Computer algebra independent integration tests

1-Algebraic-functions/1.1-Binomial-products/1.1.3-General/1.1.3.6-g-x-^m-a+b-x^n-^p-c+d-x^n-^q-e+f-x^n-^r

Nasser M. Abbasi

September 20, 2021 Compiled on September 20, 2021 at 5:22am

Contents

Chapter 1

Introduction

Local contents

This report gives the result of running the computer algebra independent integration test. Elementary Algebraic integrals version.

The download section below contains links to download the problems in plain text format used for all CAS systems.

The number of integrals in this report is [12]. This is test number [15].

1.1 Listing of CAS systems tested

The following systems were tested at this time.

1. Mathematica 12.3.1 (64 bit) on windows 10.
2. Rubi 4.16.1 in Mathematica 12.3.1 on windows 10.
3. Maple 2021.1 (64 bit) on windows 10.
4. Maxima 5.45 on Linux. (via sagemath 9.3)
5. Fricas 1.3.7 on Linux (via sagemath 9.3)
6. Giac/Xcas 1.7 on Linux. (via sagemath 9.3)
7. Sympy 1.8 under Python 3.8.8 using Anaconda distribution on Ubuntu.
8. Mupad using Matlab 2021a with Symbolic Math Toolbox Version 8.7 under windows 10 (64 bit)
9. `IntegrateAlgebraic` under Mathematica 12.3.1 on windows 10. `https://github.com/stblake/algebraic_integration`. September 15, 2021 version.

Maxima, Fricas and Giac/Xcas were called from inside SageMath. This was done using SageMath integrate command by changing the name of the algorithm to use the different CAS systems.

Sympy was called directly using Python.

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 (12)	0.00 (0)
Mathematica	100.00 (12)	0.00 (0)
Fricas	100.00 (12)	0.00 (0)
Maple	100.00 (12)	0.00 (0)
Mupad	100.00 (12)	0.00 (0)
Maxima	100.00 (12)	0.00 (0)
Giac	91.67 (11)	8.33 (1)
Sympy	25.00 (3)	% 75.00 (9)
IntegrateAlgebraic	0.00 (0)	100.00 (12)

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.

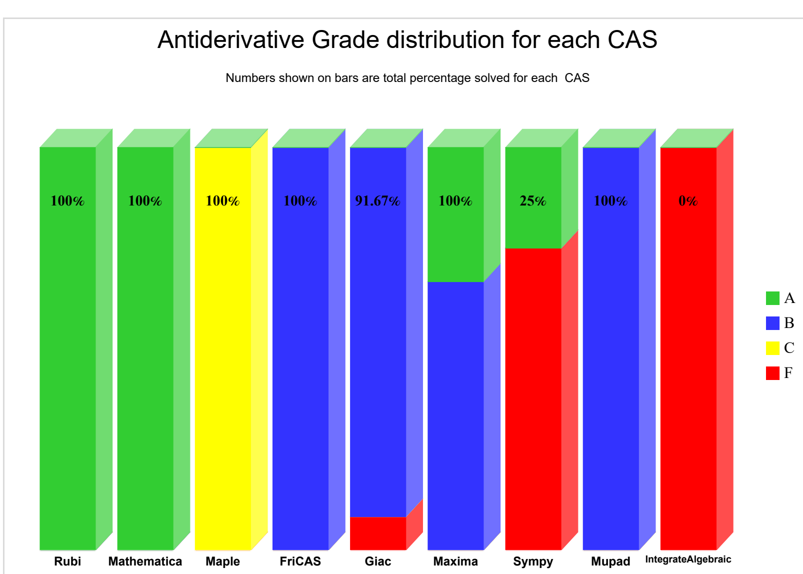
Table 1.2: Description of grading applied to integration result

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

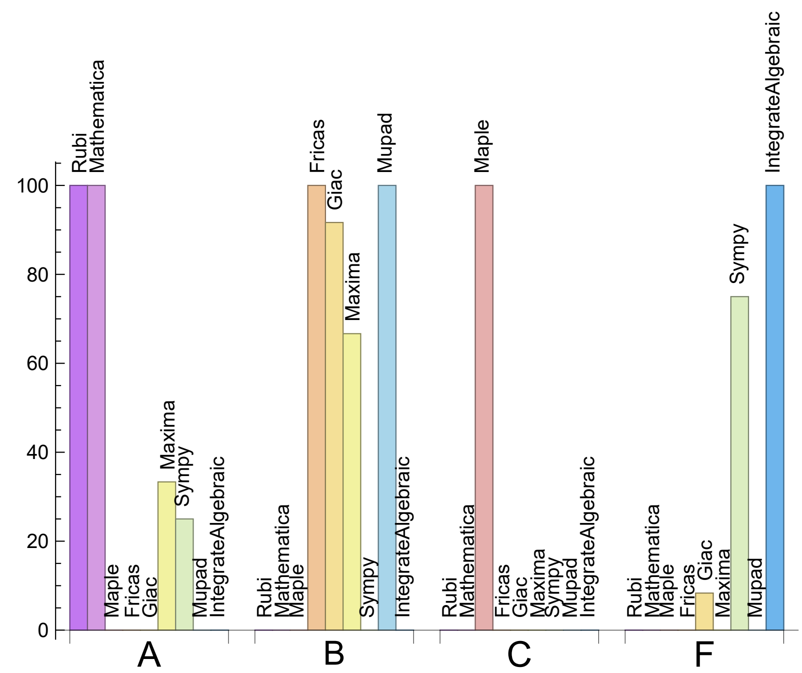
System	% A grade	% B grade	% C grade	% F grade
Rubi	100.00	0.00	0.00	0.00
Mathematica	100.00	0.00	0.00	0.00
Maxima	33.33	66.67	0.00	0.00
Sympy	25.00	0.00	0.00	75.00
Fricas	0.00	100.00	0.00	0.00
Maple	0.00	0.00	100.00	0.00
IntegrateAlgebraic	0.00	0.00	0.00	100.00
Mupad	N/A	100.00	0.00	0.00
Giac	0.00	91.67	0.00	8.33

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 CAS systems for each grade level.



The following table shows the distribution of the different types of failure for each CAS. There are 3 types of reasons why it can fail. The first is when CAS returns back the input within the time limit, which means it could not solve it. This the typical normal failure **F** .

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

The third is due to an exception generated. Assigned **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 CAS. This type of error requires more investigations to determine the cause.

System	Number failed	Percentage normal failure	Percentage time-out failure	Percentage exception failure
Rubi	0	0.00 %	0.00 %	0.00 %
Mathematica	0	0.00 %	0.00 %	0.00 %
Maple	0	0.00 %	0.00 %	0.00 %
Fricas	0	0.00 %	0.00 %	0.00 %
IntegrateAlgebraic	12	100.00 %	0.00 %	0.00 %
Giac	1	0.00 %	100.00 %	0.00 %
Maxima	0	0.00 %	0.00 %	0.00 %
Sympy	9	0.00 %	100.00 %	0.00 %
Mupad	0	0.00 %	0.00 %	0.00 %

Table 1.4: Failure statistics for each CAS

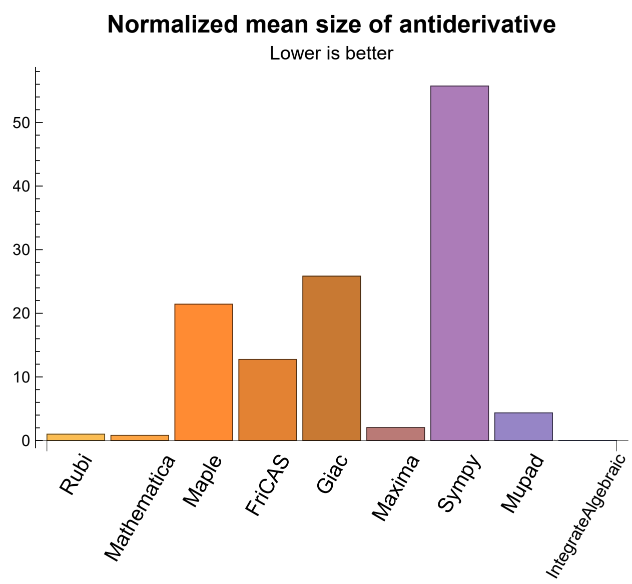
1.3 Performance

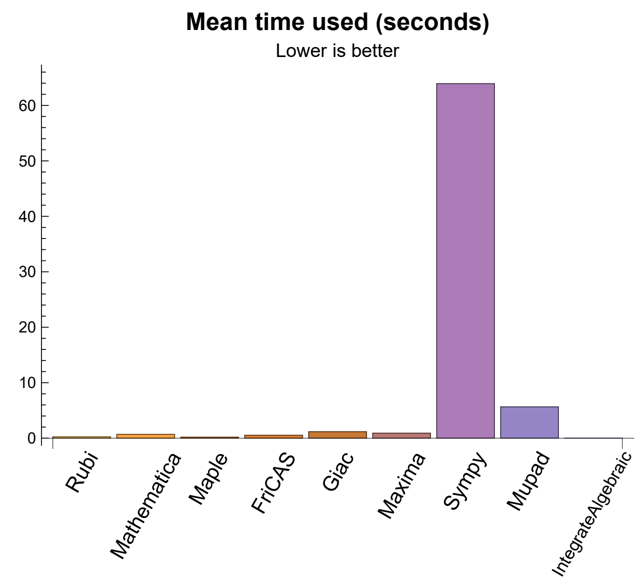
The table below summarizes the performance of each CAS system in terms of CPU time and leaf size of results.

System	Mean time (sec)	Mean size	Normalized mean	Median size	Normalized median
Rubi	0.25	202.33	1.00	185.00	1.00
Mathematica	0.69	167.83	0.81	150.50	0.81
Maple	0.18	5656.75	21.43	3691.00	19.37
Maxima	0.90	443.75	2.04	398.00	2.14
Fricas	0.52	3297.67	12.75	2178.50	11.51
Sympy	63.92	5532.33	55.72	6399.00	62.74
Giac	1.16	5856.45	25.84	3415.00	21.34
Mupad	5.65	1031.33	4.35	838.50	4.42
IntegrateAlgebraic	0.00	0.00	0.00	0.00	0.00

Table 1.5: Time and leaf size performance for each CAS

The following are bar charts for the normalized leafsize and time used columns from the above table.




Mean time used (seconds)
Lower is better
60
50
40
30
20
10
0
Rubi
Mathematica
Maple
FriCAS
Giac
Maxima
Sympy
Mupad
IntegrateAlgebraic

1.4 list of integrals that has no closed form antiderivative

{}

1.5 list of integrals solved by CAS but has no known antiderivative

Rubi {}

Mathematica {}

Maple {}

Maxima {}

Fricas {}

Sympy {}

Giac {}

Mupad {}

IntegrateAlgebraic {}

1.6 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 mean necessarily 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 easier to do further investigation to determine why it was not possible to verify the result produced.

Rubi {}

Mathematica {}

IntegrateAlgebraic {}

Maple Verification phase not implemented yet.

Maxima Verification phase not implemented yet.

Fricas Verification phase not implemented yet.

Sympy Verification phase not implemented yet.

Giac Verification phase not implemented yet.

