d*x + c)^n*C*c*d^3*n^3*x^2 + (d*
```

```
x + c)^n*B*d^4*n^3*x^2 + 3*(d*x + c)^n*D*c*d^3*n^2*x^3 + 7*(d*x + c)^n*C*d^
4*n^2*x^3 + 11*(d*x + c)^n*D*d^4*n*x^4 + (d*x + c)^n*B*c*d^3*n^3*x + (d*x +
 c)^n*A*d^4*n^3*x - 3*(d*x + c)^n*D*c^2*d^2*n^2*x^2 + 5*(d*x + c)^n*C*c*d^3
*n^2*x^2 + 8*(d*x + c)^n*B*d^4*n^2*x^2 + 2*(d*x + c)^n*D*c*d^3*n*x^3 + 14*(
d*x + c)^n*C*d^4*n*x^3 + 6*(d*x + c)^n*D*d^4*x^4 + (d*x + c)^n*A*c*d^3*n^3
- 2*(d*x + c)^n*C*c^2*d^2*n^2*x + 7*(d*x + c)^n*B*c*d^3*n^2*x + 9*(d*x + c)
^n*A*d^4*n^2*x - 3*(d*x + c)^n*D*c^2*d^2*n*x^2 + 4*(d*x + c)^n*C*c*d^3*n*x^
2 + 19*(d*x + c)^n*B*d^4*n*x^2 + 8*(d*x + c)^n*C*d^4*x^3 - (d*x + c)^n*B*c^
2*d^2*n^2 + 9*(d*x + c)^n*A*c*d^3*n^2 + 6*(d*x + c)^n*D*c^3*d*n*x - 8*(d*x
+ c)^n*C*c^2*d^2*n*x + 12*(d*x + c)^n*B*c*d^3*n*x + 26*(d*x + c)^n*A*d^4*n*
x + 12*(d*x + c)^n*B*d^4*x^2 + 2*(d*x + c)^n*C*c^3*d*n - 7*(d*x + c)^n*B*c^
2*d^2*n + 26*(d*x + c)^n*A*c*d^3*n + 24*(d*x + c)^n*A*d^4*x - 6*(d*x + c)^n
*D*c^4 + 8*(d*x + c)^n*C*c^3*d - 12*(d*x + c)^n*B*c^2*d^2 + 24*(d*x + c)^n*
A*c*d^3)/(d^4*n^4 + 10*d^4*n^3 + 35*d^4*n^2 + 50*d^4*n + 24*d^4)
```

Mupad [F(-1)]

Timed out.

$$\int (c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx=\int (c+d\,x)^n\,\left(A+B\,x+C\,x^2+x^3\,D\right)\,dx$$

```
[In] int((c + d*x)^n*(A + B*x + C*x^2 + x^3*D),x)
```

```
[Out] int((c + d*x)^n*(A + B*x + C*x^2 + x^3*D), x)
```

3.29 $\int \frac{(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)}{a+bx}\,dx$

Optimal result

Integrand size = 30, antiderivative size = 203

$$\begin{aligned}
&\int \frac{(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)}{a+bx}\,dx\\
&=\frac{(a^2d^2D-abd(Cd-cD)-b^2(cCd-Bd^2-c^2D))\,(c+dx)^{1+n}}{b^3d^3(1+n)}\\
&\quad+\frac{(bCd-2bcD-adD)(c+dx)^{2+n}}{b^2d^3(2+n)}+\frac{D(c+dx)^{3+n}}{bd^3(3+n)}\\
&\quad-\frac{(Ab^3-a(b^2B-abC+a^2D))\,(c+dx)^{1+n}\,\text{Hypergeometric2F1}\left(1,1+n,2+n,\frac{b(c+dx)}{bc-ad}\right)}{b^3(bc-ad)(1+n)}
\end{aligned}$$

```
[Out] (a^2*d^2*D-a*b*d*(C*d-D*c)-b^2*(-B*d^2+C*c*d-D*c^2))*(d*x+c)^(1+n)/b^3/d^3/
(1+n)+(C*b*d-D*a*d-2*D*b*c)*(d*x+c)^(2+n)/b^2/d^3/(2+n)+D*(d*x+c)^(3+n)/b/d
^3/(3+n)-(A*b^3-a*(B*b^2-C*a*b+D*a^2))*(d*x+c)^(1+n)*hypergeom([1, 1+n],[2+
n],b*(d*x+c)/(-a*d+b*c))/b^3/(-a*d+b*c)/(1+n)
```

Rubi [A] (verified)

Time = 0.13 (sec) , antiderivative size = 203, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.067, Rules used

$= \{1634, 70\}$

$$\int \frac{(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)}{a+bx}\,dx$$
$$= -\frac{(c+dx)^{n+1}\left(Ab^3-a(a^2D-abC+b^2B)\right)\text{Hypergeometric2F1}\left(1,n+1,n+2,\frac{b(c+dx)}{bc-ad}\right)}{b^3(n+1)(bc-ad)} + \frac{(c+dx)^{n+1}\left(a^2d^2D-abd(Cd-cD)-(b^2(-Bd^2+c^2(-D)+cCd))\right)}{b^3d^3(n+1)} + \frac{(c+dx)^{n+2}(-adD-2bcD+bCd)}{b^2d^3(n+2)} + \frac{D(c+dx)^{n+3}}{bd^3(n+3)}$$

```
[In] Int[((c + d*x)^n*(A + B*x + C*x^2 + D*x^3))/(a + b*x),x]
```

```
[Out] ((a^2*d^2*D - a*b*d*(C*d - c*D) - b^2*(c*C*d - B*d^2 - c^2*D))*(c + d*x)^(1
 + n))/(b^3*d^3*(1 + n)) + ((b*C*d - 2*b*c*D - a*d*D)*(c + d*x)^(2 + n))/(b
^2*d^3*(2 + n)) + (D*(c + d*x)^(3 + n))/(b*d^3*(3 + n)) - ((A*b^3 - a*(b^2*
B - a*b*C + a^2*D))*(c + d*x)^(1 + n)*Hypergeometric2F1[1, 1 + n, 2 + n, (b
*(c + d*x))/(b*c - a*d)])/(b^3*(b*c - a*d)*(1 + n))
```

Rule 70

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[(b
*c - a*d)^n*((a + b*x)^(m + 1)/(b^(n + 1)*(m + 1)))*Hypergeometric2F1[-n, m
 + 1, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 1634

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> Int[ExpandIntegrand[Px*(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c
, d, m, n}, x] && PolyQ[Px, x] && (IntegersQ[m, n] || IGtQ[m, -2]) && GtQ[E
xpon[Px, x], 2]
```

Rubi steps

$$\text{integral} = \int \left(\frac{\left(a^2d^2D-abd(Cd-cD)-b^2(cCd-Bd^2-c^2D)\right)(c+dx)^n}{b^3d^2} + \frac{\left(Ab^3-a(b^2B-abC+a^2D)\right)(c+dx)^n}{b^3(a+bx)} + \frac{(bCd-2bcD-adD)(c+dx)^{1+n}}{b^2d^2} + \frac{D(c+dx)^{2+n}}{bd^2}\right)dx$$

$$
\begin{aligned}
&= \frac{(a^2d^2D - abd(Cd - cD) - b^2(cCd - Bd^2 - c^2D))\,(c+dx)^{1+n}}{b^3d^3(1+n)} \\
&\quad + \frac{(bCd - 2bcD - adD)(c+dx)^{2+n}}{b^2d^3(2+n)} + \frac{D(c+dx)^{3+n}}{bd^3(3+n)} \\
&\quad + \left(A - \frac{a(b^2B - abC + a^2D)}{b^3}\right)\int \frac{(c+dx)^n}{a+bx}\,dx \\
&= \frac{(a^2d^2D - abd(Cd - cD) - b^2(cCd - Bd^2 - c^2D))\,(c+dx)^{1+n}}{b^3d^3(1+n)} \\
&\quad + \frac{(bCd - 2bcD - adD)(c+dx)^{2+n}}{b^2d^3(2+n)} + \frac{D(c+dx)^{3+n}}{bd^3(3+n)} \\
&\quad - \frac{\left(A - \frac{a(b^2B-abC+a^2D)}{b^3}\right)(c+dx)^{1+n}\,{}_2F_1\left(1,1+n;2+n;\frac{b(c+dx)}{bc-ad}\right)}{(bc-ad)(1+n)}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.36 (sec) , antiderivative size = 181, normalized size of antiderivative = 0.89

$$
\int \frac{(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)}{a+bx}\,dx
$$

$$
= \frac{(c+dx)^{1+n}\left(\frac{a^2d^2D+abd(-Cd+cD)+b^2(-cCd+Bd^2+c^2D)}{d^3(1+n)} + \frac{b(bCd-2bcD-adD)(c+dx)}{d^3(2+n)} + \frac{b^2D(c+dx)^2}{d^3(3+n)} - \frac{(Ab^3-a(b^2B-abC+a^2D}{}\right.}{b^3}
$$

```
[In] Integrate[((c + d*x)^n*(A + B*x + C*x^2 + D*x^3))/(a + b*x),x]
```

```
[Out] ((c + d*x)^(1 + n)*((a^2*d^2*D + a*b*d*(-(C*d) + c*D) + b^2*(-(c*C*d) + B*d
^2 + c^2*D))/(d^3*(1 + n)) + (b*(b*C*d - 2*b*c*D - a*d*D)*(c + d*x))/(d^3*(
2 + n)) + (b^2*D*(c + d*x)^2)/(d^3*(3 + n)) - ((A*b^3 - a*(b^2*B - a*b*C +
a^2*D))*Hypergeometric2F1[1, 1 + n, 2 + n, (b*(c + d*x))/(b*c - a*d)])/((b*
c - a*d)*(1 + n))))/b^3
```

Maple [F]

$$
\int \frac{(dx+c)^n\left(Dx^3+C\,x^2+Bx+A\right)}{bx+a}dx
$$

```
[In] int((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a),x)
```

```
[Out] int((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a),x)
```

Fricas [F]

$$\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{a+bx}\,dx=\int \frac{(capitalDx^3+Cx^2+Bx+A)(dx+c)^n}{bx+a}\,dx$$

```
[In] integrate((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a),x, algorithm="fricas")
[Out] integral((D*x^3 + C*x^2 + B*x + A)*(d*x + c)^n/(b*x + a), x)
```

Sympy [F]

$$\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{a+bx}\,dx=\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{a+bx}\,dx$$

```
[In] integrate((d*x+c)**n*(D*x**3+C*x**2+B*x+A)/(b*x+a),x)
[Out] Integral((c + d*x)**n*(A + B*x + C*x**2 + D*x**3)/(a + b*x), x)
```

Maxima [F]

$$\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{a+bx}\,dx=\int \frac{(Dx^3+Cx^2+Bx+A)(dx+c)^n}{bx+a}\,dx$$

```
[In] integrate((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a),x, algorithm="maxima")
[Out] integrate((D*x^3 + C*x^2 + B*x + A)*(d*x + c)^n/(b*x + a), x)
```

Giac [F]

$$\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{a+bx}\,dx=\int \frac{(Dx^3+Cx^2+Bx+A)(dx+c)^n}{bx+a}\,dx$$

```
[In] integrate((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a),x, algorithm="giac")
[Out] integrate((D*x^3 + C*x^2 + B*x + A)*(d*x + c)^n/(b*x + a), x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+dx)^n\,(A+Bx+Cx^2+Dx^3)}{a+bx}\,dx=\int \frac{(c+d\,x)^n\,(A+B\,x+C\,x^2+x^3\,D)}{a+b\,x}\,dx$$

```
[In] int(((c + d*x)^n*(A + B*x + C*x^2 + x^3*D))/(a + b*x),x)
[Out] int(((c + d*x)^n*(A + B*x + C*x^2 + x^3*D))/(a + b*x), x)
```

3.30 $\int \frac{(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)}{(a+bx)^2}\,dx$

Optimal result

Integrand size = 30, antiderivative size = 220

$$\int \frac{(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)}{(a+bx)^2}\,dx$$

$$=\frac{(bCd-bcD-2adD)(c+dx)^{1+n}}{b^3d^2(1+n)}-\frac{\left(A-\frac{a(b^2B-abC+a^2D)}{b^3}\right)(c+dx)^{1+n}}{(bc-ad)(a+bx)}+\frac{D(c+dx)^{2+n}}{b^2d^2(2+n)}$$

$$+\frac{(a^3dD(3+n)-b^3(Bc+Adn)+ab^2(2cC+Bd(1+n))-a^2b(3cD+Cd(2+n)))\,(c+dx)^{1+n}\,\text{Hyperg}}{b^3(bc-ad)^2(1+n)}$$