Mupad Verification phase not implemented yet.

1.7 Timing

The command `AboluteTiming[]` 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 has 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 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 is not counted in the final statistics.

1.8 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 produced was correct.

Verification phase has 3 minutes time out. An integral whose result was not verified could still be correct. Further investigation is needed on those integrals which failed verifications. Such integrals are marked in the summary table below and also in each integral separate section so they are easy to identify and locate.

1.9 Important notes about some of the results

1.9.1 Important note about Maxima results

Since these integrals are run in a batch mode, using an automated script, and by using sagemath (SageMath uses Maxima), then any integral where Maxima needs an interactive response from the user to answer a question during evaluation of the integral in order to complete the integration, will fail and is counted as failed.

The exception raised is `ValueError`. Therefore Maxima result below is lower than what could result if Maxima was run directly and each question Maxima asks 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 pecrentage can be higher or lower depending on the specific input test file.

Such integrals can be indentified by looking at the output of the integration in each section for Maxima. The exception message will indicate of the error is due to the interactive question being asked or not.

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 loading of Maxima `abs_integrate` was found to cause some problem. 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.

1.9.2 Important note about FriCAS and Giac/XCAS results

There are Few integrals which failed due to SageMath not able to translate the result back to SageMath syntax and not because these CAS system were not able to do the integrations.

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.

Hopefully the next version of SageMath will have complete translation of FriCAS and XCAS syntax and I will re-run all the tests again when this happens.

1.9.3 Important note about finding leaf size of antiderivative

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

The other CAS systems (SageMath and Sympy) do not have special buildin function for this purpose at this time. Therefore the leaf size for Fricas and Sympy and Giac antiderivatives is 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 is called directly from Python, the following code is 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
```

1.9.4 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.

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.

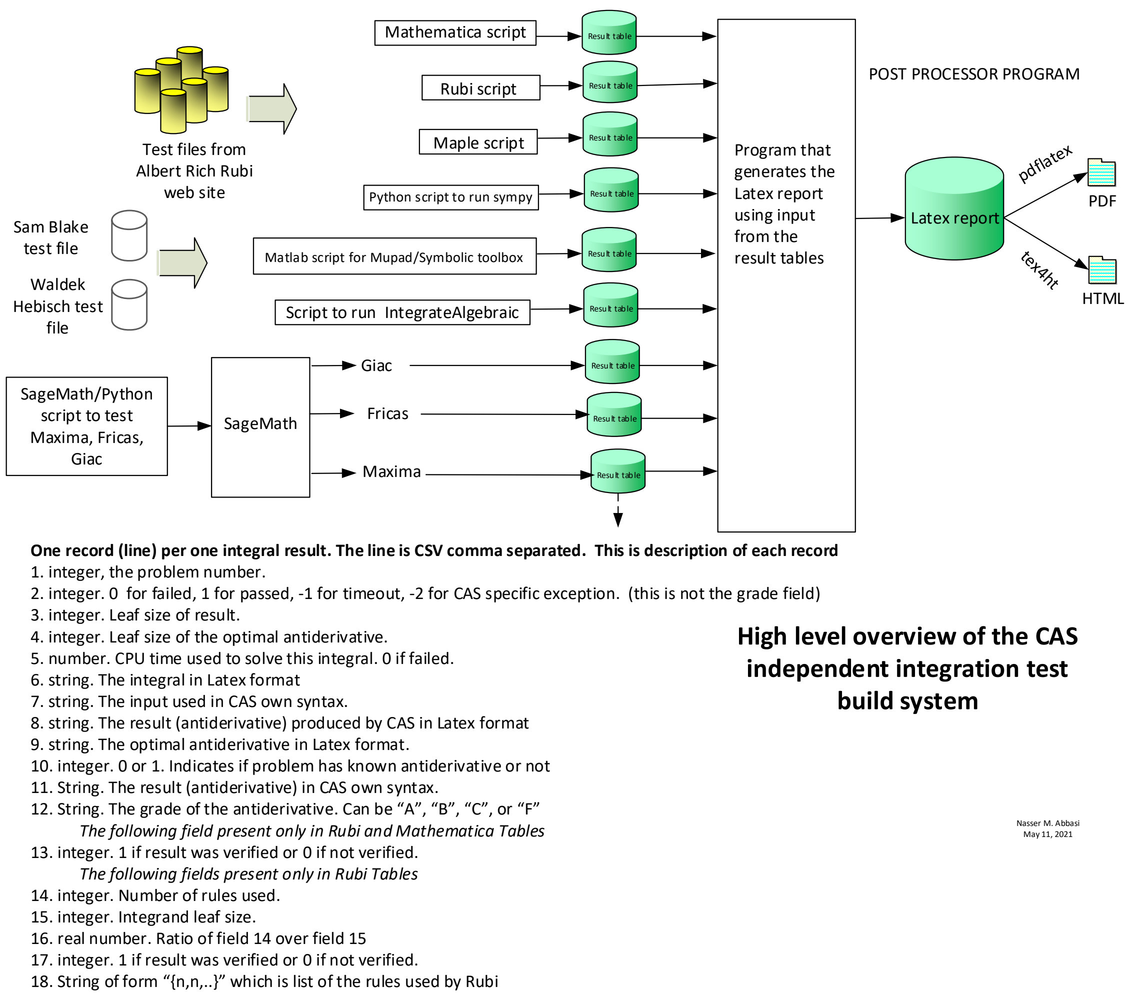
The following is an example of using Matlab's symbolic toolbox (Mupad) to solve an integral

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

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

1.10 Design of the test system

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



Chapter 2

detailed summary tables of results

Local contents

2.1 List of integrals sorted by grade for each CAS

Local contents

2.1.1 Rubi

A grade: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 }

B grade: { }

C grade: { }

F grade: { }

2.1.2 Mathematica

A grade: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 }

B grade: { }

C grade: { }

F grade: { }

2.1.3 Maple

A grade: { }

B grade: { }

C grade: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 }

F grade: { }

2.1.4 Maxima

A grade: { 3, 4, 8, 12 }

B grade: { 1, 2, 5, 6, 7, 9, 10, 11 }

C grade: { }

F grade: { }

2.1.5 FriCAS

A grade: { }

B grade: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 }

C grade: { }

F grade: { }

2.1.6 Sympy

A grade: { 3, 4, 8 }

B grade: { }

C grade: { }

F grade: { 1, 2, 5, 6, 7, 9, 10, 11, 12 }

2.1.7 Giac

A grade: { }

B grade: { 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12 }

C grade: { }

F grade: { 9 }

2.1.8 Mupad

A grade: { }

B grade: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 }

C grade: { }

F grade: { }

2.1.9 IntegrateAlgebraic

A grade: { }

B grade: { }

C grade: { }

F grade: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 }

2.2 Detailed conclusion table per each integral for all CAS systems

Detailed conclusion table per each integral is given by 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 just an F.

In this table,the column `N.S.` in the table below, which stands for **normalized size** is defined as $\frac{\text{antiderivative leaf size}}{\text{optimal antiderivative leaf size}}$. To help make the table fit, `Mathematica` was abbreviated to `MMA` and `IntegrateAlgebraic` to `I.A.`

Problem 1	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad	I.A.
grade	A	A	A	C	B	B	F(-1)	B	B	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD	Yes
size	210	210	172	4972	464	3073	0	6927	1089	0
N.S.	1	1.00	0.82	23.68	2.21	14.63	0.00	32.99	5.19	0.00
time (sec)	N/A	0.270	0.994	0.211	0.936	0.578	0.000	1.191	5.644	0.653

Problem 2	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad	I.A.
grade	A	A	A	C	B	B	F(-1)	B	B	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD	Yes
size	160	160	129	2410	332	1524	0	3415	588	0
N.S.	1	1.00	0.81	15.06	2.08	9.52	0.00	21.34	3.68	0.00
time (sec)	N/A	0.176	0.525	0.138	0.861	0.490	0.000	0.809	5.231	0.168

Problem 3	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad	I.A.
grade	A	A	A	C	A	B	A	B	B	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD	Yes
size	108	108	84	891	200	562	8500	1290	271	0
N.S.	1	1.00	0.78	8.25	1.85	5.20	78.70	11.94	2.51	0.00
time (sec)	N/A	0.084	0.247	0.109	0.697	0.452	88.266	1.796	4.958	0.114

Problem 4	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad	I.A.
grade	A	A	A	C	A	B	A	B	B	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD	Yes
size	66	66	49	262	91	185	1698	327	91	0
N.S.	1	1.00	0.74	3.97	1.38	2.80	25.73	4.95	1.38	0.00
time (sec)	N/A	0.040	0.067	0.117	0.598	0.440	29.325	0.455	4.830	0.067

Problem 5	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad	I.A.
grade	A	A	A	C	B	B	F(-1)	B	B	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD	Yes
size	318	318	273	11389	748	6638	0	15358	1882	0
N.S.	1	1.00	0.86	35.81	2.35	20.87	0.00	48.30	5.92	0.00
time (sec)	N/A	0.411	1.478	0.226	1.119	0.584	0.000	2.013	6.348	0.865

Problem 6	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad	I.A.
grade	A	A	A	C	B	B	F(-1)	B	B	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD	Yes
size	237	237	199	5908	540	3515	0	8103	1119	0
N.S.	1	1.00	0.84	24.93	2.28	14.83	0.00	34.19	4.72	0.00
time (sec)	N/A	0.310	0.605	0.171	0.905	0.506	0.000	2.006	5.591	0.303

Problem 7	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad	I.A.
grade	A	A	A	C	B	B	F(-1)	B	B	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD	Yes
size	160	160	129	2410	332	1426	0	3415	588	0
N.S.	1	1.00	0.81	15.06	2.08	8.91	0.00	21.34	3.68	0.00
time (sec)	N/A	0.172	0.313	0.144	0.782	0.484	0.000	0.818	5.196	0.161

Problem 8	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad	I.A.
grade	A	A	A	C	A	B	A	B	B	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD	Yes
size	102	102	78	732	155	527	6399	1023	265	0
N.S.	1	1.00	0.76	7.18	1.52	5.17	62.74	10.03	2.60	0.00
time (sec)	N/A	0.076	0.175	0.109	0.634	0.450	74.164	0.582	5.113	0.097

Problem 9	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad	I.A.
grade	A	A	A	C	B	B	F(-1)	F(-1)	B	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD	Yes
size	410	410	358	20937	1032	11628	0	0	2949	0
N.S.	1	1.00	0.87	51.07	2.52	28.36	0.00	0.00	7.19	0.00
time (sec)	N/A	0.624	1.430	0.344	1.467	0.735	0.000	0.000	7.495	1.251

Problem 10	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad	I.A.
grade	A	A	A	C	B	B	F(-1)	B	B	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD	Yes
size	310	310	265	11389	748	6557	0	15358	1882	0
N.S.	1	1.00	0.85	36.74	2.41	21.15	0.00	49.54	6.07	0.00
time (sec)	N/A	0.414	1.520	0.249	0.923	0.576	0.000	1.303	6.414	1.011

Problem 11	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad	I.A.
grade	A	A	A	C	B	B	F(-1)	B	B	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD	Yes
size	210	210	172	4972	464	2833	0	6927	1089	0
N.S.	1	1.00	0.82	23.68	2.21	13.49	0.00	32.99	5.19	0.00
time (sec)	N/A	0.258	0.753	0.178	1.008	0.505	0.000	1.057	5.653	0.645

Problem 12	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad	I.A.
grade	A	A	A	C	A	B	F(-1)	B	B	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD	Yes
size	137	137	106	1609	219	1104	0	2278	563	0
N.S.	1	1.00	0.77	11.74	1.60	8.06	0.00	16.63	4.11	0.00
time (sec)	N/A	0.111	0.177	0.126	0.845	0.466	0.000	0.750	5.307	0.111

2.3 Detailed conclusion table specific for Rubi results

The following table is specific to Rubi. 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 given. The larger this ratio is, the harder the integral was to solve. In this test, problem number [4] had the largest ratio of [.1500]

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	12	3	1.00	29	0.103
2	A	10	3	1.00	29	0.103
3	A	8	3	1.00	27	0.111
4	A	6	3	1.00	20	0.150
5	A	14	3	1.00	31	0.097
6	A	12	3	1.00	31	0.097
7	A	10	3	1.00	29	0.103
8	A	8	3	1.00	22	0.136
9	A	16	3	1.00	31	0.097
10	A	14	3	1.00	31	0.097
11	A	12	3	1.00	29	0.103
12	A	10	3	1.00	22	0.136

Chapter 3

Listing of integrals

Local contents

3.1 $\int (ex)^m (a+bx^n)^3 (A+Bx^n)(c+dx^n)\, dx$

Optimal. Leaf size=210

$$\frac{a^3Ac(ex)^{m+1}}{e(m+1)}+\frac{a^2x^{n+1}(ex)^m(aAd+aBc+3Abc)}{m+n+1}+\frac{b^2x^{4n+1}(ex)^m(3aBd+Abd+bBc)}{m+4n+1}+\frac{ax^{2n+1}(ex)^m(3Ab(ad+bc)+}{m+2n+1}$$

Rubi [A] time = 0.27, antiderivative size = 210, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 3, integrand size = 29, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.103, Rules used = {570, 20, 30}

$$\frac{a^2x^{n+1}(ex)^m(aAd+aBc+3Abc)}{m+n+1}+\frac{a^3Ac(ex)^{m+1}}{e(m+1)}+\frac{b^2x^{4n+1}(ex)^m(3aBd+Abd+bBc)}{m+4n+1}+\frac{ax^{2n+1}(ex)^m(3Ab(ad+bc)+aB(ad+3bc))}{m+2n+1}+\frac{bx^{3n+1}(ex)^m(Ab(3ad+bc)+3aB(ad+bc))}{m+3n+1}+\frac{b^3Bdx^{5n+1}(ex)^m}{m+5n+1}$$

Antiderivative was successfully verified.

```
[In] Int[(e*x)^m*(a + b*x^n)^3*(A + B*x^n)*(c + d*x^n),x]
```

```
[Out] (a^2*(3*A*b*c + a*B*c + a*A*d)*x^(1 + n)*(e*x)^m)/(1 + m + n) + (a*(3*A*b*(
b*c + a*d) + a*B*(3*b*c + a*d))*x^(1 + 2*n)*(e*x)^m)/(1 + m + 2*n) + (b*(3*
a*B*(b*c + a*d) + A*b*(b*c + 3*a*d))*x^(1 + 3*n)*(e*x)^m)/(1 + m + 3*n) + (
b^2*(b*B*c + A*b*d + 3*a*B*d)*x^(1 + 4*n)*(e*x)^m)/(1 + m + 4*n) + (b^3*B*d
*x^(1 + 5*n)*(e*x)^m)/(1 + m + 5*n) + (a^3*A*c*(e*x)^(1 + m))/(e*(1 + m))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 30

```
Int[(x_)^(m_.), x_Symbol] :> Simp[x^(m + 1)/(m + 1), x] /; FreeQ[m, x] && N
eQ[m, -1]
```

Rule 570

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

Rubi steps

$$\begin{aligned}\int (ex)^m (a+bx^n)^3 (A+Bx^n)(c+dx^n)\, dx &= \int \left(a^3Ac(ex)^m+a^2(3Abc+aBc+aAd)x^n(ex)^m+a(3Ab(bc+ad)+\right.\\ &= \frac{a^3Ac(ex)^{1+m}}{e(1+m)}+\left(b^3Bd\right)\int x^{5n}(ex)^m\, dx+\left(a^2(3Abc+aBc+aAd)\right)\int\\ &= \frac{a^3Ac(ex)^{1+m}}{e(1+m)}+\left(b^3Bdx^{-m}(ex)^m\right)\int x^{m+5n}\, dx+\left(a^2(3Abc+aBc+aA\right.\\ &= \frac{a^2(3Abc+aBc+aAd)x^{1+n}(ex)^m}{1+m+n}+\frac{a(3Ab(bc+ad)+aB(3bc+ad))}{1+m+2n}\end{aligned}$$

Mathematica [A] time = 0.99, size = 172, normalized size = 0.82

$$x(ex)^m\left(\frac{a^3Ac}{m+1}+\frac{a^2x^n(aAd+aBc+3Abc)}{m+n+1}+\frac{b^2x^{4n}(3aBd+Abd+bBc)}{m+4n+1}+\frac{ax^{2n}(3Ab(ad+bc)+aB(ad+3bc))}{m+2n+1}+\frac{bx^{3n}(Ab(3ad+bc)+3aB(ad+bc))}{m+3n+1}+\frac{b^3Bdx^{5n}}{m+5n+1}\right)$$

Antiderivative was successfully verified.

```
[In] Integrate[(e*x)^m*(a + b*x^n)^3*(A + B*x^n)*(c + d*x^n),x]
```

```
[Out] x*(e*x)^m*((a^3*A*c)/(1 + m) + (a^2*(3*A*b*c + a*B*c + a*A*d)*x^n)/(1 + m +
 n) + (a*(3*A*b*(b*c + a*d) + a*B*(3*b*c + a*d))*x^(2*n))/(1 + m + 2*n) + (
b*(3*a*B*(b*c + a*d) + A*b*(b*c + 3*a*d))*x^(3*n))/(1 + m + 3*n) + (b^2*(b*
B*c + A*b*d + 3*a*B*d)*x^(4*n))/(1 + m + 4*n) + (b^3*B*d*x^(5*n))/(1 + m +
5*n))
```

IntegrateAlgebraic [F] time = 0.65, size = 0, normalized size = 0.00

$$\int (ex)^m (a + bx^n)^3 (A + Bx^n) (c + dx^n) \, dx$$

Verification is not applicable to the result.

```
[In] IntegrateAlgebraic[(e*x)^m*(a + b*x^n)^3*(A + B*x^n)*(c + d*x^n),x]
```

```
[Out] Defer[IntegrateAlgebraic][(e*x)^m*(a + b*x^n)^3*(A + B*x^n)*(c + d*x^n), x]
```

fricas [B] time = 0.58, size = 3073, normalized size = 14.63

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)^3*(A+B*x^n)*(c+d*x^n),x, algorithm="fricas")
```

```
[Out] ((B*b^3*d*m^5 + 5*B*b^3*d*m^4 + 10*B*b^3*d*m^3 + 10*B*b^3*d*m^2 + 5*B*b^3*d
*m + B*b^3*d + 24*(B*b^3*d*m + B*b^3*d)*n^4 + 50*(B*b^3*d*m^2 + 2*B*b^3*d*m
 + B*b^3*d)*n^3 + 35*(B*b^3*d*m^3 + 3*B*b^3*d*m^2 + 3*B*b^3*d*m + B*b^3*d)*
n^2 + 10*(B*b^3*d*m^4 + 4*B*b^3*d*m^3 + 6*B*b^3*d*m^2 + 4*B*b^3*d*m + B*b^3
*d)*n)*x*x^(5*n)*e^(m*log(e) + m*log(x)) + ((B*b^3*c + (3*B*a*b^2 + A*b^3)*
d)*m^5 + B*b^3*c + 5*(B*b^3*c + (3*B*a*b^2 + A*b^3)*d)*m^4 + 30*(B*b^3*c +
(3*B*a*b^2 + A*b^3)*d + (B*b^3*c + (3*B*a*b^2 + A*b^3)*d)*m)*n^4 + 10*(B*b^
3*c + (3*B*a*b^2 + A*b^3)*d)*m^3 + 61*(B*b^3*c + (B*b^3*c + (3*B*a*b^2 + A*
b^3)*d)*m^2 + (3*B*a*b^2 + A*b^3)*d + 2*(B*b^3*c + (3*B*a*b^2 + A*b^3)*d)*m
)*n^3 + 10*(B*b^3*c + (3*B*a*b^2 + A*b^3)*d)*m^2 + 41*(B*b^3*c + (B*b^3*c +
 (3*B*a*b^2 + A*b^3)*d)*m^3 + 3*(B*b^3*c + (3*B*a*b^2 + A*b^3)*d)*m^2 + (3*
B*a*b^2 + A*b^3)*d + 3*(B*b^3*c + (3*B*a*b^2 + A*b^3)*d)*m)*n^2 + (3*B*a*b^
2 + A*b^3)*d + 5*(B*b^3*c + (3*B*a*b^2 + A*b^3)*d)*m + 11*(B*b^3*c + (B*b^3
*c + (3*B*a*b^2 + A*b^3)*d)*m^4 + 4*(B*b^3*c + (3*B*a*b^2 + A*b^3)*d)*m^3 +
 6*(B*b^3*c + (3*B*a*b^2 + A*b^3)*d)*m^2 + (3*B*a*b^2 + A*b^3)*d + 4*(B*b^3
*c + (3*B*a*b^2 + A*b^3)*d)*m)*n)*x*x^(4*n)*e^(m*log(e) + m*log(x)) + (((3*
B*a*b^2 + A*b^3)*c + 3*(B*a^2*b + A*a*b^2)*d)*m^5 + 5*((3*B*a*b^2 + A*b^3)*
c + 3*(B*a^2*b + A*a*b^2)*d)*m^4 + 40*((3*B*a*b^2 + A*b^3)*c + 3*(B*a^2*b +
 A*a*b^2)*d + ((3*B*a*b^2 + A*b^3)*c + 3*(B*a^2*b + A*a*b^2)*d)*m)*n^4 + 10
*((3*B*a*b^2 + A*b^3)*c + 3*(B*a^2*b + A*a*b^2)*d)*m^3 + 78*(((3*B*a*b^2 +
A*b^3)*c + 3*(B*a^2*b + A*a*b^2)*d)*m^2 + (3*B*a*b^2 + A*b^3)*c + 3*(B*a^2*
b + A*a*b^2)*d + 2*((3*B*a*b^2 + A*b^3)*c + 3*(B*a^2*b + A*a*b^2)*d)*m)*n^3
 + 10*((3*B*a*b^2 + A*b^3)*c + 3*(B*a^2*b + A*a*b^2)*d)*m^2 + 49*(((3*B*a*b
^2 + A*b^3)*c + 3*(B*a^2*b + A*a*b^2)*d)*m^3 + 3*((3*B*a*b^2 + A*b^3)*c + 3
*(B*a^2*b + A*a*b^2)*d)*m^2 + (3*B*a*b^2 + A*b^3)*c + 3*(B*a^2*b + A*a*b^2)
*d + 3*((3*B*a*b^2 + A*b^3)*c + 3*(B*a^2*b + A*a*b^2)*d)*m)*n^2 + (3*B*a*b^
2 + A*b^3)*c + 3*(B*a^2*b + A*a*b^2)*d + 5*((3*B*a*b^2 + A*b^3)*c + 3*(B*a^
2*b + A*a*b^2)*d)*m + 12*(((3*B*a*b^2 + A*b^3)*c + 3*(B*a^2*b + A*a*b^2)*d)
*m^4 + 4*((3*B*a*b^2 + A*b^3)*c + 3*(B*a^2*b + A*a*b^2)*d)*m^3 + 6*((3*B*a*
b^2 + A*b^3)*c + 3*(B*a^2*b + A*a*b^2)*d)*m^2 + (3*B*a*b^2 + A*b^3)*c + 3*(
B*a^2*b + A*a*b^2)*d + 4*((3*B*a*b^2 + A*b^3)*c + 3*(B*a^2*b + A*a*b^2)*d)*
m)*n)*x*x^(3*n)*e^(m*log(e) + m*log(x)) + ((3*(B*a^2*b + A*a*b^2)*c + (B*a^
3 + 3*A*a^2*b)*d)*m^5 + 5*(3*(B*a^2*b + A*a*b^2)*c + (B*a^3 + 3*A*a^2*b)*d)
*m^4 + 60*(3*(B*a^2*b + A*a*b^2)*c + (B*a^3 + 3*A*a^2*b)*d + (3*(B*a^2*b +
```