```
[Out] (C*b*d-2*D*a*d-D*b*c)*(d*x+c)^(1+n)/b^3/d^2/(1+n)-(A-a*(B*b^2-C*a*b+D*a^2)/
b^3)*(d*x+c)^(1+n)/(-a*d+b*c)/(b*x+a)+D*(d*x+c)^(2+n)/b^2/d^2/(2+n)+(a^3*d*
D*(3+n)-b^3*(A*d*n+B*c)+a*b^2*(2*C*c+B*d*(1+n))-a^2*b*(3*D*c+C*d*(2+n)))*(d
*x+c)^(1+n)*hypergeom([1, 1+n],[2+n],b*(d*x+c)/(-a*d+b*c))/b^3/(-a*d+b*c)^2
/(1+n)
```

Rubi [A] (verified)

Time = 0.36 (sec) , antiderivative size = 220, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {1635, 965, 81, 70}

$$\int \frac{(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)}{(a+bx)^2}\,dx=-\frac{(c+dx)^{n+1}\left(A-\frac{a(a^2D-abC+b^2B)}{b^3}\right)}{(a+bx)(bc-ad)}$$

$$+\frac{(c+dx)^{n+1}\,\text{Hypergeometric2F1}\left(1,n+1,n+2,\frac{b(c+dx)}{bc-ad}\right)(a^3dD(n+3)-a^2b(3cD+Cd(n+2))+ab^2}{b^3(n+1)(bc-ad)^2}$$

$$+\frac{(c+dx)^{n+1}(-2adD-bcD+bCd)}{b^3d^2(n+1)}+\frac{D(c+dx)^{n+2}}{b^2d^2(n+2)}$$

```
[In] Int[((c + d*x)^n*(A + B*x + C*x^2 + D*x^3))/(a + b*x)^2,x]
```

```
[Out] ((b*C*d - b*c*D - 2*a*d*D)*(c + d*x)^(1 + n))/(b^3*d^2*(1 + n)) - ((A - (a*
(b^2*B - a*b*C + a^2*D))/b^3)*(c + d*x)^(1 + n))/((b*c - a*d)*(a + b*x)) +
(D*(c + d*x)^(2 + n))/(b^2*d^2*(2 + n)) + ((a^3*d*D*(3 + n) - b^3*(B*c + A*
d*n) + a*b^2*(2*c*C + B*d*(1 + n)) - a^2*b*(3*c*D + C*d*(2 + n)))*(c + d*x)
^(1 + n)*Hypergeometric2F1[1, 1 + n, 2 + n, (b*(c + d*x))/(b*c - a*d)])/(b^
3*(b*c - a*d)^2*(1 + n))
```

Rule 70

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[(b
*c - a*d)^n*((a + b*x)^(m + 1)/(b^(n + 1)*(m + 1)))*Hypergeometric2F1[-n, m
 + 1, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 81

```
Int[((a_.) + (b_.)*(x_))*((c_.) + (d_.)*(x_))^(n_.)*((e_.) + (f_.)*(x_))^(p
_.), x_Symbol] :> Simp[b*(c + d*x)^(n + 1)*((e + f*x)^(p + 1)/(d*f*(n + p +
 2))), x] + Dist[(a*d*f*(n + p + 2) - b*(d*e*(n + 1) + c*f*(p + 1)))/(d*f*(
n + p + 2)), Int[(c + d*x)^n*(e + f*x)^p, x], x] /; FreeQ[{a, b, c, d, e, f
, n, p}, x] && NeQ[n + p + 2, 0]
```

Rule 965

```
Int[((d_.) + (e_.)*(x_))^(m_)*((f_.) + (g_.)*(x_))^(n_)*((a_.) + (b_.)*(x_)
 + (c_.)*(x_)^2)^(p_.), x_Symbol] :> Simp[c^p*(d + e*x)^(m + 2*p)*((f + g*x
)^(n + 1)/(g*e^(2*p)*(m + n + 2*p + 1))), x] + Dist[1/(g*e^(2*p)*(m + n + 2
*p + 1)), Int[(d + e*x)^m*(f + g*x)^n*ExpandToSum[g*(m + n + 2*p + 1)*(e^(2
*p)*(a + b*x + c*x^2)^p - c^p*(d + e*x)^(2*p)) - c^p*(e*f - d*g)*(m + 2*p)*
(d + e*x)^(2*p - 1), x], x], x] /; FreeQ[{a, b, c, d, e, f, g}, x] && NeQ[e
*f - d*g, 0] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IGt
Q[p, 0] && NeQ[m + n + 2*p + 1, 0] && (IntegerQ[n] ||  !IntegerQ[m])
```

Rule 1635

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol] :
> With[{Qx = PolynomialQuotient[Px, a + b*x, x], R = PolynomialRemainder[Px
, a + b*x, x]}, Simp[R*(a + b*x)^(m + 1)*((c + d*x)^(n + 1)/((m + 1)*(b*c -
 a*d))), x] + Dist[1/((m + 1)*(b*c - a*d)), Int[(a + b*x)^(m + 1)*(c + d*x)
^n*ExpandToSum[(m + 1)*(b*c - a*d)*Qx - d*R*(m + n + 2), x], x], x]] /; Fre
eQ[{a, b, c, d, n}, x] && PolyQ[Px, x] && ILtQ[m, -1] && GtQ[Expon[Px, x],
2]
```

Rubi steps

integral

$$= -\frac{\left(A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}\right)(c+dx)^{1+n}}{(bc-ad)(a+bx)} + \frac{\int \frac{(c+dx)^n\left(\frac{a^3dD(1+n)-b^3(Bc+Adn)+ab^2(cC+Bd(1+n))-a^2b(cD+Cd(1+n))}{b^3}-\frac{(bc-ad)(bC-aD)x}{b^2}-\left(c-\frac{ad}{b}\right)Dx^2\right)}{a+bx}\,dx}{-bc+ad}$$

$$= -\frac{\left(A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}\right)(c+dx)^{1+n}}{(bc-ad)(a+bx)} + \frac{D(c+dx)^{2+n}}{b^2d^2(2+n)} - \frac{\int \frac{(c+dx)^n\left(\frac{d(2+n)\left(a^3d^2D(1+n)-b^3d(Bc+Adn)-a^2bd(2cD+Cd(1+n))+ab^2\left(cCd+c^2D+Bd^2(1+n)\right)\right)}{b^2}-\frac{d(bc-ad)(bCd-bcD-2adD)(2+n)x}{b}\right)}{a+bx}\,dx}{bd^2(bc-ad)(2+n)}$$

$$= \frac{(bCd-bcD-2adD)(c+dx)^{1+n}}{b^3d^2(1+n)} - \frac{\left(A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}\right)(c+dx)^{1+n}}{(bc-ad)(a+bx)} + \frac{D(c+dx)^{2+n}}{b^2d^2(2+n)} - \frac{\left(a^3dD(3+n)-b^3(Bc+Adn)+ab^2(2cC+Bd(1+n))-a^2b(3cD+Cd(2+n))\right)\int \frac{(c+dx)^n}{a+bx}\,dx}{b^3(bc-ad)}$$

$$= \frac{(bCd-bcD-2adD)(c+dx)^{1+n}}{b^3d^2(1+n)} - \frac{\left(A-\frac{a\left(b^2B-abC+a^2D\right)}{b^3}\right)(c+dx)^{1+n}}{(bc-ad)(a+bx)} + \frac{D(c+dx)^{2+n}}{b^2d^2(2+n)} + \frac{\left(a^3dD(3+n)-b^3(Bc+Adn)+ab^2(2cC+Bd(1+n))-a^2b(3cD+Cd(2+n))\right)(c+dx)^{1+n}\,{}_2F_1(1,1}{b^3(bc-ad)^2(1+n)}$$

Mathematica [A] (verified)

Time = 0.69 (sec) , antiderivative size = 180, normalized size of antiderivative = 0.82

$$\int \frac{(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)}{(a+bx)^2}\,dx$$

$$= \frac{(c+dx)^{1+n}\left(\frac{bCd-bcD-2adD}{d^2(1+n)}+\frac{bD(c+dx)}{d^2(2+n)}-\frac{\left(b^2B-2abC+3a^2D\right)\text{Hypergeometric2F1}\left(1,1+n,2+n,\frac{b(c+dx)}{bc-ad}\right)}{(bc-ad)(1+n)}+\frac{d(Ab^3-a(b^2B-abC}{}\right.}{b^3}$$

```
[In] Integrate[((c + d*x)^n*(A + B*x + C*x^2 + D*x^3))/(a + b*x)^2,x]
```

```
[Out] ((c + d*x)^(1 + n)*((b*C*d - b*c*D - 2*a*d*D)/(d^2*(1 + n)) + (b*D*(c + d*x
))/(d^2*(2 + n)) - ((b^2*B - 2*a*b*C + 3*a^2*D)*Hypergeometric2F1[1, 1 + n,
 2 + n, (b*(c + d*x))/(b*c - a*d)])/((b*c - a*d)*(1 + n)) + (d*(A*b^3 - a*(
b^2*B - a*b*C + a^2*D))*Hypergeometric2F1[2, 1 + n, 2 + n, (b*(c + d*x))/(b
*c - a*d)])/((b*c - a*d)^2*(1 + n))))/b^3
```

Maple [F]

$$\int \frac{(dx+c)^{n} \left(D x^{3}+C \,x^{2}+B x +A \right)}{\left(b x +a \right)^{2}}d x$$

```
[In] int((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a)^2,x)
```

```
[Out] int((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a)^2,x)
```

Fricas [F]

$$\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{(a+bx)^2} \, dx = \int \frac{{\left(capitalDx^{3} + Cx^{2} + Bx + A\right)}{\left(dx + c\right)}^{n}}{{\left(bx + a\right)}^{2}} \,dx$$

```
[In] integrate((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a)^2,x, algorithm="fricas")
```

```
[Out] integral((D*x^3 + C*x^2 + B*x + A)*(d*x + c)^n/(b^2*x^2 + 2*a*b*x + a^2), x
)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{(a+bx)^2} \, dx = \text{Exception raised: HeuristicGCDFailed}$$

```
[In] integrate((d*x+c)**n*(D*x**3+C*x**2+B*x+A)/(b*x+a)**2,x)
```

```
[Out] Exception raised: HeuristicGCDFailed >> no luck
```

Maxima [F]

$$\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{(a+bx)^2} \, dx = \int \frac{{\left(Dx^{3} + Cx^{2} + Bx + A\right)}{\left(dx + c\right)}^{n}}{{\left(bx + a\right)}^{2}} \,dx$$

```
[In] integrate((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a)^2,x, algorithm="maxima")
```

```
[Out] integrate((D*x^3 + C*x^2 + B*x + A)*(d*x + c)^n/(b*x + a)^2, x)
```

Giac [F]

$$\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{(a+bx)^2} dx = \int \frac{(Dx^3+Cx^2+Bx+A)(dx+c)^n}{(bx+a)^2} dx$$

```
[In] integrate((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a)^2,x, algorithm="giac")
[Out] integrate((D*x^3 + C*x^2 + B*x + A)*(d*x + c)^n/(b*x + a)^2, x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{(a+bx)^2} dx = \int \frac{(c+d\,x)^n\,(A+B\,x+C\,x^2+x^3\,D)}{(a+b\,x)^2} dx$$

```
[In] int(((c + d*x)^n*(A + B*x + C*x^2 + x^3*D))/(a + b*x)^2,x)
[Out] int(((c + d*x)^n*(A + B*x + C*x^2 + x^3*D))/(a + b*x)^2, x)
```

3.31 $\int \frac{(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)}{(a+bx)^3}\,dx$

Optimal result

Integrand size = 30, antiderivative size = 329

$$\begin{aligned}
&\int \frac{(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)}{(a+bx)^3}\,dx \\
&= \frac{D(c+dx)^{1+n}}{b^3d(1+n)}-\frac{\left(Ab^3-a\left(b^2B-abC+a^2D\right)\right)(c+dx)^{1+n}}{2b^3(bc-ad)(a+bx)^2} \\
&\quad -\frac{\left(b^3(2Bc-Ad(1-n))-a^3dD(5+n)-ab^2(4cC+Bd(1+n))+a^2b(6cD+Cd(3+n))\right)(c+dx)^{1+n}}{2b^3(bc-ad)^2(a+bx)} \\
&\quad -\frac{\left(b^3\left(2c^2C+2Bcdn-Ad^2(1-n)n\right)-a^3d^2D\left(6+5n+n^2\right)+a^2bd(2+n)(6cD+Cd(1+n))-ab^2(6c^2D\right.}{2b^3(bc-ad)^3(1-}
\end{aligned}$$

```
[Out] D*(d*x+c)^(1+n)/b^3/d/(1+n)-1/2*(A*b^3-a*(B*b^2-C*a*b+D*a^2))*(d*x+c)^(1+n)
/b^3/(-a*d+b*c)/(b*x+a)^2-1/2*(b^3*(2*B*c-A*d*(1-n))-a^3*d*D*(5+n)-a*b^2*(4
*C*c+B*d*(1+n))+a^2*b*(6*D*c+C*d*(3+n)))*(d*x+c)^(1+n)/b^3/(-a*d+b*c)^2/(b*
x+a)-1/2*(b^3*(2*C*c^2+2*B*c*d*n-A*d^2*(1-n)*n)-a^3*d^2*D*(n^2+5*n+6)+a^2*b
*d*(2+n)*(6*D*c+C*d*(1+n))-a*b^2*(6*D*c^2+4*c*C*d*(1+n)+B*d^2*n*(1+n)))*(d*
x+c)^(1+n)*hypergeom([1, 1+n],[2+n],b*(d*x+c)/(-a*d+b*c))/b^3/(-a*d+b*c)^3/
(1+n)
```