```
A*a*b^2)*c + (B*a^3 + 3*A*a^2*b)*d)*m)*n^4 + 10*(3*(B*a^2*b + A*a*b^2)*c +
(B*a^3 + 3*A*a^2*b)*d)*m^3 + 107*((3*(B*a^2*b + A*a*b^2)*c + (B*a^3 + 3*A*a
^2*b)*d)*m^2 + 3*(B*a^2*b + A*a*b^2)*c + (B*a^3 + 3*A*a^2*b)*d + 2*(3*(B*a^
2*b + A*a*b^2)*c + (B*a^3 + 3*A*a^2*b)*d)*m)*n^3 + 10*(3*(B*a^2*b + A*a*b^2
)*c + (B*a^3 + 3*A*a^2*b)*d)*m^2 + 59*((3*(B*a^2*b + A*a*b^2)*c + (B*a^3 +
3*A*a^2*b)*d)*m^3 + 3*(3*(B*a^2*b + A*a*b^2)*c + (B*a^3 + 3*A*a^2*b)*d)*m^2
 + 3*(B*a^2*b + A*a*b^2)*c + (B*a^3 + 3*A*a^2*b)*d + 3*(3*(B*a^2*b + A*a*b^
2)*c + (B*a^3 + 3*A*a^2*b)*d)*m)*n^2 + 3*(B*a^2*b + A*a*b^2)*c + (B*a^3 + 3
*A*a^2*b)*d + 5*(3*(B*a^2*b + A*a*b^2)*c + (B*a^3 + 3*A*a^2*b)*d)*m + 13*((
3*(B*a^2*b + A*a*b^2)*c + (B*a^3 + 3*A*a^2*b)*d)*m^4 + 4*(3*(B*a^2*b + A*a*
b^2)*c + (B*a^3 + 3*A*a^2*b)*d)*m^3 + 6*(3*(B*a^2*b + A*a*b^2)*c + (B*a^3 +
 3*A*a^2*b)*d)*m^2 + 3*(B*a^2*b + A*a*b^2)*c + (B*a^3 + 3*A*a^2*b)*d + 4*(3
*(B*a^2*b + A*a*b^2)*c + (B*a^3 + 3*A*a^2*b)*d)*m)*n)*x*x^(2*n)*e^(m*log(e)
 + m*log(x)) + ((A*a^3*d + (B*a^3 + 3*A*a^2*b)*c)*m^5 + A*a^3*d + 5*(A*a^3*
d + (B*a^3 + 3*A*a^2*b)*c)*m^4 + 120*(A*a^3*d + (B*a^3 + 3*A*a^2*b)*c + (A*
a^3*d + (B*a^3 + 3*A*a^2*b)*c)*m)*n^4 + 10*(A*a^3*d + (B*a^3 + 3*A*a^2*b)*c
)*m^3 + 154*(A*a^3*d + (A*a^3*d + (B*a^3 + 3*A*a^2*b)*c)*m^2 + (B*a^3 + 3*A
*a^2*b)*c + 2*(A*a^3*d + (B*a^3 + 3*A*a^2*b)*c)*m)*n^3 + 10*(A*a^3*d + (B*a
^3 + 3*A*a^2*b)*c)*m^2 + 71*(A*a^3*d + (A*a^3*d + (B*a^3 + 3*A*a^2*b)*c)*m^
3 + 3*(A*a^3*d + (B*a^3 + 3*A*a^2*b)*c)*m^2 + (B*a^3 + 3*A*a^2*b)*c + 3*(A*
a^3*d + (B*a^3 + 3*A*a^2*b)*c)*m)*n^2 + (B*a^3 + 3*A*a^2*b)*c + 5*(A*a^3*d
+ (B*a^3 + 3*A*a^2*b)*c)*m + 14*(A*a^3*d + (A*a^3*d + (B*a^3 + 3*A*a^2*b)*c
)*m^4 + 4*(A*a^3*d + (B*a^3 + 3*A*a^2*b)*c)*m^3 + 6*(A*a^3*d + (B*a^3 + 3*A
*a^2*b)*c)*m^2 + (B*a^3 + 3*A*a^2*b)*c + 4*(A*a^3*d + (B*a^3 + 3*A*a^2*b)*c
)*m)*n)*x*x^n*e^(m*log(e) + m*log(x)) + (A*a^3*c*m^5 + 120*A*a^3*c*n^5 + 5*
A*a^3*c*m^4 + 10*A*a^3*c*m^3 + 10*A*a^3*c*m^2 + 5*A*a^3*c*m + A*a^3*c + 274
*(A*a^3*c*m + A*a^3*c)*n^4 + 225*(A*a^3*c*m^2 + 2*A*a^3*c*m + A*a^3*c)*n^3
+ 85*(A*a^3*c*m^3 + 3*A*a^3*c*m^2 + 3*A*a^3*c*m + A*a^3*c)*n^2 + 15*(A*a^3*
c*m^4 + 4*A*a^3*c*m^3 + 6*A*a^3*c*m^2 + 4*A*a^3*c*m + A*a^3*c)*n)*x*e^(m*lo
g(e) + m*log(x)))/(m^6 + 120*(m + 1)*n^5 + 6*m^5 + 274*(m^2 + 2*m + 1)*n^4
+ 15*m^4 + 225*(m^3 + 3*m^2 + 3*m + 1)*n^3 + 20*m^3 + 85*(m^4 + 4*m^3 + 6*m
^2 + 4*m + 1)*n^2 + 15*m^2 + 15*(m^5 + 5*m^4 + 10*m^3 + 10*m^2 + 5*m + 1)*n
 + 6*m + 1)
```

giac [B] time = 1.19, size = 6927, normalized size = 32.99

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)^3*(A+B*x^n)*(c+d*x^n),x, algorithm="giac")
```

```
[Out] (B*b^3*d*m^5*x*x^m*x^(5*n)*e^m + 10*B*b^3*d*m^4*n*x*x^m*x^(5*n)*e^m + 35*B*
b^3*d*m^3*n^2*x*x^m*x^(5*n)*e^m + 50*B*b^3*d*m^2*n^3*x*x^m*x^(5*n)*e^m + 24
*B*b^3*d*m*n^4*x*x^m*x^(5*n)*e^m + B*b^3*c*m^5*x*x^m*x^(4*n)*e^m + 3*B*a*b^
2*d*m^5*x*x^m*x^(4*n)*e^m + A*b^3*d*m^5*x*x^m*x^(4*n)*e^m + 11*B*b^3*c*m^4*
n*x*x^m*x^(4*n)*e^m + 33*B*a*b^2*d*m^4*n*x*x^m*x^(4*n)*e^m + 11*A*b^3*d*m^4
*n*x*x^m*x^(4*n)*e^m + 41*B*b^3*c*m^3*n^2*x*x^m*x^(4*n)*e^m + 123*B*a*b^2*d
*m^3*n^2*x*x^m*x^(4*n)*e^m + 41*A*b^3*d*m^3*n^2*x*x^m*x^(4*n)*e^m + 61*B*b^
3*c*m^2*n^3*x*x^m*x^(4*n)*e^m + 183*B*a*b^2*d*m^2*n^3*x*x^m*x^(4*n)*e^m + 6
1*A*b^3*d*m^2*n^3*x*x^m*x^(4*n)*e^m + 30*B*b^3*c*m*n^4*x*x^m*x^(4*n)*e^m +
90*B*a*b^2*d*m*n^4*x*x^m*x^(4*n)*e^m + 30*A*b^3*d*m*n^4*x*x^m*x^(4*n)*e^m +
 3*B*a*b^2*c*m^5*x*x^m*x^(3*n)*e^m + A*b^3*c*m^5*x*x^m*x^(3*n)*e^m + 3*B*a^
2*b*d*m^5*x*x^m*x^(3*n)*e^m + 3*A*a*b^2*d*m^5*x*x^m*x^(3*n)*e^m + 36*B*a*b^
2*c*m^4*n*x*x^m*x^(3*n)*e^m + 12*A*b^3*c*m^4*n*x*x^m*x^(3*n)*e^m + 36*B*a^2
*b*d*m^4*n*x*x^m*x^(3*n)*e^m + 36*A*a*b^2*d*m^4*n*x*x^m*x^(3*n)*e^m + 147*B
*a*b^2*c*m^3*n^2*x*x^m*x^(3*n)*e^m + 49*A*b^3*c*m^3*n^2*x*x^m*x^(3*n)*e^m +
 147*B*a^2*b*d*m^3*n^2*x*x^m*x^(3*n)*e^m + 147*A*a*b^2*d*m^3*n^2*x*x^m*x^(3
*n)*e^m + 234*B*a*b^2*c*m^2*n^3*x*x^m*x^(3*n)*e^m + 78*A*b^3*c*m^2*n^3*x*x^
m*x^(3*n)*e^m + 234*B*a^2*b*d*m^2*n^3*x*x^m*x^(3*n)*e^m + 234*A*a*b^2*d*m^2
*n^3*x*x^m*x^(3*n)*e^m + 120*B*a*b^2*c*m*n^4*x*x^m*x^(3*n)*e^m + 40*A*b^3*c
```