Rubi [A] (verified)

Time = 0.40 (sec) , antiderivative size = 329, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}} = 0.133$, Rules used = {1635, 963, 81, 70}

$$\int \frac{(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)}{(a+bx)^3}\,dx = -\frac{(c+dx)^{n+1}\left(Ab^3-a(a^2D-abC+b^2B)\right)}{2b^3(a+bx)^2(bc-ad)}$$
$$-\frac{(c+dx)^{n+1}\,\text{Hypergeometric2F1}\left(1,n+1,n+2,\frac{b(c+dx)}{bc-ad}\right)\left(a^3\left(-d^2\right)D(n^2+5n+6)+a^2bd(n+2)(6cD\right.}{2b^3(n+1)(b}$$
$$-\frac{(c+dx)^{n+1}\left(a^3(-d)D(n+5)+a^2b(6cD+Cd(n+3))-ab^2(Bd(n+1)+4cC)+b^3(2Bc-Ad(1-n\right.}{2b^3(a+bx)(bc-ad)^2}$$
$$+\frac{D(c+dx)^{n+1}}{b^3d(n+1)}$$

```
[In] Int[((c + d*x)^n*(A + B*x + C*x^2 + D*x^3))/(a + b*x)^3,x]
```

```
[Out] (D*(c + d*x)^(1 + n))/(b^3*d*(1 + n)) - ((A*b^3 - a*(b^2*B - a*b*C + a^2*D)
)*(c + d*x)^(1 + n))/(2*b^3*(b*c - a*d)*(a + b*x)^2) - ((b^3*(2*B*c - A*d*(
1 - n)) - a^3*d*D*(5 + n) - a*b^2*(4*c*C + B*d*(1 + n)) + a^2*b*(6*c*D + C*
d*(3 + n)))*(c + d*x)^(1 + n))/(2*b^3*(b*c - a*d)^2*(a + b*x)) - ((b^3*(2*c
^2*C + 2*B*c*d*n - A*d^2*(1 - n)*n) - a^3*d^2*D*(6 + 5*n + n^2) + a^2*b*d*(
2 + n)*(6*c*D + C*d*(1 + n)) - a*b^2*(6*c^2*D + 4*c*C*d*(1 + n) + B*d^2*n*(
1 + n)))*(c + d*x)^(1 + n)*Hypergeometric2F1[1, 1 + n, 2 + n, (b*(c + d*x))
/(b*c - a*d)])/(2*b^3*(b*c - a*d)^3*(1 + n))
```

Rule 70

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[(b
*c - a*d)^n*((a + b*x)^(m + 1)/(b^(n + 1)*(m + 1)))*Hypergeometric2F1[-n, m
 + 1, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 81

```
Int[((a_.) + (b_.)*(x_))*((c_.) + (d_.)*(x_))^(n_.)*((e_.) + (f_.)*(x_))^(p
_.), x_Symbol] :> Simp[b*(c + d*x)^(n + 1)*((e + f*x)^(p + 1)/(d*f*(n + p +
 2))), x] + Dist[(a*d*f*(n + p + 2) - b*(d*e*(n + 1) + c*f*(p + 1)))/(d*f*(
n + p + 2)), Int[(c + d*x)^n*(e + f*x)^p, x], x] /; FreeQ[{a, b, c, d, e, f
, n, p}, x] && NeQ[n + p + 2, 0]
```

Rule 963

```
Int[((d_.) + (e_.)*(x_))^(m_)*((f_.) + (g_.)*(x_))^(n_)*((a_.) + (b_.)*(x_)
 + (c_.)*(x_)^2)^(p_.), x_Symbol] :> With[{Qx = PolynomialQuotient[(a + b*x
 + c*x^2)^p, d + e*x, x], R = PolynomialRemainder[(a + b*x + c*x^2)^p, d +
```

```
e*x, x]}, Simp[R*(d + e*x)^(m + 1)*((f + g*x)^(n + 1)/((m + 1)*(e*f - d*g))
), x] + Dist[1/((m + 1)*(e*f - d*g)), Int[(d + e*x)^(m + 1)*(f + g*x)^n*Exp
andToSum[(m + 1)*(e*f - d*g)*Qx - g*R*(m + n + 2), x], x], x]] /; FreeQ[{a,
 b, c, d, e, f, g}, x] && NeQ[e*f - d*g, 0] && NeQ[b^2 - 4*a*c, 0] && NeQ[c
*d^2 - b*d*e + a*e^2, 0] && IGtQ[p, 0] && LtQ[m, -1]
```

Rule 1635

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol] :
> With[{Qx = PolynomialQuotient[Px, a + b*x, x], R = PolynomialRemainder[Px
, a + b*x, x]}, Simp[R*(a + b*x)^(m + 1)*((c + d*x)^(n + 1)/((m + 1)*(b*c -
 a*d))), x] + Dist[1/((m + 1)*(b*c - a*d)), Int[(a + b*x)^(m + 1)*(c + d*x)
^n*ExpandToSum[(m + 1)*(b*c - a*d)*Qx - d*R*(m + n + 2), x], x], x]] /; Fre
eQ[{a, b, c, d, n}, x] && PolyQ[Px, x] && ILtQ[m, -1] && GtQ[Expon[Px, x],
2]
```

Rubi steps

$$\text{integral} = -\frac{(Ab^3-a(b^2B-abC+a^2D))\,(c+dx)^{1+n}}{2b^3(bc-ad)(a+bx)^2} - \frac{\int \frac{(c+dx)^n\left(-2Bc+Ad(1-n)+\frac{a^3dD(1+n)}{b^3}+\frac{a(2cC+Bd(1+n))}{b}-\frac{a^2(2cD+Cd(1+n))}{b^2}-\frac{2(bc-ad)(bC-aD)x}{b^2}-2\left(c-\frac{ad}{b}\right)Dx^2\right)}{(a+bx)^2}\,dx}{2(bc-ad)}$$

$$= -\frac{(Ab^3-a(b^2B-abC+a^2D))\,(c+dx)^{1+n}}{2b^3(bc-ad)(a+bx)^2} - \frac{(b^3(2Bc-Ad(1-n))-a^3dD(5+n)-ab^2(4cC+Bd(1+n))+a^2b(6cD+Cd(3+n)))\,(c+d}{2b^3(bc-ad)^2(a+bx)} + \frac{\int \frac{(c+dx)^n\Bigg(2c^2C+2Bcdn-Ad^2(1-n)n-\frac{a^3d^2D(4+5n+n^2)}{b^3}-\frac{a(4c^2D+4cCd(1+n)+Bd^2n(1+n))}{b}+\frac{a^2d(2cD(4+3n)+Cd(2+3n+n^2))}{b^2}+2(}{a+bx}}{2(bc-ad)^2}$$

$$= \frac{D(c+dx)^{1+n}}{b^3d(1+n)} - \frac{(Ab^3-a(b^2B-abC+a^2D))\,(c+dx)^{1+n}}{2b^3(bc-ad)(a+bx)^2} - \frac{(b^3(2Bc-Ad(1-n))-a^3dD(5+n)-ab^2(4cC+Bd(1+n))+a^2b(6cD+Cd(3+n)))\,(c+d}{2b^3(bc-ad)^2(a+bx)} + \frac{(b^3(2c^2C+2Bcdn-Ad^2(1-n)n)-a^3d^2D(6+5n+n^2)+a^2bd(2+n)(6cD+Cd(1+n))-a}{2b^3(bc-ad)^2}$$

$$= \frac{D(c+dx)^{1+n}}{b^3d(1+n)} - \frac{(Ab^3 - a(b^2B - abC + a^2D))\,(c+dx)^{1+n}}{2b^3(bc-ad)(a+bx)^2}$$
$$- \frac{(b^3(2Bc - Ad(1-n)) - a^3dD(5+n) - ab^2(4cC + Bd(1+n)) + a^2b(6cD + Cd(3+n)))\,(c+}{2b^3(bc-ad)^2(a+bx)}$$
$$- \frac{(b^3(2c^2C + 2Bcdn - Ad^2(1-n)n) - a^3d^2D(6+5n+n^2) + a^2bd(2+n)(6cD + Cd(1+n)) -}{2b^3(bc-ad)^3(1}$$

Mathematica [A] (verified)

Time = 1.31 (sec) , antiderivative size = 188, normalized size of antiderivative = 0.57

$$\int \frac{(c+dx)^n\,(A+Bx+Cx^2+Dx^3)}{(a+bx)^3}\,dx$$
$$= \frac{(c+dx)^{1+n}\left(\frac{D}{d} - \frac{(bC-3aD)\,\text{Hypergeometric2F1}\left(1,1+n,2+n,\frac{b(c+dx)}{bc-ad}\right)}{bc-ad} + \frac{d(b^2B-2abC+3a^2D)\,\text{Hypergeometric2F1}\left(2,1+n,2+n,\frac{b(c+}{bc-}\right.}{(bc-ad)^2}\right.}{b^3(1+n)}$$

```
[In] Integrate[((c + d*x)^n*(A + B*x + C*x^2 + D*x^3))/(a + b*x)^3,x]
```

```
[Out] ((c + d*x)^(1 + n)*(D/d - ((b*C - 3*a*D)*Hypergeometric2F1[1, 1 + n, 2 + n,
 (b*(c + d*x))/(b*c - a*d)])/(b*c - a*d) + (d*(b^2*B - 2*a*b*C + 3*a^2*D)*H
ypergeometric2F1[2, 1 + n, 2 + n, (b*(c + d*x))/(b*c - a*d)])/(b*c - a*d)^2
 - (d^2*(A*b^3 - a*(b^2*B - a*b*C + a^2*D))*Hypergeometric2F1[3, 1 + n, 2 +
 n, (b*(c + d*x))/(b*c - a*d)])/(b*c - a*d)^3))/(b^3*(1 + n))
```

Maple [F]

$$\int \frac{(dx+c)^n\,(Dx^3+C\,x^2+Bx+A)}{(bx+a)^3}dx$$

```
[In] int((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a)^3,x)
```

```
[Out] int((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a)^3,x)
```

Fricas [F]

$$\int \frac{(c+dx)^n\,(A+Bx+Cx^2+Dx^3)}{(a+bx)^3}\,dx = \int \frac{(capitalDx^3 + Cx^2 + Bx + A)(dx+c)^n}{(bx+a)^3}\,dx$$

```
[In] integrate((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a)^3,x, algorithm="fricas")
```

```
[Out] integral((D*x^3 + C*x^2 + B*x + A)*(d*x + c)^n/(b^3*x^3 + 3*a*b^2*x^2 + 3*a
^2*b*x + a^3), x)
```

Sympy [F]

$$\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{(a+bx)^3} \, dx = \int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{(a+bx)^3} \, dx$$

```
[In] integrate((d*x+c)**n*(D*x**3+C*x**2+B*x+A)/(b*x+a)**3,x)
[Out] Integral((c + d*x)**n*(A + B*x + C*x**2 + D*x**3)/(a + b*x)**3, x)
```

Maxima [F]

$$\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{(a+bx)^3} \, dx = \int \frac{(Dx^3+Cx^2+Bx+A)(dx+c)^n}{(bx+a)^3} \, dx$$

```
[In] integrate((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a)^3,x, algorithm="maxima")
[Out] integrate((D*x^3 + C*x^2 + B*x + A)*(d*x + c)^n/(b*x + a)^3, x)
```

Giac [F]

$$\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{(a+bx)^3} \, dx = \int \frac{(Dx^3+Cx^2+Bx+A)(dx+c)^n}{(bx+a)^3} \, dx$$

```
[In] integrate((d*x+c)^n*(D*x^3+C*x^2+B*x+A)/(b*x+a)^3,x, algorithm="giac")
[Out] integrate((D*x^3 + C*x^2 + B*x + A)*(d*x + c)^n/(b*x + a)^3, x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+dx)^n (A+Bx+Cx^2+Dx^3)}{(a+bx)^3} \, dx = \int \frac{(c+d\,x)^n \left(A+B\,x+C\,x^2+x^3\,D\right)}{(a+b\,x)^3} \, dx$$

```
[In] int(((c + d*x)^n*(A + B*x + C*x^2 + x^3*D))/(a + b*x)^3,x)
[Out] int(((c + d*x)^n*(A + B*x + C*x^2 + x^3*D))/(a + b*x)^3, x)
```

3.32 $\int (a+bx)^m(A+Bx)(c+dx)^n\,dx$

Optimal result

Integrand size = 20, antiderivative size = 141

$$\int (a+bx)^m(A+Bx)(c+dx)^n\,dx = \frac{B(a+bx)^{1+m}(c+dx)^{1+n}}{bd(2+m+n)} + \frac{(Abd(2+m+n)-B(bc(1+m)+ad(1+n)))(a+bx)^{1+m}(c+dx)^n\left(\frac{b(c+dx)}{bc-ad}\right)^{-n}\text{Hypergeometric2F1}}{b^2d(1+m)(2+m+n)}$$

```
[Out] B*(b*x+a)^(1+m)*(d*x+c)^(1+n)/b/d/(2+m+n)+(A*b*d*(2+m+n)-B*(b*c*(1+m)+a*d*(
1+n)))*(b*x+a)^(1+m)*(d*x+c)^n*hypergeom([-n, 1+m],[2+m],-d*(b*x+a)/(-a*d+b
*c))/b^2/d/(1+m)/(2+m+n)/((b*(d*x+c)/(-a*d+b*c))^n)
```