*m*n^4*x*x^m*x^(3*n)*e^m + 120*B*a^2*b*d*m*n^4*x*x^m*x^(3*n)*e^m + 120*A*a*
b^2*d*m*n^4*x*x^m*x^(3*n)*e^m + 3*B*a^2*b*c*m^5*x*x^m*x^(2*n)*e^m + 3*A*a*b
^2*c*m^5*x*x^m*x^(2*n)*e^m + B*a^3*d*m^5*x*x^m*x^(2*n)*e^m + 3*A*a^2*b*d*m^
5*x*x^m*x^(2*n)*e^m + 39*B*a^2*b*c*m^4*n*x*x^m*x^(2*n)*e^m + 39*A*a*b^2*c*m
^4*n*x*x^m*x^(2*n)*e^m + 13*B*a^3*d*m^4*n*x*x^m*x^(2*n)*e^m + 39*A*a^2*b*d*
m^4*n*x*x^m*x^(2*n)*e^m + 177*B*a^2*b*c*m^3*n^2*x*x^m*x^(2*n)*e^m + 177*A*a
*b^2*c*m^3*n^2*x*x^m*x^(2*n)*e^m + 59*B*a^3*d*m^3*n^2*x*x^m*x^(2*n)*e^m + 1
77*A*a^2*b*d*m^3*n^2*x*x^m*x^(2*n)*e^m + 321*B*a^2*b*c*m^2*n^3*x*x^m*x^(2*n
)*e^m + 321*A*a*b^2*c*m^2*n^3*x*x^m*x^(2*n)*e^m + 107*B*a^3*d*m^2*n^3*x*x^m
*x^(2*n)*e^m + 321*A*a^2*b*d*m^2*n^3*x*x^m*x^(2*n)*e^m + 180*B*a^2*b*c*m*n^
4*x*x^m*x^(2*n)*e^m + 180*A*a*b^2*c*m*n^4*x*x^m*x^(2*n)*e^m + 60*B*a^3*d*m*
n^4*x*x^m*x^(2*n)*e^m + 180*A*a^2*b*d*m*n^4*x*x^m*x^(2*n)*e^m + B*a^3*c*m^5
*x*x^m*x^n*e^m + 3*A*a^2*b*c*m^5*x*x^m*x^n*e^m + A*a^3*d*m^5*x*x^m*x^n*e^m
+ 14*B*a^3*c*m^4*n*x*x^m*x^n*e^m + 42*A*a^2*b*c*m^4*n*x*x^m*x^n*e^m + 14*A*
a^3*d*m^4*n*x*x^m*x^n*e^m + 71*B*a^3*c*m^3*n^2*x*x^m*x^n*e^m + 213*A*a^2*b*
c*m^3*n^2*x*x^m*x^n*e^m + 71*A*a^3*d*m^3*n^2*x*x^m*x^n*e^m + 154*B*a^3*c*m^
2*n^3*x*x^m*x^n*e^m + 462*A*a^2*b*c*m^2*n^3*x*x^m*x^n*e^m + 154*A*a^3*d*m^2
*n^3*x*x^m*x^n*e^m + 120*B*a^3*c*m*n^4*x*x^m*x^n*e^m + 360*A*a^2*b*c*m*n^4*
x*x^m*x^n*e^m + 120*A*a^3*d*m*n^4*x*x^m*x^n*e^m + A*a^3*c*m^5*x*x^m*e^m + 1
5*A*a^3*c*m^4*n*x*x^m*e^m + 85*A*a^3*c*m^3*n^2*x*x^m*e^m + 225*A*a^3*c*m^2*
n^3*x*x^m*e^m + 274*A*a^3*c*m*n^4*x*x^m*e^m + 120*A*a^3*c*n^5*x*x^m*e^m + 5
*B*b^3*d*m^4*x*x^m*x^(5*n)*e^m + 40*B*b^3*d*m^3*n*x*x^m*x^(5*n)*e^m + 105*B
*b^3*d*m^2*n^2*x*x^m*x^(5*n)*e^m + 100*B*b^3*d*m*n^3*x*x^m*x^(5*n)*e^m + 24
*B*b^3*d*n^4*x*x^m*x^(5*n)*e^m + 5*B*b^3*c*m^4*x*x^m*x^(4*n)*e^m + 15*B*a*b
^2*d*m^4*x*x^m*x^(4*n)*e^m + 5*A*b^3*d*m^4*x*x^m*x^(4*n)*e^m + 44*B*b^3*c*m
^3*n*x*x^m*x^(4*n)*e^m + 132*B*a*b^2*d*m^3*n*x*x^m*x^(4*n)*e^m + 44*A*b^3*d
*m^3*n*x*x^m*x^(4*n)*e^m + 123*B*b^3*c*m^2*n^2*x*x^m*x^(4*n)*e^m + 369*B*a*
b^2*d*m^2*n^2*x*x^m*x^(4*n)*e^m + 123*A*b^3*d*m^2*n^2*x*x^m*x^(4*n)*e^m + 1
22*B*b^3*c*m*n^3*x*x^m*x^(4*n)*e^m + 366*B*a*b^2*d*m*n^3*x*x^m*x^(4*n)*e^m
+ 122*A*b^3*d*m*n^3*x*x^m*x^(4*n)*e^m + 30*B*b^3*c*n^4*x*x^m*x^(4*n)*e^m +
90*B*a*b^2*d*n^4*x*x^m*x^(4*n)*e^m + 30*A*b^3*d*n^4*x*x^m*x^(4*n)*e^m + 15*
B*a*b^2*c*m^4*x*x^m*x^(3*n)*e^m + 5*A*b^3*c*m^4*x*x^m*x^(3*n)*e^m + 15*B*a^
2*b*d*m^4*x*x^m*x^(3*n)*e^m + 15*A*a*b^2*d*m^4*x*x^m*x^(3*n)*e^m + 144*B*a*
b^2*c*m^3*n*x*x^m*x^(3*n)*e^m + 48*A*b^3*c*m^3*n*x*x^m*x^(3*n)*e^m + 144*B*
a^2*b*d*m^3*n*x*x^m*x^(3*n)*e^m + 144*A*a*b^2*d*m^3*n*x*x^m*x^(3*n)*e^m + 4
41*B*a*b^2*c*m^2*n^2*x*x^m*x^(3*n)*e^m + 147*A*b^3*c*m^2*n^2*x*x^m*x^(3*n)*
e^m + 441*B*a^2*b*d*m^2*n^2*x*x^m*x^(3*n)*e^m + 441*A*a*b^2*d*m^2*n^2*x*x^m
*x^(3*n)*e^m + 468*B*a*b^2*c*m*n^3*x*x^m*x^(3*n)*e^m + 156*A*b^3*c*m*n^3*x*
x^m*x^(3*n)*e^m + 468*B*a^2*b*d*m*n^3*x*x^m*x^(3*n)*e^m + 468*A*a*b^2*d*m*n
^3*x*x^m*x^(3*n)*e^m + 120*B*a*b^2*c*n^4*x*x^m*x^(3*n)*e^m + 40*A*b^3*c*n^4
*x*x^m*x^(3*n)*e^m + 120*B*a^2*b*d*n^4*x*x^m*x^(3*n)*e^m + 120*A*a*b^2*d*n^
4*x*x^m*x^(3*n)*e^m + 15*B*a^2*b*c*m^4*x*x^m*x^(2*n)*e^m + 15*A*a*b^2*c*m^4
*x*x^m*x^(2*n)*e^m + 5*B*a^3*d*m^4*x*x^m*x^(2*n)*e^m + 15*A*a^2*b*d*m^4*x*x
^m*x^(2*n)*e^m + 156*B*a^2*b*c*m^3*n*x*x^m*x^(2*n)*e^m + 156*A*a*b^2*c*m^3*
n*x*x^m*x^(2*n)*e^m + 52*B*a^3*d*m^3*n*x*x^m*x^(2*n)*e^m + 156*A*a^2*b*d*m^
3*n*x*x^m*x^(2*n)*e^m + 531*B*a^2*b*c*m^2*n^2*x*x^m*x^(2*n)*e^m + 531*A*a*b
^2*c*m^2*n^2*x*x^m*x^(2*n)*e^m + 177*B*a^3*d*m^2*n^2*x*x^m*x^(2*n)*e^m + 53
1*A*a^2*b*d*m^2*n^2*x*x^m*x^(2*n)*e^m + 642*B*a^2*b*c*m*n^3*x*x^m*x^(2*n)*e
^m + 642*A*a*b^2*c*m*n^3*x*x^m*x^(2*n)*e^m + 214*B*a^3*d*m*n^3*x*x^m*x^(2*n
)*e^m + 642*A*a^2*b*d*m*n^3*x*x^m*x^(2*n)*e^m + 180*B*a^2*b*c*n^4*x*x^m*x^(
2*n)*e^m + 180*A*a*b^2*c*n^4*x*x^m*x^(2*n)*e^m + 60*B*a^3*d*n^4*x*x^m*x^(2*
n)*e^m + 180*A*a^2*b*d*n^4*x*x^m*x^(2*n)*e^m + 5*B*a^3*c*m^4*x*x^m*x^n*e^m
+ 15*A*a^2*b*c*m^4*x*x^m*x^n*e^m + 5*A*a^3*d*m^4*x*x^m*x^n*e^m + 56*B*a^3*c
*m^3*n*x*x^m*x^n*e^m + 168*A*a^2*b*c*m^3*n*x*x^m*x^n*e^m + 56*A*a^3*d*m^3*n
*x*x^m*x^n*e^m + 213*B*a^3*c*m^2*n^2*x*x^m*x^n*e^m + 639*A*a^2*b*c*m^2*n^2*
x*x^m*x^n*e^m + 213*A*a^3*d*m^2*n^2*x*x^m*x^n*e^m + 308*B*a^3*c*m*n^3*x*x^m
*x^n*e^m + 924*A*a^2*b*c*m*n^3*x*x^m*x^n*e^m + 308*A*a^3*d*m*n^3*x*x^m*x^n*
e^m + 120*B*a^3*c*n^4*x*x^m*x^n*e^m + 360*A*a^2*b*c*n^4*x*x^m*x^n*e^m + 120
*A*a^3*d*n^4*x*x^m*x^n*e^m + 5*A*a^3*c*m^4*x*x^m*e^m + 60*A*a^3*c*m^3*n*x*x

```
^m*e^m + 255*A*a^3*c*m^2*n^2*x*x^m*e^m + 450*A*a^3*c*m*n^3*x*x^m*e^m + 274*
A*a^3*c*n^4*x*x^m*e^m + 10*B*b^3*d*m^3*x*x^m*x^(5*n)*e^m + 60*B*b^3*d*m^2*n
*x*x^m*x^(5*n)*e^m + 105*B*b^3*d*m*n^2*x*x^m*x^(5*n)*e^m + 50*B*b^3*d*n^3*x
*x^m*x^(5*n)*e^m + 10*B*b^3*c*m^3*x*x^m*x^(4*n)*e^m + 30*B*a*b^2*d*m^3*x*x^
m*x^(4*n)*e^m + 10*A*b^3*d*m^3*x*x^m*x^(4*n)*e^m + 66*B*b^3*c*m^2*n*x*x^m*x
^(4*n)*e^m + 198*B*a*b^2*d*m^2*n*x*x^m*x^(4*n)*e^m + 66*A*b^3*d*m^2*n*x*x^m
*x^(4*n)*e^m + 123*B*b^3*c*m*n^2*x*x^m*x^(4*n)*e^m + 369*B*a*b^2*d*m*n^2*x*
x^m*x^(4*n)*e^m + 123*A*b^3*d*m*n^2*x*x^m*x^(4*n)*e^m + 61*B*b^3*c*n^3*x*x^
m*x^(4*n)*e^m + 183*B*a*b^2*d*n^3*x*x^m*x^(4*n)*e^m + 61*A*b^3*d*n^3*x*x^m*
x^(4*n)*e^m + 30*B*a*b^2*c*m^3*x*x^m*x^(3*n)*e^m + 10*A*b^3*c*m^3*x*x^m*x^(
3*n)*e^m + 30*B*a^2*b*d*m^3*x*x^m*x^(3*n)*e^m + 30*A*a*b^2*d*m^3*x*x^m*x^(3
*n)*e^m + 216*B*a*b^2*c*m^2*n*x*x^m*x^(3*n)*e^m + 72*A*b^3*c*m^2*n*x*x^m*x^
(3*n)*e^m + 216*B*a^2*b*d*m^2*n*x*x^m*x^(3*n)*e^m + 216*A*a*b^2*d*m^2*n*x*x
^m*x^(3*n)*e^m + 441*B*a*b^2*c*m*n^2*x*x^m*x^(3*n)*e^m + 147*A*b^3*c*m*n^2*
x*x^m*x^(3*n)*e^m + 441*B*a^2*b*d*m*n^2*x*x^m*x^(3*n)*e^m + 441*A*a*b^2*d*m
*n^2*x*x^m*x^(3*n)*e^m + 234*B*a*b^2*c*n^3*x*x^m*x^(3*n)*e^m + 78*A*b^3*c*n
^3*x*x^m*x^(3*n)*e^m + 234*B*a^2*b*d*n^3*x*x^m*x^(3*n)*e^m + 234*A*a*b^2*d*
n^3*x*x^m*x^(3*n)*e^m + 30*B*a^2*b*c*m^3*x*x^m*x^(2*n)*e^m + 30*A*a*b^2*c*m
^3*x*x^m*x^(2*n)*e^m + 10*B*a^3*d*m^3*x*x^m*x^(2*n)*e^m + 30*A*a^2*b*d*m^3*
x*x^m*x^(2*n)*e^m + 234*B*a^2*b*c*m^2*n*x*x^m*x^(2*n)*e^m + 234*A*a*b^2*c*m
^2*n*x*x^m*x^(2*n)*e^m + 78*B*a^3*d*m^2*n*x*x^m*x^(2*n)*e^m + 234*A*a^2*b*d
*m^2*n*x*x^m*x^(2*n)*e^m + 531*B*a^2*b*c*m*n^2*x*x^m*x^(2*n)*e^m + 531*A*a*
b^2*c*m*n^2*x*x^m*x^(2*n)*e^m + 177*B*a^3*d*m*n^2*x*x^m*x^(2*n)*e^m + 531*A
*a^2*b*d*m*n^2*x*x^m*x^(2*n)*e^m + 321*B*a^2*b*c*n^3*x*x^m*x^(2*n)*e^m + 32
1*A*a*b^2*c*n^3*x*x^m*x^(2*n)*e^m + 107*B*a^3*d*n^3*x*x^m*x^(2*n)*e^m + 321
*A*a^2*b*d*n^3*x*x^m*x^(2*n)*e^m + 10*B*a^3*c*m^3*x*x^m*x^n*e^m + 30*A*a^2*
b*c*m^3*x*x^m*x^n*e^m + 10*A*a^3*d*m^3*x*x^m*x^n*e^m + 84*B*a^3*c*m^2*n*x*x
^m*x^n*e^m + 252*A*a^2*b*c*m^2*n*x*x^m*x^n*e^m + 84*A*a^3*d*m^2*n*x*x^m*x^n
*e^m + 213*B*a^3*c*m*n^2*x*x^m*x^n*e^m + 639*A*a^2*b*c*m*n^2*x*x^m*x^n*e^m
+ 213*A*a^3*d*m*n^2*x*x^m*x^n*e^m + 154*B*a^3*c*n^3*x*x^m*x^n*e^m + 462*A*a
^2*b*c*n^3*x*x^m*x^n*e^m + 154*A*a^3*d*n^3*x*x^m*x^n*e^m + 10*A*a^3*c*m^3*x
*x^m*e^m + 90*A*a^3*c*m^2*n*x*x^m*e^m + 255*A*a^3*c*m*n^2*x*x^m*e^m + 225*A
*a^3*c*n^3*x*x^m*e^m + 10*B*b^3*d*m^2*x*x^m*x^(5*n)*e^m + 40*B*b^3*d*m*n*x*
x^m*x^(5*n)*e^m + 35*B*b^3*d*n^2*x*x^m*x^(5*n)*e^m + 10*B*b^3*c*m^2*x*x^m*x
^(4*n)*e^m + 30*B*a*b^2*d*m^2*x*x^m*x^(4*n)*e^m + 10*A*b^3*d*m^2*x*x^m*x^(4
*n)*e^m + 44*B*b^3*c*m*n*x*x^m*x^(4*n)*e^m + 132*B*a*b^2*d*m*n*x*x^m*x^(4*n
)*e^m + 44*A*b^3*d*m*n*x*x^m*x^(4*n)*e^m + 41*B*b^3*c*n^2*x*x^m*x^(4*n)*e^m
 + 123*B*a*b^2*d*n^2*x*x^m*x^(4*n)*e^m + 41*A*b^3*d*n^2*x*x^m*x^(4*n)*e^m +
 30*B*a*b^2*c*m^2*x*x^m*x^(3*n)*e^m + 10*A*b^3*c*m^2*x*x^m*x^(3*n)*e^m + 30
*B*a^2*b*d*m^2*x*x^m*x^(3*n)*e^m + 30*A*a*b^2*d*m^2*x*x^m*x^(3*n)*e^m + 144
*B*a*b^2*c*m*n*x*x^m*x^(3*n)*e^m + 48*A*b^3*c*m*n*x*x^m*x^(3*n)*e^m + 144*B
*a^2*b*d*m*n*x*x^m*x^(3*n)*e^m + 144*A*a*b^2*d*m*n*x*x^m*x^(3*n)*e^m + 147*
B*a*b^2*c*n^2*x*x^m*x^(3*n)*e^m + 49*A*b^3*c*n^2*x*x^m*x^(3*n)*e^m + 147*B*
a^2*b*d*n^2*x*x^m*x^(3*n)*e^m + 147*A*a*b^2*d*n^2*x*x^m*x^(3*n)*e^m + 30*B*
a^2*b*c*m^2*x*x^m*x^(2*n)*e^m + 30*A*a*b^2*c*m^2*x*x^m*x^(2*n)*e^m + 10*B*a
^3*d*m^2*x*x^m*x^(2*n)*e^m + 30*A*a^2*b*d*m^2*x*x^m*x^(2*n)*e^m + 156*B*a^2
*b*c*m*n*x*x^m*x^(2*n)*e^m + 156*A*a*b^2*c*m*n*x*x^m*x^(2*n)*e^m + 52*B*a^3
*d*m*n*x*x^m*x^(2*n)*e^m + 156*A*a^2*b*d*m*n*x*x^m*x^(2*n)*e^m + 177*B*a^2*
b*c*n^2*x*x^m*x^(2*n)*e^m + 177*A*a*b^2*c*n^2*x*x^m*x^(2*n)*e^m + 59*B*a^3*
d*n^2*x*x^m*x^(2*n)*e^m + 177*A*a^2*b*d*n^2*x*x^m*x^(2*n)*e^m + 10*B*a^3*c*
m^2*x*x^m*x^n*e^m + 30*A*a^2*b*c*m^2*x*x^m*x^n*e^m + 10*A*a^3*d*m^2*x*x^m*x
^n*e^m + 56*B*a^3*c*m*n*x*x^m*x^n*e^m + 168*A*a^2*b*c*m*n*x*x^m*x^n*e^m + 5
6*A*a^3*d*m*n*x*x^m*x^n*e^m + 71*B*a^3*c*n^2*x*x^m*x^n*e^m + 213*A*a^2*b*c*
n^2*x*x^m*x^n*e^m + 71*A*a^3*d*n^2*x*x^m*x^n*e^m + 10*A*a^3*c*m^2*x*x^m*e^m
 + 60*A*a^3*c*m*n*x*x^m*e^m + 85*A*a^3*c*n^2*x*x^m*e^m + 5*B*b^3*d*m*x*x^m*
x^(5*n)*e^m + 10*B*b^3*d*n*x*x^m*x^(5*n)*e^m + 5*B*b^3*c*m*x*x^m*x^(4*n)*e^
m + 15*B*a*b^2*d*m*x*x^m*x^(4*n)*e^m + 5*A*b^3*d*m*x*x^m*x^(4*n)*e^m + 11*B
*b^3*c*n*x*x^m*x^(4*n)*e^m + 33*B*a*b^2*d*n*x*x^m*x^(4*n)*e^m + 11*A*b^3*d*
n*x*x^m*x^(4*n)*e^m + 15*B*a*b^2*c*m*x*x^m*x^(3*n)*e^m + 5*A*b^3*c*m*x*x^m*
```

```
x^(3*n)*e^m + 15*B*a^2*b*d*m*x*x^m*x^(3*n)*e^m + 15*A*a*b^2*d*m*x*x^m*x^(3*
n)*e^m + 36*B*a*b^2*c*n*x*x^m*x^(3*n)*e^m + 12*A*b^3*c*n*x*x^m*x^(3*n)*e^m
+ 36*B*a^2*b*d*n*x*x^m*x^(3*n)*e^m + 36*A*a*b^2*d*n*x*x^m*x^(3*n)*e^m + 15*
B*a^2*b*c*m*x*x^m*x^(2*n)*e^m + 15*A*a*b^2*c*m*x*x^m*x^(2*n)*e^m + 5*B*a^3*
d*m*x*x^m*x^(2*n)*e^m + 15*A*a^2*b*d*m*x*x^m*x^(2*n)*e^m + 39*B*a^2*b*c*n*x
*x^m*x^(2*n)*e^m + 39*A*a*b^2*c*n*x*x^m*x^(2*n)*e^m + 13*B*a^3*d*n*x*x^m*x^
(2*n)*e^m + 39*A*a^2*b*d*n*x*x^m*x^(2*n)*e^m + 5*B*a^3*c*m*x*x^m*x^n*e^m +
15*A*a^2*b*c*m*x*x^m*x^n*e^m + 5*A*a^3*d*m*x*x^m*x^n*e^m + 14*B*a^3*c*n*x*x
^m*x^n*e^m + 42*A*a^2*b*c*n*x*x^m*x^n*e^m + 14*A*a^3*d*n*x*x^m*x^n*e^m + 5*
A*a^3*c*m*x*x^m*e^m + 15*A*a^3*c*n*x*x^m*e^m + B*b^3*d*x*x^m*x^(5*n)*e^m +
B*b^3*c*x*x^m*x^(4*n)*e^m + 3*B*a*b^2*d*x*x^m*x^(4*n)*e^m + A*b^3*d*x*x^m*x
^(4*n)*e^m + 3*B*a*b^2*c*x*x^m*x^(3*n)*e^m + A*b^3*c*x*x^m*x^(3*n)*e^m + 3*
B*a^2*b*d*x*x^m*x^(3*n)*e^m + 3*A*a*b^2*d*x*x^m*x^(3*n)*e^m + 3*B*a^2*b*c*x
*x^m*x^(2*n)*e^m + 3*A*a*b^2*c*x*x^m*x^(2*n)*e^m + B*a^3*d*x*x^m*x^(2*n)*e^
m + 3*A*a^2*b*d*x*x^m*x^(2*n)*e^m + B*a^3*c*x*x^m*x^n*e^m + 3*A*a^2*b*c*x*x
^m*x^n*e^m + A*a^3*d*x*x^m*x^n*e^m + A*a^3*c*x*x^m*e^m)/(m^6 + 15*m^5*n + 8
5*m^4*n^2 + 225*m^3*n^3 + 274*m^2*n^4 + 120*m*n^5 + 6*m^5 + 75*m^4*n + 340*
m^3*n^2 + 675*m^2*n^3 + 548*m*n^4 + 120*n^5 + 15*m^4 + 150*m^3*n + 510*m^2*
n^2 + 675*m*n^3 + 274*n^4 + 20*m^3 + 150*m^2*n + 340*m*n^2 + 225*n^3 + 15*m
^2 + 75*m*n + 85*n^2 + 6*m + 15*n + 1)
```