Rubi [A] (verified)

Time = 0.06 (sec) , antiderivative size = 141, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {81, 72, 71}

$$\int (a+bx)^m(A+Bx)(c+dx)^n\,dx$$

$$= \frac{(a+bx)^{m+1}(c+dx)^n\left(\frac{b(c+dx)}{bc-ad}\right)^{-n}(Abd(m+n+2)-B(ad(n+1)+bc(m+1)))\,\text{Hypergeometric2F1}\,(}{b^2d(m+1)(m+n+2)} + \frac{B(a+bx)^{m+1}(c+dx)^{n+1}}{bd(m+n+2)}$$

```
[In] Int[(a + b*x)^m*(A + B*x)*(c + d*x)^n,x]
```

```
[Out] (B*(a + b*x)^(1 + m)*(c + d*x)^(1 + n))/(b*d*(2 + m + n)) + ((A*b*d*(2 + m
+ n) - B*(b*c*(1 + m) + a*d*(1 + n)))*(a + b*x)^(1 + m)*(c + d*x)^n*Hyperge
```

```
ometric2F1[1 + m, -n, 2 + m, -((d*(a + b*x))/(b*c - a*d))])/(b^2*d*(1 + m)*
(2 + m + n)*((b*(c + d*x))/(b*c - a*d))^n)
```

Rule 71

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[((
a + b*x)^(m + 1)/(b*(m + 1)*(b/(b*c - a*d))^n))*Hypergeometric2F1[-n, m + 1
, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m, n}, x]
&& NeQ[b*c - a*d, 0] &&  !IntegerQ[m] &&  !IntegerQ[n] && GtQ[b/(b*c - a*d)
, 0] && (RationalQ[m] ||  !(RationalQ[n] && GtQ[-d/(b*c - a*d), 0]))
```

Rule 72

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Dist[(c
 + d*x)^FracPart[n]/((b/(b*c - a*d))^IntPart[n]*(b*((c + d*x)/(b*c - a*d)))
^FracPart[n]), Int[(a + b*x)^m*Simp[b*(c/(b*c - a*d)) + b*d*(x/(b*c - a*d))
, x]^n, x], x] /; FreeQ[{a, b, c, d, m, n}, x] && NeQ[b*c - a*d, 0] &&  !In
tegerQ[m] &&  !IntegerQ[n] && (RationalQ[m] ||  !SimplerQ[n + 1, m + 1])
```

Rule 81

```
Int[((a_.) + (b_.)*(x_))*((c_.) + (d_.)*(x_))^(n_.)*((e_.) + (f_.)*(x_))^(p
_.), x_Symbol] :> Simp[b*(c + d*x)^(n + 1)*((e + f*x)^(p + 1)/(d*f*(n + p +
 2))), x] + Dist[(a*d*f*(n + p + 2) - b*(d*e*(n + 1) + c*f*(p + 1)))/(d*f*(
n + p + 2)), Int[(c + d*x)^n*(e + f*x)^p, x], x] /; FreeQ[{a, b, c, d, e, f
, n, p}, x] && NeQ[n + p + 2, 0]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \frac{B(a+bx)^{1+m}(c+dx)^{1+n}}{bd(2+m+n)} \\
&\quad + \left(A - \frac{B(bc(1+m)+ad(1+n))}{bd(2+m+n)}\right) \int (a+bx)^m (c+dx)^n \, dx \\
&= \frac{B(a+bx)^{1+m}(c+dx)^{1+n}}{bd(2+m+n)} \\
&\quad + \left(\left(A - \frac{B(bc(1+m)+ad(1+n))}{bd(2+m+n)}\right)(c+dx)^n \left(\frac{b(c+dx)}{bc-ad}\right)^{-n}\right) \int (a \\
&\quad + bx)^m \left(\frac{bc}{bc-ad} + \frac{bdx}{bc-ad}\right)^n \, dx \\
&= \frac{B(a+bx)^{1+m}(c+dx)^{1+n}}{bd(2+m+n)} \\
&\quad + \frac{\left(A - \frac{B(bc(1+m)+ad(1+n))}{bd(2+m+n)}\right)(a+bx)^{1+m}(c+dx)^n \left(\frac{b(c+dx)}{bc-ad}\right)^{-n} {}_2F_1\left(1+m,-n;2+m;-\frac{d(a+bx)}{bc-ad}\right)}{b(1+m)}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.10 (sec) , antiderivative size = 117, normalized size of antiderivative = 0.83

$$\int (a+bx)^m(A+Bx)(c+dx)^n\,dx$$
$$=\frac{(a+bx)^{1+m}(c+dx)^n\left(bB(c+dx)-\frac{(bBc(1+m)+aBd(1+n)-Abd(2+m+n))\left(\frac{b(c+dx)}{bc-ad}\right)^{-n}\text{Hypergeometric2F1}\left(1+m,-n,2+m,\right.}{1+m}\right.}{b^2d(2+m+n)}$$

```
[In] Integrate[(a + b*x)^m*(A + B*x)*(c + d*x)^n,x]
```

```
[Out] ((a + b*x)^(1 + m)*(c + d*x)^n*(b*B*(c + d*x) - ((b*B*c*(1 + m) + a*B*d*(1
+ n) - A*b*d*(2 + m + n))*Hypergeometric2F1[1 + m, -n, 2 + m, (d*(a + b*x))
/(-(b*c) + a*d)])/((1 + m)*((b*(c + d*x))/(b*c - a*d))^n)))/(b^2*d*(2 + m +
 n))
```

Maple [F]

$$\int \left(bx+a\right)^{m} \left(Bx+A\right) \left(dx+c\right)^{n}dx$$

```
[In] int((b*x+a)^m*(B*x+A)*(d*x+c)^n,x)
```

```
[Out] int((b*x+a)^m*(B*x+A)*(d*x+c)^n,x)
```

Fricas [F]

$$\int (a+bx)^m(A+Bx)(c+dx)^n\,dx=\int (Bx+A)(bx+a)^m(dx+c)^n\,dx$$

```
[In] integrate((b*x+a)^m*(B*x+A)*(d*x+c)^n,x, algorithm="fricas")
```

```
[Out] integral((B*x + A)*(b*x + a)^m*(d*x + c)^n, x)
```

Sympy [F(-2)]

Exception generated.

$$\int (a+bx)^m(A+Bx)(c+dx)^n\,dx=\text{Exception raised: HeuristicGCDFailed}$$

```
[In] integrate((b*x+a)**m*(B*x+A)*(d*x+c)**n,x)
```

```
[Out] Exception raised: HeuristicGCDFailed >> no luck
```

Maxima [F]

$$\int (a+bx)^m(A+Bx)(c+dx)^n\,dx = \int (Bx+A)(bx+a)^m(dx+c)^n\,dx$$

```
[In] integrate((b*x+a)^m*(B*x+A)*(d*x+c)^n,x, algorithm="maxima")
[Out] integrate((B*x + A)*(b*x + a)^m*(d*x + c)^n, x)
```

Giac [F]

$$\int (a+bx)^m(A+Bx)(c+dx)^n\,dx = \int (Bx+A)(bx+a)^m(dx+c)^n\,dx$$

```
[In] integrate((b*x+a)^m*(B*x+A)*(d*x+c)^n,x, algorithm="giac")
[Out] integrate((B*x + A)*(b*x + a)^m*(d*x + c)^n, x)
```

Mupad [F(-1)]

Timed out.

$$\int (a+bx)^m(A+Bx)(c+dx)^n\,dx = \int (A+B\,x)\,\left(a+b\,x\right)^m\,\left(c+d\,x\right)^n\,dx$$

```
[In] int((A + B*x)*(a + b*x)^m*(c + d*x)^n,x)
[Out] int((A + B*x)*(a + b*x)^m*(c + d*x)^n, x)
```

3.33 $\int (a+bx)^m(c+dx)^n \left(A+Bx+Cx^2\right) \, dx$

Optimal result

Integrand size = 25, antiderivative size = 268

$$\begin{aligned}
&\int (a+bx)^m(c+dx)^n \left(A+Bx+Cx^2\right) \, dx \\
&= -\frac{(aCd(4+m+2n)+b(cC(2+m)-Bd(3+m+n)))(a+bx)^{1+m}(c+dx)^{1+n}}{b^2d^2(2+m+n)(3+m+n)} \\
&\quad +\frac{C(a+bx)^{2+m}(c+dx)^{1+n}}{b^2d(3+m+n)} \\
&\quad -\frac{(d(2+m+n)\left(abcC(2+m)+a^2Cd(1+n)-Ab^2d(3+m+n)\right)-(bc(1+m)+ad(1+n))(aCd(4+}{b^3d^2}
\end{aligned}$$

```
[Out] -(a*C*d*(4+m+2*n)+b*(c*C*(2+m)-B*d*(3+m+n)))*(b*x+a)^(1+m)*(d*x+c)^(1+n)/b^
2/d^2/(2+m+n)/(3+m+n)+C*(b*x+a)^(2+m)*(d*x+c)^(1+n)/b^2/d/(3+m+n)-(d*(2+m+n
)*(a*b*c*C*(2+m)+a^2*C*d*(1+n)-A*b^2*d*(3+m+n))-(b*c*(1+m)+a*d*(1+n))*(a*C*
d*(4+m+2*n)+b*(c*C*(2+m)-B*d*(3+m+n))))*(b*x+a)^(1+m)*(d*x+c)^n*hypergeom([
-n, 1+m],[2+m],-d*(b*x+a)/(-a*d+b*c))/b^3/d^2/(1+m)/(2+m+n)/(3+m+n)/((b*(d*
x+c)/(-a*d+b*c))^n)
```

Rubi [A] (verified)

Time = 0.20 (sec) , antiderivative size = 266, normalized size of antiderivative = 0.99, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.160, Rules used

$= \{965, 81, 72, 71\}$

$$\int (a+bx)^m (c+dx)^n \left(A+Bx+Cx^2\right)\, dx =$$

$$-\frac{(a+bx)^{m+1}(c+dx)^n \left(\frac{b(c+dx)}{bc-ad}\right)^{-n} \text{Hypergeometric2F1}\left(m+1,-n,m+2,-\frac{d(a+bx)}{bc-ad}\right) (d(m+n+2)\,(a^2}{b^3d^2(}$$

$$-\frac{(a+bx)^{m+1}(c+dx)^{n+1}(aCd(m+2n+4)-bBd(m+n+3)+bcC(m+2))}{b^2d^2(m+n+2)(m+n+3)}$$

$$+\frac{C(a+bx)^{m+2}(c+dx)^{n+1}}{b^2d(m+n+3)}$$

```
[In] Int[(a + b*x)^m*(c + d*x)^n*(A + B*x + C*x^2),x]
```

```
[Out] -(((b*c*C*(2 + m) - b*B*d*(3 + m + n) + a*C*d*(4 + m + 2*n))*(a + b*x)^(1 +
 m)*(c + d*x)^(1 + n))/(b^2*d^2*(2 + m + n)*(3 + m + n))) + (C*(a + b*x)^(2
 + m)*(c + d*x)^(1 + n))/(b^2*d*(3 + m + n)) - ((d*(2 + m + n)*(a*b*c*C*(2
+ m) + a^2*C*d*(1 + n) - A*b^2*d*(3 + m + n)) - (b*c*(1 + m) + a*d*(1 + n))
*(b*c*C*(2 + m) - b*B*d*(3 + m + n) + a*C*d*(4 + m + 2*n)))*(a + b*x)^(1 +
m)*(c + d*x)^n*Hypergeometric2F1[1 + m, -n, 2 + m, -((d*(a + b*x))/(b*c - a
*d))])/(b^3*d^2*(1 + m)*(2 + m + n)*(3 + m + n)*((b*(c + d*x))/(b*c - a*d))
^n)
```

Rule 71

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[((
a + b*x)^(m + 1)/(b*(m + 1)*(b/(b*c - a*d))^n))*Hypergeometric2F1[-n, m + 1
, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m, n}, x]
&& NeQ[b*c - a*d, 0] &&  !IntegerQ[m] &&  !IntegerQ[n] && GtQ[b/(b*c - a*d)
, 0] && (RationalQ[m] ||  !(RationalQ[n] && GtQ[-d/(b*c - a*d), 0]))
```

Rule 72

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Dist[(c
 + d*x)^FracPart[n]/((b/(b*c - a*d))^IntPart[n]*(b*((c + d*x)/(b*c - a*d)))
^FracPart[n]), Int[(a + b*x)^m*Simp[b*(c/(b*c - a*d)) + b*d*(x/(b*c - a*d))
, x]^n, x], x] /; FreeQ[{a, b, c, d, m, n}, x] && NeQ[b*c - a*d, 0] &&  !In
tegerQ[m] &&  !IntegerQ[n] && (RationalQ[m] ||  !SimplerQ[n + 1, m + 1])
```

Rule 81