maple [C] time = 0.21, size = 4972, normalized size = 23.68

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(b*x^n+a)^3*(A+B*x^n)*(d*x^n+c),x)
```

```
[Out] x*(41*A*b^3*d*m^3*n^2*(x^n)^4+61*A*b^3*d*m^2*n^3*(x^n)^4+30*A*b^3*d*m*n^4*(
x^n)^4+24*B*b^3*d*m*n^4*(x^n)^5+11*A*b^3*d*m^4*n*(x^n)^4+40*B*b^3*d*m^3*n*(
x^n)^5+3*B*a*b^2*d*m^5*(x^n)^4+11*B*b^3*c*m^4*n*(x^n)^4+41*B*b^3*c*m^3*n^2*
(x^n)^4+61*B*b^3*c*m^2*n^3*(x^n)^4+10*B*b^3*d*m^4*n*(x^n)^5+35*B*b^3*d*m^3*
n^2*(x^n)^5+50*B*b^3*d*m^2*n^3*(x^n)^5+105*B*b^3*d*m^2*n^2*(x^n)^5+100*B*b^
3*d*m*n^3*(x^n)^5+3*A*a*b^2*d*m^5*(x^n)^3+12*A*b^3*c*m^4*n*(x^n)^3+49*A*b^3
*c*m^3*n^2*(x^n)^3+78*A*b^3*c*m^2*n^3*(x^n)^3+40*A*b^3*c*m*n^4*(x^n)^3+44*A
*b^3*d*m^3*n*(x^n)^4+123*A*b^3*d*m^2*n^2*(x^n)^4+30*B*b^3*c*m*n^4*(x^n)^4+3
*B*a*b^2*c*m^5*(x^n)^3+15*B*a*b^2*d*m^4*(x^n)^4+90*B*a*b^2*d*n^4*(x^n)^4+44
*B*b^3*c*m^3*n*(x^n)^4+123*B*b^3*c*m^2*n^2*(x^n)^4+122*B*b^3*c*m*n^3*(x^n)^
4+60*B*b^3*d*m^2*n*(x^n)^5+105*B*b^3*d*m*n^2*(x^n)^5+3*A*a^2*b*d*m^5*(x^n)^
2+3*A*a*b^2*c*m^5*(x^n)^2+66*A*b^3*d*m^2*n*(x^n)^4+123*A*b^3*d*m*n^2*(x^n)^
4+122*A*b^3*d*m*n^3*(x^n)^4+3*B*a^2*b*d*m^5*(x^n)^3+15*A*a*b^2*d*m^4*(x^n)^
3+120*A*a*b^2*d*n^4*(x^n)^3+48*A*b^3*c*m^3*n*(x^n)^3+147*A*b^3*c*m^2*n^2*(x
^n)^3+156*A*b^3*c*m*n^3*(x^n)^3+15*B*a*b^2*c*m^4*(x^n)^3+120*B*a*b^2*c*n^4*
(x^n)^3+30*B*a*b^2*d*m^3*(x^n)^4+183*B*a*b^2*d*n^3*(x^n)^4+66*B*b^3*c*m^2*n
*(x^n)^4+123*B*b^3*c*m*n^2*(x^n)^4+40*B*b^3*d*m*n*(x^n)^5+14*A*a^3*d*m^4*n*
x^n+71*A*a^3*d*m^3*n^2*x^n+120*B*a^3*c*m*n^4*x^n+52*B*a^3*d*m^3*n*(x^n)^2+1
3*B*a^3*d*m^4*n*(x^n)^2+59*B*a^3*d*m^3*n^2*(x^n)^2+107*B*a^3*d*m^2*n^3*(x^n
)^2+60*B*a^3*d*m*n^4*(x^n)^2+3*B*a^2*b*c*m^5*(x^n)^2+15*B*a^2*b*d*m^4*(x^n)
^3+120*B*a^2*b*d*n^4*(x^n)^3+180*A*a*b^2*c*n^4*(x^n)^2+30*A*a*b^2*d*m^3*(x^
n)^3+234*A*a*b^2*d*n^3*(x^n)^3+72*A*b^3*c*m^2*n*(x^n)^3+147*A*b^3*c*m*n^2*(
x^n)^3+44*A*b^3*d*m*n*(x^n)^4+14*B*a^3*c*m^4*n*x^n+71*B*a^3*c*m^3*n^2*x^n+1
54*B*a^3*c*m^2*n^3*x^n+321*A*a^2*b*d*n^3*(x^n)^2+30*A*a*b^2*c*m^3*(x^n)^2+3
21*A*a*b^2*c*n^3*(x^n)^2+154*A*a^3*d*m^2*n^3*x^n+120*A*a^3*d*m*n^4*x^n+3*A*
a^2*b*c*m^5*x^n+15*A*a^2*b*d*m^4*(x^n)^2+180*A*a^2*b*d*n^4*(x^n)^2+15*A*a*b
^2*c*m^4*(x^n)^2+234*B*a^2*b*d*n^3*(x^n)^3+30*B*a*b^2*c*m^3*(x^n)^3+234*B*a
*b^2*c*n^3*(x^n)^3+30*B*a*b^2*d*m^2*(x^n)^4+123*B*a*b^2*d*n^2*(x^n)^4+44*B*
b^3*c*m*n*(x^n)^4+56*A*a^3*d*m^3*n*x^n+213*A*a^3*d*m^2*n^2*x^n+308*A*a^3*d*
m*n^3*x^n+15*A*a^2*b*c*m^4*x^n+360*A*a^2*b*c*n^4*x^n+30*A*a^2*b*d*m^3*(x^n)
^2+84*A*a^3*d*m^2*n*x^n+213*A*a^3*d*m*n^2*x^n+30*A*a^2*b*c*m^3*x^n+462*A*a^
2*b*c*n^3*x^n+177*B*a^3*d*m^2*n^2*(x^n)^2+214*B*a^3*d*m*n^3*(x^n)^2+15*B*a^
```