```
Int[((a_.) + (b_.)*(x_))*((c_.) + (d_.)*(x_))^(n_.)*((e_.) + (f_.)*(x_))^(p
_.), x_Symbol] :> Simp[b*(c + d*x)^(n + 1)*((e + f*x)^(p + 1)/(d*f*(n + p +
 2))), x] + Dist[(a*d*f*(n + p + 2) - b*(d*e*(n + 1) + c*f*(p + 1)))/(d*f*(
n + p + 2)), Int[(c + d*x)^n*(e + f*x)^p, x], x] /; FreeQ[{a, b, c, d, e, f
, n, p}, x] && NeQ[n + p + 2, 0]
```

Rule 965

```
Int[((d_.) + (e_.)*(x_))^(m_)*((f_.) + (g_.)*(x_))^(n_)*((a_.) + (b_.)*(x_)
 + (c_.)*(x_)^2)^(p_.), x_Symbol] :> Simp[c^p*(d + e*x)^(m + 2*p)*((f + g*x
)^(n + 1)/(g*e^(2*p)*(m + n + 2*p + 1))), x] + Dist[1/(g*e^(2*p)*(m + n + 2
*p + 1)), Int[(d + e*x)^m*(f + g*x)^n*ExpandToSum[g*(m + n + 2*p + 1)*(e^(2
*p)*(a + b*x + c*x^2)^p - c^p*(d + e*x)^(2*p)) - c^p*(e*f - d*g)*(m + 2*p)*
(d + e*x)^(2*p - 1), x], x], x] /; FreeQ[{a, b, c, d, e, f, g}, x] && NeQ[e
*f - d*g, 0] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IGt
Q[p, 0] && NeQ[m + n + 2*p + 1, 0] && (IntegerQ[n] ||  !IntegerQ[m])
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \frac{C(a+bx)^{2+m}(c+dx)^{1+n}}{b^2d(3+m+n)} \\
&\quad + \frac{\int (a+bx)^m(c+dx)^n\,(-abcC(2+m)-a^2Cd(1+n)+Ab^2d(3+m+n)-b(bcC(2+m)-bBd(3+}{b^2d(3+m+n)} \\
&= -\frac{(bcC(2+m)-bBd(3+m+n)+aCd(4+m+2n))(a+bx)^{1+m}(c+dx)^{1+n}}{b^2d^2(2+m+n)(3+m+n)} \\
&\quad + \frac{C(a+bx)^{2+m}(c+dx)^{1+n}}{b^2d(3+m+n)} \\
&\quad - \frac{(abcC(2+m)+a^2Cd(1+n)-Ab^2d(3+m+n)-\frac{(bc(1+m)+ad(1+n))(bcC(2+m)-bBd(3+m+n)+aCd(4}{d(2+m+n)}}{b^2d(3+m+n)} \\
&= -\frac{(bcC(2+m)-bBd(3+m+n)+aCd(4+m+2n))(a+bx)^{1+m}(c+dx)^{1+n}}{b^2d^2(2+m+n)(3+m+n)} \\
&\quad + \frac{C(a+bx)^{2+m}(c+dx)^{1+n}}{b^2d(3+m+n)} \\
&\quad - \frac{((abcC(2+m)+a^2Cd(1+n)-Ab^2d(3+m+n)-\frac{(bc(1+m)+ad(1+n))(bcC(2+m)-bBd(3+m+n)+aC}{d(2+m+n)}}{b^2d(3+m+n)} \\
&= -\frac{(bcC(2+m)-bBd(3+m+n)+aCd(4+m+2n))(a+bx)^{1+m}(c+dx)^{1+n}}{b^2d^2(2+m+n)(3+m+n)} \\
&\quad + \frac{C(a+bx)^{2+m}(c+dx)^{1+n}}{b^2d(3+m+n)} \\
&\quad - \frac{(abcC(2+m)+a^2Cd(1+n)-Ab^2d(3+m+n)-\frac{(bc(1+m)+ad(1+n))(bcC(2+m)-bBd(3+m+n)+aCd(4}{d(2+m+n)}}{b^3d(1+m)(3+m+}
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.19 (sec) , antiderivative size = 187, normalized size of antiderivative = 0.70

$$\int (a+bx)^m(c+dx)^n \left(A+Bx+Cx^2\right)\,dx$$

$$= \frac{(a+bx)^{1+m}(c+dx)^n \left(\frac{b(c+dx)}{bc-ad}\right)^{-n} \left(C(bc-ad)^2\,\text{Hypergeometric2F1}\left(1+m,-2-n,2+m,\frac{d(a+bx)}{-bc+ad}\right)+b\Big(\right.}{}$$

```
[In] Integrate[(a + b*x)^m*(c + d*x)^n*(A + B*x + C*x^2),x]
```

```
[Out] ((a + b*x)^(1 + m)*(c + d*x)^n*(C*(b*c - a*d)^2*Hypergeometric2F1[1 + m, -2
 - n, 2 + m, (d*(a + b*x))/(-(b*c) + a*d)] + b*(-((b*c - a*d)*(2*c*C - B*d)
*Hypergeometric2F1[1 + m, -1 - n, 2 + m, (d*(a + b*x))/(-(b*c) + a*d)]) + b
*(c^2*C - B*c*d + A*d^2)*Hypergeometric2F1[1 + m, -n, 2 + m, (d*(a + b*x))/
(-(b*c) + a*d)])))/(b^3*d^2*(1 + m)*((b*(c + d*x))/(b*c - a*d))^n)
```

Maple [F]

$$\int (bx+a)^m (dx+c)^n \left(C\,x^2+Bx+A\right)dx$$

```
[In] int((b*x+a)^m*(d*x+c)^n*(C*x^2+B*x+A),x)
```

```
[Out] int((b*x+a)^m*(d*x+c)^n*(C*x^2+B*x+A),x)
```

Fricas [F]

$$\int (a+bx)^m(c+dx)^n \left(A+Bx+Cx^2\right)\,dx = \int (Cx^2+Bx+A)(bx+a)^m(dx+c)^n\,dx$$

```
[In] integrate((b*x+a)^m*(d*x+c)^n*(C*x^2+B*x+A),x, algorithm="fricas")
```

```
[Out] integral((C*x^2 + B*x + A)*(b*x + a)^m*(d*x + c)^n, x)
```

Sympy [F(-2)]

Exception generated.

$$\int (a+bx)^m(c+dx)^n \left(A+Bx+Cx^2\right)\,dx = \text{Exception raised: HeuristicGCDFailed}$$

```
[In] integrate((b*x+a)**m*(d*x+c)**n*(C*x**2+B*x+A),x)
```

```
[Out] Exception raised: HeuristicGCDFailed >> no luck
```

Maxima [F]

$$\int (a+bx)^m (c+dx)^n \left(A+Bx+Cx^2\right) \, dx = \int (Cx^2+Bx+A)(bx+a)^m (dx+c)^n \, dx$$

```
[In] integrate((b*x+a)^m*(d*x+c)^n*(C*x^2+B*x+A),x, algorithm="maxima")
[Out] integrate((C*x^2 + B*x + A)*(b*x + a)^m*(d*x + c)^n, x)
```

Giac [F]

$$\int (a+bx)^m (c+dx)^n \left(A+Bx+Cx^2\right) \, dx = \int (Cx^2+Bx+A)(bx+a)^m (dx+c)^n \, dx$$

```
[In] integrate((b*x+a)^m*(d*x+c)^n*(C*x^2+B*x+A),x, algorithm="giac")
[Out] integrate((C*x^2 + B*x + A)*(b*x + a)^m*(d*x + c)^n, x)
```

Mupad [F(-1)]

Timed out.

$$\int (a+bx)^m (c+dx)^n \left(A+Bx+Cx^2\right) \, dx = \int (a+b\,x)^m\,(c+d\,x)^n\,\left(C\,x^2+B\,x+A\right) \, dx$$

```
[In] int((a + b*x)^m*(c + d*x)^n*(A + B*x + C*x^2),x)
[Out] int((a + b*x)^m*(c + d*x)^n*(A + B*x + C*x^2), x)
```

3.34 $\int (a+bx)^m (c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) dx$

Optimal result

Integrand size = 30, antiderivative size = 610

$$\int (a+bx)^m (c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) dx$$

$$= \frac{(a^2d^2D(m^2+m(8+3n)+3(6+5n+n^2))+b^2(c^2D(6+5m+m^2)-cCd(2+m)(4+m+n)+Bd^2(12}{b^3d^3(2+}$$

$$-\frac{(adD(9+2m+3n)+b(cD(3+m)-Cd(4+m+n)))(a+bx)^{2+m}(c+dx)^{1+n}}{b^3d^2(3+m+n)(4+m+n)}$$

$$+\frac{D(a+bx)^{3+m}(c+dx)^{1+n}}{b^3d(4+m+n)}$$

$$+\frac{(d(2+m+n)\left(a^3d^2D(1+n)(6+m+2n)+ab^2c(2+m)(cD(3+m)-Cd(4+m+n))+Ab^3d^2(12+\right.}{}$$

```
[Out] (a^2*d^2*D*(m^2+m*(8+3*n)+3*n^2+15*n+18)+b^2*(c^2*D*(m^2+5*m+6)-c*C*d*(2+m)
*(4+m+n)+B*d^2*(12+m^2+7*n+n^2+m*(7+2*n)))+a*b*d*(c*D*(2+m)*(6+m+3*n)-C*d*(
m^2+m*(8+3*n)+2*n^2+12*n+16)))*(b*x+a)^(1+m)*(d*x+c)^(1+n)/b^3/d^3/(2+m+n)/
(3+m+n)/(4+m+n)-(a*d*D*(9+2*m+3*n)+b*(c*D*(3+m)-C*d*(4+m+n)))*(b*x+a)^(2+m)
*(d*x+c)^(1+n)/b^3/d^2/(3+m+n)/(4+m+n)+D*(b*x+a)^(3+m)*(d*x+c)^(1+n)/b^3/d/
(4+m+n)+(d*(2+m+n)*(a^3*d^2*D*(1+n)*(6+m+2*n)+a*b^2*c*(2+m)*(c*D*(3+m)-C*d*
(4+m+n))+A*b^3*d^2*(12+m^2+7*n+n^2+m*(7+2*n))-a^2*b*d*(C*d*(1+n)*(4+m+n)-c*
D*(2+m)*(6+m+3*n)))-(b*c*(1+m)+a*d*(1+n))*(a^2*d^2*D*(m^2+m*(8+3*n)+3*n^2+1
5*n+18)+b^2*(c^2*D*(m^2+5*m+6)-c*C*d*(2+m)*(4+m+n)+B*d^2*(12+m^2+7*n+n^2+m*
(7+2*n)))+a*b*d*(c*D*(2+m)*(6+m+3*n)-C*d*(m^2+m*(8+3*n)+2*n^2+12*n+16))))*(
b*x+a)^(1+m)*(d*x+c)^n*hypergeom([-n, 1+m],[2+m],-d*(b*x+a)/(-a*d+b*c))/b^4
/d^3/(1+m)/(2+m+n)/(3+m+n)/(4+m+n)/((b*(d*x+c)/(-a*d+b*c))^n)
```

Rubi [A] (verified)

Time = 0.71 (sec) , antiderivative size = 605, normalized size of antiderivative = 0.99, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}} = 0.167$, Rules used = {1637, 965, 81, 72, 71}

$$\int (a+bx)^m(c+dx)^n \left(A+Bx+Cx^2+Dx^3\right) \, dx$$

$$= \frac{(a+bx)^{m+1}(c+dx)^{n+1} (a^2d^2D(m^2+m(3n+8)+3(n^2+5n+6))+abd(cD(m+2)(m+3n+6)-C}{b^3d^3(m}$$

$$+ \frac{(a+bx)^{m+1}(c+dx)^n \left(\frac{b(c+dx)}{bc-ad}\right)^{-n} \text{Hypergeometric2F1}\left(m+1,-n,m+2,-\frac{d(a+bx)}{bc-ad}\right) \left(a^3d^2D(n+1)(m\right.}{}$$

$$- \frac{(a+bx)^{m+2}(c+dx)^{n+1}(adD(2m+3n+9)+bcD(m+3)-bCd(m+n+4))}{b^3d^2(m+n+3)(m+n+4)}$$

$$+ \frac{D(a+bx)^{m+3}(c+dx)^{n+1}}{b^3d(m+n+4)}$$

```
[In] Int[(a + b*x)^m*(c + d*x)^n*(A + B*x + C*x^2 + D*x^3),x]
```

```
[Out] ((a^2*d^2*D*(m^2 + m*(8 + 3*n) + 3*(6 + 5*n + n^2)) + b^2*(c^2*D*(6 + 5*m +
 m^2) - c*C*d*(2 + m)*(4 + m + n) + B*d^2*(12 + m^2 + 7*n + n^2 + m*(7 + 2*
n))) + a*b*d*(c*D*(2 + m)*(6 + m + 3*n) - C*d*(m^2 + m*(8 + 3*n) + 2*(8 + 6
*n + n^2))))*(a + b*x)^(1 + m)*(c + d*x)^(1 + n))/(b^3*d^3*(2 + m + n)*(3 +
 m + n)*(4 + m + n)) - ((b*c*D*(3 + m) - b*C*d*(4 + m + n) + a*d*D*(9 + 2*m
 + 3*n))*(a + b*x)^(2 + m)*(c + d*x)^(1 + n))/(b^3*d^2*(3 + m + n)*(4 + m +
 n)) + (D*(a + b*x)^(3 + m)*(c + d*x)^(1 + n))/(b^3*d*(4 + m + n)) + ((a^3*
d^2*D*(1 + n)*(6 + m + 2*n) + a*b^2*c*(2 + m)*(c*D*(3 + m) - C*d*(4 + m + n
)) + A*b^3*d^2*(12 + m^2 + 7*n + n^2 + m*(7 + 2*n)) - a^2*b*d*(C*d*(1 + n)*
(4 + m + n) - c*D*(2 + m)*(6 + m + 3*n)) - ((b*c*(1 + m) + a*d*(1 + n))*(a^
2*d^2*D*(m^2 + m*(8 + 3*n) + 3*(6 + 5*n + n^2)) + b^2*(c^2*D*(6 + 5*m + m^2
) - c*C*d*(2 + m)*(4 + m + n) + B*d^2*(12 + m^2 + 7*n + n^2 + m*(7 + 2*n)))
 + a*b*d*(c*D*(2 + m)*(6 + m + 3*n) - C*d*(m^2 + m*(8 + 3*n) + 2*(8 + 6*n +
 n^2)))))/(d*(2 + m + n)))*(a + b*x)^(1 + m)*(c + d*x)^n*Hypergeometric2F1[
1 + m, -n, 2 + m, -((d*(a + b*x))/(b*c - a*d))])/(b^4*d^2*(1 + m)*(3 + m +
n)*(4 + m + n)*((b*(c + d*x))/(b*c - a*d))^n)
```

Rule 71

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[((
a + b*x)^(m + 1)/(b*(m + 1)*(b/(b*c - a*d))^n))*Hypergeometric2F1[-n, m + 1
, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m, n}, x]
&& NeQ[b*c - a*d, 0] &&  !IntegerQ[m] &&  !IntegerQ[n] && GtQ[b/(b*c - a*d)
, 0] && (RationalQ[m] ||  !(RationalQ[n] && GtQ[-d/(b*c - a*d), 0]))
```

Rule 72

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Dist[(c
 + d*x)^FracPart[n]/((b/(b*c - a*d))^IntPart[n]*(b*((c + d*x)/(b*c - a*d)))
^FracPart[n]), Int[(a + b*x)^m*Simp[b*(c/(b*c - a*d)) + b*d*(x/(b*c - a*d))
, x]^n, x], x] /; FreeQ[{a, b, c, d, m, n}, x] && NeQ[b*c - a*d, 0] &&  !In
tegerQ[m] &&  !IntegerQ[n] && (RationalQ[m] ||  !SimplerQ[n + 1, m + 1])
```

Rule 81

```
Int[((a_.) + (b_.)*(x_))*((c_.) + (d_.)*(x_))^(n_.)*((e_.) + (f_.)*(x_))^(p
_.), x_Symbol] :> Simp[b*(c + d*x)^(n + 1)*((e + f*x)^(p + 1)/(d*f*(n + p +
 2))), x] + Dist[(a*d*f*(n + p + 2) - b*(d*e*(n + 1) + c*f*(p + 1)))/(d*f*(
n + p + 2)), Int[(c + d*x)^n*(e + f*x)^p, x], x] /; FreeQ[{a, b, c, d, e, f
, n, p}, x] && NeQ[n + p + 2, 0]
```

Rule 965

```
Int[((d_.) + (e_.)*(x_))^(m_)*((f_.) + (g_.)*(x_))^(n_)*((a_.) + (b_.)*(x_)
 + (c_.)*(x_)^2)^(p_.), x_Symbol] :> Simp[c^p*(d + e*x)^(m + 2*p)*((f + g*x
)^(n + 1)/(g*e^(2*p)*(m + n + 2*p + 1))), x] + Dist[1/(g*e^(2*p)*(m + n + 2
*p + 1)), Int[(d + e*x)^m*(f + g*x)^n*ExpandToSum[g*(m + n + 2*p + 1)*(e^(2
*p)*(a + b*x + c*x^2)^p - c^p*(d + e*x)^(2*p)) - c^p*(e*f - d*g)*(m + 2*p)*
(d + e*x)^(2*p - 1), x], x], x] /; FreeQ[{a, b, c, d, e, f, g}, x] && NeQ[e
*f - d*g, 0] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IGt
Q[p, 0] && NeQ[m + n + 2*p + 1, 0] && (IntegerQ[n] ||  !IntegerQ[m])
```

Rule 1637

```
Int[(Px_)*((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol]
:> With[{q = Expon[Px, x], k = Coeff[Px, x, Expon[Px, x]]}, Simp[k*(a + b*x
)^(m + q)*((c + d*x)^(n + 1)/(d*b^q*(m + n + q + 1))), x] + Dist[1/(d*b^q*(
m + n + q + 1)), Int[(a + b*x)^m*(c + d*x)^n*ExpandToSum[d*b^q*(m + n + q +
 1)*Px - d*k*(m + n + q + 1)*(a + b*x)^q - k*(b*c - a*d)*(m + q)*(a + b*x)^
(q - 1), x], x], x] /; NeQ[m + n + q + 1, 0]] /; FreeQ[{a, b, c, d, m, n},
x] && PolyQ[Px, x] && GtQ[Expon[Px, x], 2]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \frac{D(a+bx)^{3+m}(c+dx)^{1+n}}{b^3d(4+m+n)} \\ &\quad + \frac{\int (a+bx)^m(c+dx)^n\left(Ab^3d(4+m+n) - a^2D(bc(3+m)+ad(1+n)) - b(2abcD(3+m) - b^2Bd(4+\right.}{b^3d(4+m}\end{aligned}$$

$$= -\frac{(bcD(3+m) - bCd(4+m+n) + adD(9+2m+3n))(a+bx)^{2+m}(c+dx)^{1+n}}{b^3d^2(3+m+n)(4+m+n)}$$

$$+ \frac{D(a+bx)^{3+m}(c+dx)^{1+n}}{b^3d(4+m+n)}$$

$$+ \frac{\int (a+bx)^m(c+dx)^n \left(b^2(a^3d^2D(1+n)(6+m+2n) + ab^2c(2+m)(cD(3+m) - Cd(4+m+\right.}{}$$

$$= \frac{(a^2d^2D(m^2+m(8+3n)+3(6+5n+n^2)) + b^2(c^2D(6+5m+m^2) - cCd(2+m)(4+m+n) +}{}$$

$$- \frac{(bcD(3+m) - bCd(4+m+n) + adD(9+2m+3n))(a+bx)^{2+m}(c+dx)^{1+n}}{b^3d^2(3+m+n)(4+m+n)}$$

$$+ \frac{D(a+bx)^{3+m}(c+dx)^{1+n}}{b^3d(4+m+n)}$$

$$+ \frac{\Big(a^3d^2D(1+n)(6+m+2n) + ab^2c(2+m)(cD(3+m) - Cd(4+m+n)) + Ab^3d^2(12+m^2+}{}$$

$$= \frac{(a^2d^2D(m^2+m(8+3n)+3(6+5n+n^2)) + b^2(c^2D(6+5m+m^2) - cCd(2+m)(4+m+n) +}{}$$

$$- \frac{(bcD(3+m) - bCd(4+m+n) + adD(9+2m+3n))(a+bx)^{2+m}(c+dx)^{1+n}}{b^3d^2(3+m+n)(4+m+n)}$$

$$+ \frac{D(a+bx)^{3+m}(c+dx)^{1+n}}{b^3d(4+m+n)}$$

$$+ \frac{\Bigg(\Big(a^3d^2D(1+n)(6+m+2n) + ab^2c(2+m)(cD(3+m) - Cd(4+m+n)) + Ab^3d^2(12+m^2}{}$$

$$= \frac{(a^2d^2D(m^2+m(8+3n)+3(6+5n+n^2)) + b^2(c^2D(6+5m+m^2) - cCd(2+m)(4+m+n) +}{}$$

$$- \frac{(bcD(3+m) - bCd(4+m+n) + adD(9+2m+3n))(a+bx)^{2+m}(c+dx)^{1+n}}{b^3d^2(3+m+n)(4+m+n)}$$

$$+ \frac{D(a+bx)^{3+m}(c+dx)^{1+n}}{b^3d(4+m+n)}$$

$$+ \frac{\Big(a^3d^2D(1+n)(6+m+2n) + ab^2c(2+m)(cD(3+m) - Cd(4+m+n)) + Ab^3d^2(12+m^2+}{}$$

Mathematica [A] (verified)

Time = 0.24 (sec) , antiderivative size = 254, normalized size of antiderivative = 0.42

$$\int (a+bx)^m(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx$$
$$=\frac{(a+bx)^{1+m}(c+dx)^n\left(\frac{b(c+dx)}{bc-ad}\right)^{-n}\left((bc-ad)^3D\,\text{Hypergeometric2F1}\left(1+m,-3-n,2+m,\frac{d(a+bx)}{-bc+ad}\right)+b(}{}$$

```
[In] Integrate[(a + b*x)^m*(c + d*x)^n*(A + B*x + C*x^2 + D*x^3),x]
```

```
[Out] ((a + b*x)^(1 + m)*(c + d*x)^n*((b*c - a*d)^3*D*Hypergeometric2F1[1 + m, -3
 - n, 2 + m, (d*(a + b*x))/(-(b*c) + a*d)] + b*(b*c - a*d)^2*(C*d - 3*c*D)*
Hypergeometric2F1[1 + m, -2 - n, 2 + m, (d*(a + b*x))/(-(b*c) + a*d)] + b^2
*(b*c - a*d)*(-2*c*C*d + B*d^2 + 3*c^2*D)*Hypergeometric2F1[1 + m, -1 - n,
2 + m, (d*(a + b*x))/(-(b*c) + a*d)] + b^3*(c^2*C*d - B*c*d^2 + A*d^3 - c^3
*D)*Hypergeometric2F1[1 + m, -n, 2 + m, (d*(a + b*x))/(-(b*c) + a*d)]))/(b^
4*d^3*(1 + m)*((b*(c + d*x))/(b*c - a*d))^n)
```

Maple [F]

$$\int (bx+a)^m\,(dx+c)^n\left(Dx^3+C\,x^2+Bx+A\right)dx$$

```
[In] int((b*x+a)^m*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x)
```

```
[Out] int((b*x+a)^m*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x)
```

Fricas [F]

$$\int (a+bx)^m(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx$$
$$=\int \left(capitalDx^3+Cx^2+Bx+A\right)(bx+a)^m(dx+c)^n\,dx$$

```
[In] integrate((b*x+a)^m*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="fricas")
```

```
[Out] integral((D*x^3 + C*x^2 + B*x + A)*(b*x + a)^m*(d*x + c)^n, x)
```

Sympy [F(-2)]

Exception generated.

$$\int (a+bx)^m(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx=\text{Exception raised: HeuristicGCDFailed}$$

```
[In] integrate((b*x+a)**m*(d*x+c)**n*(D*x**3+C*x**2+B*x+A),x)
[Out] Exception raised: HeuristicGCDFailed >> no luck
```

Maxima [F]

$$\int (a+bx)^m(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx$$
$$=\int \left(Dx^3+Cx^2+Bx+A\right)(bx+a)^m(dx+c)^n\,dx$$

```
[In] integrate((b*x+a)^m*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="maxima")
[Out] integrate((D*x^3 + C*x^2 + B*x + A)*(b*x + a)^m*(d*x + c)^n, x)
```

Giac [F]

$$\int (a+bx)^m(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx$$
$$=\int \left(Dx^3+Cx^2+Bx+A\right)(bx+a)^m(dx+c)^n\,dx$$

```
[In] integrate((b*x+a)^m*(d*x+c)^n*(D*x^3+C*x^2+B*x+A),x, algorithm="giac")
[Out] integrate((D*x^3 + C*x^2 + B*x + A)*(b*x + a)^m*(d*x + c)^n, x)
```

Mupad [F(-1)]

Timed out.

$$\int (a+bx)^m(c+dx)^n\left(A+Bx+Cx^2+Dx^3\right)\,dx$$
$$=\int (a+b\,x)^m\,(c+d\,x)^n\,\left(A+B\,x+C\,x^2+x^3\,D\right)\,dx$$

```
[In] int((a + b*x)^m*(c + d*x)^n*(A + B*x + C*x^2 + x^3*D),x)
[Out] int((a + b*x)^m*(c + d*x)^n*(A + B*x + C*x^2 + x^3*D), x)
```

CHAPTER 4

APPENDIX

4.1 Listing of Grading functions

The following are the current version of the grading functions used for grading the quality of the antiderivative with reference to the optimal antiderivative included in the test suite.

There is a version for Maple and for Mathematica/Rubi. There is a version for grading Sympy and version for use with Sagemath.

The following are links to the current source code.

The following are the listings of source code of the grading functions.

Mathematica and Rubi grading function

```
(* Original version thanks to Albert Rich emailed on 03/21/2017 *)
(* ::Package:: *)

(* Nasser: April 7,2022. add second output which gives reason for the grade *)
(*                       Small rewrite of logic in main function to make it*)
(*                       match Maple's logic. No change in functionality otherwise*)

(* ::Subsection:: *)
(*GradeAntiderivative[result,optimal]*)



(* ::Text:: *)
(*If result and optimal are mathematical expressions, *)
```

```
(*          GradeAntiderivative[result,optimal] returns*)
(*  "F" if the result fails to integrate an expression that*)
(*      is integrable*)
(*  "C" if result involves higher level functions than necessary*)
(*  "B" if result is more than twice the size of the optimal*)
(*      antiderivative*)
(*  "A" if result can be considered optimal*)


GradeAntiderivative[result_,optimal_] := Module[{expnResult,expnOptimal,leafCountResult,leafCou
    expnResult  = ExpnType[result];
    expnOptimal = ExpnType[optimal];
    leafCountResult = LeafCount[result];
    leafCountOptimal = LeafCount[optimal];

    (*Print["expnResult=",expnResult," expnOptimal=",expnOptimal];*)
    If[expnResult<=expnOptimal,
        If[Not[FreeQ[result,Complex]], (*result contains complex*)
            If[Not[FreeQ[optimal,Complex]], (*optimal contains complex*)
                If[leafCountResult<=2*leafCountOptimal,
                    finalresult={"A",""}
                    ,(*ELSE*)
                    finalresult={"B","Both result and optimal contain complex but leaf count i
                  ]
            ,(*ELSE*)
                finalresult={"C","Result contains complex when optimal does not."}
            ]
        ,(*ELSE*)(*result does not contains complex*)
            If[leafCountResult<=2*leafCountOptimal,
               finalresult={"A",""}
            ,(*ELSE*)
               finalresult={"B","Leaf count is larger than twice the leaf count of optimal. $"
              ]
        ]
    ,(*ELSE*) (*expnResult>expnOptimal*)
        If[FreeQ[result,Integrate] && FreeQ[result,Int],
            finalresult={"C","Result contains higher order function than in optimal. Order "<>
            ,
            finalresult={"F","Contains unresolved integral."}
        ]
    ];

    finalresult
]

(* ::Text:: *)
(*The following summarizes the type number assigned an *)
```

```
(*expression based on the functions it involves*)
(*1 = rational function*)
(*2 = algebraic function*)
(*3 = elementary function*)
(*4 = special function*)
(*5 = hyperpergeometric function*)
(*6 = appell function*)
(*7 = rootsum function*)
(*8 = integrate function*)
(*9 = unknown function*)


ExpnType[expn_] :=
  If[AtomQ[expn],
    1,
  If[ListQ[expn],
    Max[Map[ExpnType,expn]],
  If[Head[expn]===Power,
    If[IntegerQ[expn[[2]]],
      ExpnType[expn[[1]]],
    If[Head[expn[[2]]]===Rational,
      If[IntegerQ[expn[[1]]] || Head[expn[[1]]]===Rational,
        1,
      Max[ExpnType[expn[[1]]],2]],
    Max[ExpnType[expn[[1]]],ExpnType[expn[[2]]],3]]],
  If[Head[expn]===Plus || Head[expn]===Times,
    Max[ExpnType[First[expn]],ExpnType[Rest[expn]]],
  If[ElementaryFunctionQ[Head[expn]],
    Max[3,ExpnType[expn[[1]]]],
  If[SpecialFunctionQ[Head[expn]],
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],4]],
  If[HypergeometricFunctionQ[Head[expn]],
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],5]],
  If[AppellFunctionQ[Head[expn]],
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],6]],
  If[Head[expn]===RootSum,
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],7]],
  If[Head[expn]===Integrate || Head[expn]===Int,
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],8]],
  9]]]]]]]]]]]


ElementaryFunctionQ[func_] :=
  MemberQ[{
  Exp,Log,
  Sin,Cos,Tan,Cot,Sec,Csc,
  ArcSin,ArcCos,ArcTan,ArcCot,ArcSec,ArcCsc,
```

```
  Sinh,Cosh,Tanh,Coth,Sech,Csch,
  ArcSinh,ArcCosh,ArcTanh,ArcCoth,ArcSech,ArcCsch
},func]


SpecialFunctionQ[func_] :=
  MemberQ[{
  Erf, Erfc, Erfi,
  FresnelS, FresnelC,
  ExpIntegralE, ExpIntegralEi, LogIntegral,
  SinIntegral, CosIntegral, SinhIntegral, CoshIntegral,
  Gamma, LogGamma, PolyGamma,
  Zeta, PolyLog, ProductLog,
  EllipticF, EllipticE, EllipticPi
},func]


HypergeometricFunctionQ[func_] :=
  MemberQ[{Hypergeometric1F1,Hypergeometric2F1,HypergeometricPFQ},func]


AppellFunctionQ[func_] :=
  MemberQ[{AppellF1},func]
```

Maple grading function

```
# File: GradeAntiderivative.mpl
# Original version thanks to Albert Rich emailed on 03/21/2017

#Nasser 03/22/2017  Use Maple leaf count instead since buildin
#Nasser 03/23/2017  missing 'ln' for ElementaryFunctionQ added
#Nasser 03/24/2017  corrected the check for complex result
#Nasser 10/27/2017  check for leafsize and do not call ExpnType()
#                   if leaf size is "too large". Set at 500,000
#Nasser 12/22/2019  Added debug flag, added 'dilog' to special functions
#                   see problem 156, file Apostol_Problems
#Nasser 4/07/2022   add second output which gives reason for the grade

GradeAntiderivative := proc(result,optimal)
local leaf_count_result,
        leaf_count_optimal,
        ExpnType_result,
        ExpnType_optimal,
        debug:=false;

        leaf_count_result:=leafcount(result);
```

```
        #do NOT call ExpnType() if leaf size is too large. Recursion problem
        if leaf_count_result > 500000 then
            return "B","result has leaf size over 500,000. Avoiding possible recursion issues
        fi;

        leaf_count_optimal := leafcount(optimal);
        ExpnType_result  := ExpnType(result);
        ExpnType_optimal := ExpnType(optimal);

        if debug then
            print("ExpnType_result",ExpnType_result," ExpnType_optimal=",ExpnType_optimal);
        fi;

# If result and optimal are mathematical expressions,
#  GradeAntiderivative[result,optimal] returns
#   "F" if the result fails to integrate an expression that
#       is integrable
#   "C" if result involves higher level functions than necessary
#   "B" if result is more than twice the size of the optimal
#       antiderivative
#   "A" if result can be considered optimal

    #This check below actually is not needed, since I only
    #call this grading only for passed integrals. i.e. I check
    #for "F" before calling this. But no harm of keeping it here.
    #just in case.


    if not type(result,freeof('int')) then
        return "F","Result contains unresolved integral";
    fi;


    if ExpnType_result<=ExpnType_optimal then
        if debug then
             print("ExpnType_result<=ExpnType_optimal");
        fi;
        if is_contains_complex(result) then
            if is_contains_complex(optimal) then
                if debug then
                        print("both result and optimal complex");
                fi;
                if leaf_count_result<=2*leaf_count_optimal then
                    return "A"," ";
                else
                   return "B",cat("Both result and optimal contain complex but leaf count of
                                   convert(leaf_count_result,string)," vs. $2 (",
```

```
                                   convert(leaf_count_optimal,string)," ) = ",convert(2*leaf_c
                end if
            else #result contains complex but optimal is not
                if debug then
                        print("result contains complex but optimal is not");
                fi;
                return "C","Result contains complex when optimal does not.";
            fi;
        else # result do not contain complex
             # this assumes optimal do not as well. No check is needed here.
            if debug then
                   print("result do not contain complex, this assumes optimal do not as well")
            fi;
            if leaf_count_result<=2*leaf_count_optimal then
                if debug then
                    print("leaf_count_result<=2*leaf_count_optimal");
                fi;
                return "A"," ";
            else
                if debug then
                    print("leaf_count_result>2*leaf_count_optimal");
                fi;
                return "B",cat("Leaf count of result is larger than twice the leaf count of op
                                   convert(leaf_count_result,string),"$ vs. $2(",
                                   convert(leaf_count_optimal,string),")=",convert(2*leaf_coun
            fi;
        fi;
    else #ExpnType(result) > ExpnType(optimal)
        if debug then
            print("ExpnType(result) > ExpnType(optimal)");
        fi;
        return "C",cat("Result contains higher order function than in optimal. Order ",
                       convert(ExpnType_result,string)," vs. order ",
                       convert(ExpnType_optimal,string),".");
    fi;

end proc:

#
# is_contains_complex(result)
# takes expressions and returns true if it contains "I" else false
#
#Nasser 032417
is_contains_complex:= proc(expression)
  return (has(expression,I));
end proc:
```

```
# The following summarizes the type number assigned an expression
# based on the functions it involves
# 1 = rational function
# 2 = algebraic function
# 3 = elementary function
# 4 = special function
# 5 = hyperpergeometric function
# 6 = appell function
# 7 = rootsum function
# 8 = integrate function
# 9 = unknown function

ExpnType := proc(expn)
  if type(expn,'atomic') then
    1
  elif type(expn,'list') then
    apply(max,map(ExpnType,expn))
  elif type(expn,'sqrt') then
    if type(op(1,expn),'rational') then
       1
    else
       max(2,ExpnType(op(1,expn)))
    end if
  elif type(expn,'`^`') then
    if type(op(2,expn),'integer') then
      ExpnType(op(1,expn))
    elif type(op(2,expn),'rational') then
      if type(op(1,expn),'rational') then
         1
      else
         max(2,ExpnType(op(1,expn)))
      end if
    else
         max(3,ExpnType(op(1,expn)),ExpnType(op(2,expn)))
    end if
  elif type(expn,'`+`') or type(expn,'`*`') then
    max(ExpnType(op(1,expn)),max(ExpnType(rest(expn))))
  elif ElementaryFunctionQ(op(0,expn)) then
    max(3,ExpnType(op(1,expn)))
  elif SpecialFunctionQ(op(0,expn)) then
    max(4,apply(max,map(ExpnType,[op(expn)])))
  elif HypergeometricFunctionQ(op(0,expn)) then
    max(5,apply(max,map(ExpnType,[op(expn)])))
  elif AppellFunctionQ(op(0,expn)) then
    max(6,apply(max,map(ExpnType,[op(expn)])))
  elif op(0,expn)='int' then
    max(8,apply(max,map(ExpnType,[op(expn)]))) else
```

```
  9
  end if
end proc:


ElementaryFunctionQ := proc(func)
  member(func,[
        exp,log,ln,
        sin,cos,tan,cot,sec,csc,
        arcsin,arccos,arctan,arccot,arcsec,arccsc,
        sinh,cosh,tanh,coth,sech,csch,
        arcsinh,arccosh,arctanh,arccoth,arcsech,arccsch])
end proc:

SpecialFunctionQ := proc(func)
  member(func,[
        erf,erfc,erfi,
        FresnelS,FresnelC,
        Ei,Ei,Li,Si,Ci,Shi,Chi,
        GAMMA,lnGAMMA,Psi,Zeta,polylog,dilog,LambertW,
        EllipticF,EllipticE,EllipticPi])
end proc:

HypergeometricFunctionQ := proc(func)
  member(func,[Hypergeometric1F1,hypergeom,HypergeometricPFQ])
end proc:

AppellFunctionQ := proc(func)
  member(func,[AppellF1])
end proc:

# u is a sum or product.  rest(u) returns all but the
# first term or factor of u.
rest := proc(u) local v;
  if nops(u)=2 then
     op(2,u)
  else
     apply(op(0,u),op(2..nops(u),u))
  end if
end proc:

#leafcount(u) returns the number of nodes in u.
#Nasser 3/23/17  Replaced by build-in leafCount from package in Maple
leafcount := proc(u)
   MmaTranslator[Mma][LeafCount](u);
end proc:
```

Sympy grading function

```
#Dec 24, 2019. Nasser M. Abbasi:
#             Port of original Maple grading function by
#             Albert Rich to use with Sympy/Python
#Dec 27, 2019  Nasser. Added `RootSum`. See problem 177, Timofeev file
#             added 'exp_polar'
from sympy import *

def leaf_count(expr):
    #sympy do not have leaf count function. This is approximation
    return round(1.7*count_ops(expr))

def is_sqrt(expr):
    if isinstance(expr,Pow):
        if expr.args[1] == Rational(1,2):
            return True
        else:
            return False
    else:
        return False

def is_elementary_function(func):
    return func in [exp,log,ln,sin,cos,tan,cot,sec,csc,
            asin,acos,atan,acot,asec,acsc,sinh,cosh,tanh,coth,sech,csch,
            asinh,acosh,atanh,acoth,asech,acsch
        ]

def is_special_function(func):
    return func in [ erf,erfc,erfi,
             fresnels,fresnelc,Ei,Ei,Li,Si,Ci,Shi,Chi,
             gamma,loggamma,digamma,zeta,polylog,LambertW,
             elliptic_f,elliptic_e,elliptic_pi,exp_polar
        ]

def is_hypergeometric_function(func):
    return func in [hyper]

def is_appell_function(func):
    return func in [appellf1]

def is_atom(expn):
    try:
        if expn.isAtom or isinstance(expn,int) or isinstance(expn,float):
           return True
        else:
           return False
```

```
    except AttributeError as error:
        return False

def expnType(expn):
    debug=False
    if debug:
        print("expn=",expn,"type(expn)=",type(expn))

    if is_atom(expn):
       return 1
    elif isinstance(expn,list):
        return max(map(expnType, expn))   #apply(max,map(ExpnType,expn))
    elif  is_sqrt(expn):
        if isinstance(expn.args[0],Rational): #type(op(1,expn),'rational')
            return 1
        else:
            return max(2,expnType(expn.args[0]))  #max(2,ExpnType(op(1,expn)))
    elif isinstance(expn,Pow):   #type(expn,'`^`')
        if isinstance(expn.args[1],Integer):  #type(op(2,expn),'integer')
            return expnType(expn.args[0])   #ExpnType(op(1,expn))
        elif isinstance(expn.args[1],Rational):  #type(op(2,expn),'rational')
            if isinstance(expn.args[0],Rational): #type(op(1,expn),'rational')
                return 1
            else:
                return max(2,expnType(expn.args[0]))  #max(2,ExpnType(op(1,expn)))
        else:
            return max(3,expnType(expn.args[0]),expnType(expn.args[1])) #max(3,ExpnType(op(1,expn)),ExpnTy
    elif isinstance(expn,Add) or isinstance(expn,Mul): #type(expn,'`+`') or type(expn,'`*`')
        m1 = expnType(expn.args[0])
        m2 = expnType(list(expn.args[1:]))
        return max(m1,m2)  #max(ExpnType(op(1,expn)),max(ExpnType(rest(expn))))
    elif is_elementary_function(expn.func):  #ElementaryFunctionQ(op(0,expn))
        return max(3,expnType(expn.args[0]))  #max(3,ExpnType(op(1,expn)))
    elif is_special_function(expn.func): #SpecialFunctionQ(op(0,expn))
        m1 = max(map(expnType, list(expn.args)))
        return max(4,m1)   #max(4,apply(max,map(ExpnType,[op(expn)])))
    elif is_hypergeometric_function(expn.func): #HypergeometricFunctionQ(op(0,expn))
        m1 = max(map(expnType, list(expn.args)))
        return max(5,m1)   #max(5,apply(max,map(ExpnType,[op(expn)])))
    elif is_appell_function(expn.func):
        m1 = max(map(expnType, list(expn.args)))
        return max(6,m1)   #max(5,apply(max,map(ExpnType,[op(expn)])))
    elif isinstance(expn,RootSum):
        m1 = max(map(expnType, list(expn.args))) #Apply[Max,Append[Map[ExpnType,Apply[List,expn]],7]],
        return max(7,m1)
    elif str(expn).find("Integral") != -1:
```

```
        m1 = max(map(expnType, list(expn.args)))
        return max(8,m1)    #max(5,apply(max,map(ExpnType,[op(expn)])))
    else:
        return 9

#main function
def grade_antiderivative(result,optimal):

    #print ("Enter grade_antiderivative for sagemath")
    #print("Enter grade_antiderivative, result=",result," optimal=",optimal)

    leaf_count_result  = leaf_count(result)
    leaf_count_optimal = leaf_count(optimal)

    #print("leaf_count_result=",leaf_count_result)
    #print("leaf_count_optimal=",leaf_count_optimal)

    expnType_result  = expnType(result)
    expnType_optimal = expnType(optimal)

    if str(result).find("Integral") != -1:
        grade = "F"
        grade_annotation =""
    else:
        if expnType_result <= expnType_optimal:
            if result.has(I):
                if optimal.has(I): #both result and optimal complex
                    if leaf_count_result <= 2*leaf_count_optimal:
                        grade = "A"
                        grade_annotation =""
                    else:
                        grade = "B"
                        grade_annotation ="Both result and optimal contain complex but leaf count of result is larg
                else: #result contains complex but optimal is not
                    grade = "C"
                    grade_annotation ="Result contains complex when optimal does not."
            else: # result do not contain complex, this assumes optimal do not as well
                if leaf_count_result <= 2*leaf_count_optimal:
                    grade = "A"
                    grade_annotation =""
                else:
                    grade = "B"
                    grade_annotation ="Leaf count of result is larger than twice the leaf count of optimal. "+str(le
        else:
            grade = "C"
            grade_annotation ="Result contains higher order function than in optimal. Order "+str(ExpnType_
```

```
    #print("Before returning. grade=",grade, " grade_annotation=",grade_annotation)

    return grade, grade_annotation
```

SageMath grading function

```
#Dec 24, 2019. Nasser: Ported original Maple grading function by
#          Albert Rich to use with Sagemath. This is used to
#          grade Fricas, Giac and Maxima results.
#Dec 24, 2019. Nasser: Added 'exp_integral_e' and 'sng', 'sin_integral'
#          'arctan2','floor','abs','log_integral'
#June 4, 2022  Made default grade_annotation "none" instead of "" due
#          issue later when reading the file.
#July 14, 2022. Added ellipticF. This is until they fix sagemath, then remove it.

from sage.all import *
from sage.symbolic.operators import add_vararg, mul_vararg

debug=False;

def tree_size(expr):
    r"""
    Return the tree size of this expression.
    """
    #print("Enter tree_size, expr is ",expr)

    if expr not in SR:
        # deal with lists, tuples, vectors
        return 1 + sum(tree_size(a) for a in expr)
    expr = SR(expr)
    x, aa = expr.operator(), expr.operands()
    if x is None:
        return 1
    else:
        return 1 + sum(tree_size(a) for a in aa)

def is_sqrt(expr):
    if expr.operator() == operator.pow:   #isinstance(expr,Pow):
        if expr.operands()[1]==1/2: #expr.args[1] == Rational(1,2):
            if debug: print ("expr is sqrt")
            return True
        else:
            return False
    else:
        return False
```

```
def is_elementary_function(func):
    #debug=False
    m = func.name() in ['exp','log','ln',
            'sin','cos','tan','cot','sec','csc',
            'arcsin','arccos','arctan','arccot','arcsec','arccsc',
            'sinh','cosh','tanh','coth','sech','csch',
            'arcsinh','arccosh','arctanh','arccoth','arcsech','arccsch','sgn',
        'arctan2','floor','abs'
        ]
    if debug:
        if m:
            print ("func ", func , " is elementary_function")
        else:
            print ("func ", func , " is NOT elementary_function")


    return m

def is_special_function(func):
    #debug=False
    if debug:
        print ("type(func)=", type(func))

    m= func.name() in ['erf','erfc','erfi','fresnel_sin','fresnel_cos','Ei',
           'Ei','Li','Si','sin_integral','Ci','cos_integral','Shi','sinh_integral'
           'Chi','cosh_integral','gamma','log_gamma','psi,zeta',
           'polylog','lambert_w','elliptic_f','elliptic_e','ellipticF',
           'elliptic_pi','exp_integral_e','log_integral']

    if debug:
        print ("m=",m)
        if m:
            print ("func ", func ," is special_function")
        else:
            print ("func ", func ," is NOT special_function")


    return m


def is_hypergeometric_function(func):
    return func.name() in ['hypergeometric','hypergeometric_M','hypergeometric_U']

def is_appell_function(func):
    return func.name() in ['hypergeometric']   #[appellf1] can't find this in sagemath
```

```
def is_atom(expn):

   #debug=False
   if debug:
       print ("Enter is_atom, expn=",expn)

   if not hasattr(expn, 'parent'):
      return False


   #thanks to answer at https://ask.sagemath.org/question/49179/what-is-sagemath-equivalent-to-atomic-t
   try:
      if expn.parent() is SR:
         return expn.operator() is None
      if expn.parent() in (ZZ, QQ, AA, QQbar):
         return expn in expn.parent() # Should always return True
      if hasattr(expn.parent(),"base_ring") and hasattr(expn.parent(),"gens"):
         return expn in expn.parent().base_ring() or expn in expn.parent().gens()

      return False

   except AttributeError as error:
      print("Exception,AttributeError in  is_atom")
      print ("cought exception" , type(error).__name__ )
      return False


def expnType(expn):

   if debug:
      print (">>>>>Enter expnType, expn=", expn)
      print (">>>>>is_atom(expn)=", is_atom(expn))

   if is_atom(expn):
      return 1
   elif type(expn)==list:   #isinstance(expn,list):
      return max(map(expnType, expn))   #apply(max,map(ExpnType,expn))
   elif  is_sqrt(expn):
      if  type(expn.operands()[0])==Rational: #type(isinstance(expn.args[0],Rational):
         return 1
      else:
         return max(2,expnType(expn.operands()[0]))   #max(2,expnType(expn.args[0]))
   elif expn.operator() == operator.pow:   #isinstance(expn,Pow)
      if type(expn.operands()[1])==Integer:   #isinstance(expn.args[1],Integer)
         return expnType(expn.operands()[0])   #expnType(expn.args[0])
      elif type(expn.operands()[1])==Rational:   #isinstance(expn.args[1],Rational)
         if type(expn.operands()[0])==Rational:  #isinstance(expn.args[0],Rational)
```

```
                return 1
            else:
                return max(2,expnType(expn.operands()[0]))  #max(2,expnType(expn.args[0]))
        else:
            return max(3,expnType(expn.operands()[0]),expnType(expn.operands()[1])) #max(3,expnType(expn.
    elif expn.operator() == add_vararg or expn.operator() == mul_vararg: #isinstance(expn,Add) or isinst
        m1 = expnType(expn.operands()[0]) #expnType(expn.args[0])
        m2 = expnType(expn.operands()[1:]) #expnType(list(expn.args[1:]))
        return max(m1,m2)  #max(ExpnType(op(1,expn)),max(ExpnType(rest(expn))))
    elif is_elementary_function(expn.operator()):  #is_elementary_function(expn.func)
        return max(3,expnType(expn.operands()[0]))
    elif is_special_function(expn.operator()): #is_special_function(expn.func)
        m1 = max(map(expnType, expn.operands()))       #max(map(expnType, list(expn.args)))
        return max(4,m1)    #max(4,m1)
    elif is_hypergeometric_function(expn.operator()): #is_hypergeometric_function(expn.func)
        m1 = max(map(expnType, expn.operands()))        #max(map(expnType, list(expn.args)))
        return max(5,m1)    #max(5,m1)
    elif is_appell_function(expn.operator()):
        m1 = max(map(expnType, expn.operands()))        #max(map(expnType, list(expn.args)))
        return max(6,m1)    #max(6,m1)
    elif str(expn).find("Integral") != −1: #this will never happen, since it
                #is checked before calling the grading function that is passed.
                #but kept it here.
        m1 = max(map(expnType, expn.operands()))        #max(map(expnType, list(expn.args)))
        return max(8,m1)    #max(5,apply(max,map(ExpnType,[op(expn)])))
    else:
        return 9

#main function
def grade_antiderivative(result,optimal):


    if debug:
        print ("Enter grade_antiderivative for sagemath")
        print("Enter grade_antiderivative, result=",result)
        print("Enter grade_antiderivative, optimal=",optimal)
        print("type(anti)=",type(result))
        print("type(optimal)=",type(optimal))

    leaf_count_result  = tree_size(result) #leaf_count(result)
    leaf_count_optimal = tree_size(optimal) #leaf_count(optimal)

    #if debug: print ("leaf_count_result=", leaf_count_result, "leaf_count_optimal=",leaf_count_optimal)



    expnType_result  = expnType(result)
    expnType_optimal = expnType(optimal)
```

```
if debug: print ("expnType_result=", expnType_result, "expnType_optimal=",expnType_optimal)

if expnType_result <= expnType_optimal:
    if result.has(I):
        if optimal.has(I): #both result and optimal complex
            if leaf_count_result <= 2*leaf_count_optimal:
                grade = "A"
                grade_annotation ="none"
            else:
                grade = "B"
                grade_annotation ="Both result and optimal contain complex but leaf count of result is larger th
        else: #result contains complex but optimal is not
            grade = "C"
            grade_annotation ="Result contains complex when optimal does not."
    else: # result do not contain complex, this assumes optimal do not as well
        if leaf_count_result <= 2*leaf_count_optimal:
            grade = "A"
            grade_annotation ="none"
        else:
            grade = "B"
            grade_annotation ="Leaf count of result is larger than twice the leaf count of optimal. "+str(leaf_
else:
    grade = "C"
    grade_annotation ="Result contains higher order function than in optimal. Order "+str(expnType_resul


print("Before returning. grade=",grade, " grade_annotation=",grade_annotation)

return grade, grade_annotation
```