```
2*b*c*m^4*(x^n)^2+180*B*a^2*b*c*n^4*(x^n)^2+30*B*a^2*b*d*m^3*(x^n)^3+177*B*
a^3*d*m*n^2*(x^n)^2+30*B*a^2*b*c*m^3*(x^n)^2+321*B*a^2*b*c*n^3*(x^n)^2+30*B
*a^2*b*d*m^2*(x^n)^3+147*B*a^2*b*d*n^2*(x^n)^3+30*B*a*b^2*c*m^2*(x^n)^3+147
*B*a*b^2*c*n^2*(x^n)^3+15*B*a*b^2*d*(x^n)^4*m+33*B*a*b^2*d*(x^n)^4*n+30*A*a
*b^2*d*m^2*(x^n)^3+147*A*a*b^2*d*n^2*(x^n)^3+48*A*b^3*c*m*n*(x^n)^3+56*B*a^
3*c*m^3*n*x^n+213*B*a^3*c*m^2*n^2*x^n+308*B*a^3*c*m*n^3*x^n+78*B*a^3*d*m^2*
n*(x^n)^2+213*B*a^3*c*m*n^2*x^n+52*B*a^3*d*m*n*(x^n)^2+30*B*a^2*b*c*m^2*(x^
n)^2+177*B*a^2*b*c*n^2*(x^n)^2+15*B*a^2*b*d*(x^n)^3*m+36*B*a^2*b*d*(x^n)^3*
n+15*B*a*b^2*c*(x^n)^3*m+30*A*a^2*b*d*m^2*(x^n)^2+177*A*a^2*b*d*n^2*(x^n)^2
+30*A*a*b^2*c*m^2*(x^n)^2+177*A*a*b^2*c*n^2*(x^n)^2+15*A*a*b^2*d*(x^n)^3*m+
36*A*a*b^2*d*(x^n)^3*n+84*B*a^3*c*m^2*n*x^n+30*A*a^2*b*c*m^2*x^n+213*A*a^2*
b*c*n^2*x^n+15*A*a^2*b*d*(x^n)^2*m+39*A*a^2*b*d*(x^n)^2*n+15*A*a*b^2*c*(x^n
)^2*m+39*A*a*b^2*c*(x^n)^2*n+42*A*a^2*b*c*x^n*n+36*B*a*b^2*c*(x^n)^3*n+56*A
*a^3*d*m*n*x^n+56*B*a^3*c*m*n*x^n+15*B*a^2*b*c*(x^n)^2*m+39*B*a^2*b*c*(x^n)
^2*n+15*A*a^2*b*c*x^n*m+B*b^3*c*(x^n)^4+A*b^3*c*(x^n)^3+B*a^3*d*(x^n)^2+A*a
^3*d*x^n+B*a^3*c*x^n+b^3*B*d*(x^n)^5+A*b^3*d*(x^n)^4+10*A*a^3*c*m^3+225*A*a
^3*c*n^3+10*A*a^3*c*m^2+85*A*a^3*c*n^2+A*a^3*c*m^5+5*A*a^3*c*m^4+274*A*a^3*
c*n^4+120*A*a^3*c*n^5+5*a^3*A*c*m+15*a^3*A*c*n+90*B*a*b^2*d*m*n^4*(x^n)^4+3
6*A*a*b^2*d*m^4*n*(x^n)^3+147*A*a*b^2*d*m^3*n^2*(x^n)^3+234*A*a*b^2*d*m^2*n
^3*(x^n)^3+a^3*A*c+468*B*a^2*b*d*m*n^3*(x^n)^3+144*A*a*b^2*d*m*n*(x^n)^3+23
4*B*a^2*b*c*m^2*n*(x^n)^2+531*B*a^2*b*c*m*n^2*(x^n)^2+144*B*a^2*b*d*m*n*(x^
n)^3+144*B*a*b^2*c*m*n*(x^n)^3+3*(x^n)^3*B*a^2*b*d+3*(x^n)^4*B*a*b^2*d+156*
A*a^2*b*d*m^3*n*(x^n)^2+531*A*a^2*b*d*m^2*n^2*(x^n)^2+642*A*a^2*b*d*m*n^3*(
x^n)^2+156*A*a*b^2*c*m^3*n*(x^n)^2+531*A*a*b^2*c*m^2*n^2*(x^n)^2+642*A*a*b^
2*c*m*n^3*(x^n)^2+216*A*a*b^2*d*m^2*n*(x^n)^3+441*A*a*b^2*d*m*n^2*(x^n)^3+2
52*A*a^2*b*c*m^2*n*x^n+639*A*a^2*b*c*m*n^2*x^n+156*A*a^2*b*d*m*n*(x^n)^2+15
6*A*a*b^2*c*m*n*(x^n)^2+156*B*a^2*b*c*m^3*n*(x^n)^2+531*B*a^2*b*c*m^2*n^2*(
x^n)^2+642*B*a^2*b*c*m*n^3*(x^n)^2+216*B*a^2*b*d*m^2*n*(x^n)^3+369*B*a*b^2*
d*m^2*n^2*(x^n)^4+366*B*a*b^2*d*m*n^3*(x^n)^4+39*A*a^2*b*d*m^4*n*(x^n)^2+17
7*A*a^2*b*d*m^3*n^2*(x^n)^2+321*A*a^2*b*d*m^2*n^3*(x^n)^2+321*B*a^2*b*c*m^2
*n^3*(x^n)^2+180*B*a^2*b*c*m*n^4*(x^n)^2+144*B*a^2*b*d*m^3*n*(x^n)^3+441*B*
a^2*b*d*m^2*n^2*(x^n)^3+33*B*a*b^2*d*m^4*n*(x^n)^4+123*B*a*b^2*d*m^3*n^2*(x
^n)^4+183*B*a*b^2*d*m^2*n^3*(x^n)^4+42*A*a^2*b*c*m^4*n*x^n+213*A*a^2*b*c*m^
3*n^2*x^n+462*A*a^2*b*c*m^2*n^3*x^n+360*A*a^2*b*c*m*n^4*x^n+639*A*a^2*b*c*m
^2*n^2*x^n+924*A*a^2*b*c*m*n^3*x^n+234*A*a^2*b*d*m^2*n*(x^n)^2+531*A*a^2*b*
d*m*n^2*(x^n)^2+234*A*a*b^2*c*m^2*n*(x^n)^2+531*A*a*b^2*c*m*n^2*(x^n)^2+120
*A*a*b^2*d*m*n^4*(x^n)^3+36*B*a^2*b*d*m^4*n*(x^n)^3+147*B*a^2*b*d*m^3*n^2*(
x^n)^3+180*A*a^2*b*d*m*n^4*(x^n)^2+39*A*a*b^2*c*m^4*n*(x^n)^2+177*A*a*b^2*c
*m^3*n^2*(x^n)^2+321*A*a*b^2*c*m^2*n^3*(x^n)^2+180*A*a*b^2*c*m*n^4*(x^n)^2+
144*A*a*b^2*d*m^3*n*(x^n)^3+441*A*a*b^2*d*m^2*n^2*(x^n)^3+468*A*a*b^2*d*m*n
^3*(x^n)^3+39*B*a^2*b*c*m^4*n*(x^n)^2+177*B*a^2*b*c*m^3*n^2*(x^n)^2+156*B*a
^2*b*c*m*n*(x^n)^2+168*A*a^2*b*c*m*n*x^n+234*B*a^2*b*d*m^2*n^3*(x^n)^3+120*
B*a^2*b*d*m*n^4*(x^n)^3+36*B*a*b^2*c*m^4*n*(x^n)^3+147*B*a*b^2*c*m^3*n^2*(x
^n)^3+234*B*a*b^2*c*m^2*n^3*(x^n)^3+120*B*a*b^2*c*m*n^4*(x^n)^3+132*B*a*b^2
*d*m^3*n*(x^n)^4+144*B*a*b^2*c*m^3*n*(x^n)^3+441*B*a*b^2*c*m^2*n^2*(x^n)^3+
468*B*a*b^2*c*m*n^3*(x^n)^3+198*B*a*b^2*d*m^2*n*(x^n)^4+369*B*a*b^2*d*m*n^2
*(x^n)^4+441*B*a^2*b*d*m*n^2*(x^n)^3+216*B*a*b^2*c*m^2*n*(x^n)^3+441*B*a*b^
2*c*m*n^2*(x^n)^3+132*B*a*b^2*d*m*n*(x^n)^4+168*A*a^2*b*c*m^3*n*x^n+107*B*a
^3*d*n^3*(x^n)^2+5*B*b^3*c*(x^n)^4*m+3*(x^n)^2*d*a^2*b*A+3*(x^n)^2*c*a*b^2*
A+3*(x^n)^3*A*a*b^2*d+10*B*b^3*d*m^3*(x^n)^5+50*B*b^3*d*n^3*(x^n)^5+5*A*b^3
*c*m^4*(x^n)^3+40*A*b^3*c*n^4*(x^n)^3+10*A*b^3*d*m^3*(x^n)^4+61*A*b^3*d*n^3
*(x^n)^4+B*a^3*d*m^5*(x^n)^2+10*B*b^3*c*m^3*(x^n)^4+61*B*b^3*c*n^3*(x^n)^4+
10*B*b^3*d*m^2*(x^n)^5+35*B*b^3*d*n^2*(x^n)^5+A*a^3*d*m^5*x^n+5*B*a^3*c*m^4
*x^n+120*B*a^3*c*n^4*x^n+10*A*a^3*d*m^3*x^n+3*(x^n)^2*c*a^2*b*B+3*x^n*c*a^2
*b*A+3*(x^n)^3*B*a*b^2*c+10*B*a^3*c*m^3*x^n+154*B*a^3*c*n^3*x^n+10*B*a^3*d*
m^2*(x^n)^2+59*B*a^3*d*n^2*(x^n)^2+10*A*a^3*d*m^2*x^n+71*A*a^3*d*n^2*x^n+10
*B*a^3*c*m^2*x^n+71*B*a^3*c*n^2*x^n+5*B*a^3*d*(x^n)^2*m+13*B*a^3*d*(x^n)^2*
n+B*b^3*d*m^5*(x^n)^5+11*B*b^3*c*(x^n)^4*n+154*A*a^3*d*n^3*x^n+450*A*a^3*c*
m*n^3+90*A*a^3*c*m^2*n+255*A*a^3*c*m*n^2+60*A*a^3*c*m*n+274*A*a^3*c*m*n^4+6
```

```
0*A*a^3*c*m^3*n+255*A*a^3*c*m^2*n^2+15*A*a^3*c*m^4*n+85*A*a^3*c*m^3*n^2+225
*A*a^3*c*m^2*n^3+A*b^3*d*m^5*(x^n)^4+B*b^3*c*m^5*(x^n)^4+5*B*b^3*d*m^4*(x^n
)^5+24*B*b^3*d*n^4*(x^n)^5+A*b^3*c*m^5*(x^n)^3+5*A*b^3*d*m^4*(x^n)^4+30*A*b
^3*d*n^4*(x^n)^4+5*B*b^3*c*m^4*(x^n)^4+30*B*b^3*c*n^4*(x^n)^4+5*B*a^3*c*x^n
*m+14*B*a^3*c*x^n*n+10*A*b^3*c*m^3*(x^n)^3+78*A*b^3*c*n^3*(x^n)^3+10*A*b^3*
d*m^2*(x^n)^4+41*A*b^3*d*n^2*(x^n)^4+B*a^3*c*m^5*x^n+5*B*a^3*d*m^4*(x^n)^2+
60*B*a^3*d*n^4*(x^n)^2+10*B*b^3*c*m^2*(x^n)^4+120*A*a^3*d*n^4*x^n+10*A*b^3*
c*m^2*(x^n)^3+49*A*b^3*c*n^2*(x^n)^3+5*A*b^3*d*(x^n)^4*m+11*A*b^3*d*(x^n)^4
*n+5*A*b^3*c*(x^n)^3*m+12*A*b^3*c*(x^n)^3*n+41*B*b^3*c*n^2*(x^n)^4+5*m*b^3*
B*d*(x^n)^5+10*b^3*B*d*(x^n)^5*n+5*A*a^3*d*m^4*x^n+10*B*a^3*d*m^3*(x^n)^2+5
*A*a^3*d*x^n*m+14*A*a^3*d*x^n*n)/(m+1)/(m+n+1)/(m+2*n+1)/(m+3*n+1)/(1+m+4*n
)/(1+m+5*n)*exp(1/2*m*(-I*Pi*csgn(I*e*x)^3+I*Pi*csgn(I*e*x)^2*csgn(I*e)+I*P
i*csgn(I*e*x)^2*csgn(I*x)-I*Pi*csgn(I*e*x)*csgn(I*e)*csgn(I*x)+2*ln(e)+2*ln
(x)))
```

maxima [B] time = 0.94, size = 464, normalized size = 2.21

$$\frac{Bb^3de^mxe^{\left(m\log(x)+5n\log(x)\right)}}{m+5n+1}+\frac{Bb^3ce^mxe^{\left(m\log(x)+4n\log(x)\right)}}{m+4n+1}+\frac{3Bab^2de^mxe^{\left(m\log(x)+4n\log(x)\right)}}{m+4n+1}+\frac{Ab^3de^mxe^{\left(m\log(x)+4n\log(x)\right)}}{m+4n+1}+\frac{3Bab^2ce^mxe^{\left(m\log(x)+3n\log(x)\right)}}{m+3n+1}+\frac{Ab^3ce^mxe^{\left(m\log(x)+3n\log(x)\right)}}{m+3n+1}+\frac{3Ba^2bde^mxe^{\left(m\log(x)+3n\log(x)\right)}}{m+3n+1}+\frac{3Aab^2de^mxe^{\left(m\log(x)+3n\log(x)\right)}}{m+3n+1}+\frac{3Ba^2bce^mxe^{\left(m\log(x)+2n\log(x)\right)}}{m+2n+1}+\frac{3Aab^2ce^mxe^{\left(m\log(x)+2n\log(x)\right)}}{m+2n+1}+\frac{Ba^3de^mxe^{\left(m\log(x)+2n\log(x)\right)}}{m+2n+1}+\frac{3Aa^2bde^mxe^{\left(m\log(x)+2n\log(x)\right)}}{m+2n+1}+\frac{Ba^3ce^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{3Aa^2bce^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{Aa^3de^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{(ex)^{m+1}Aa^3c}{e(m+1)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)^3*(A+B*x^n)*(c+d*x^n),x, algorithm="maxima")
```

```
[Out] B*b^3*d*e^m*x*e^(m*log(x) + 5*n*log(x))/(m + 5*n + 1) + B*b^3*c*e^m*x*e^(m*
log(x) + 4*n*log(x))/(m + 4*n + 1) + 3*B*a*b^2*d*e^m*x*e^(m*log(x) + 4*n*lo
g(x))/(m + 4*n + 1) + A*b^3*d*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1)
 + 3*B*a*b^2*c*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + A*b^3*c*e^m*
x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + 3*B*a^2*b*d*e^m*x*e^(m*log(x) +
 3*n*log(x))/(m + 3*n + 1) + 3*A*a*b^2*d*e^m*x*e^(m*log(x) + 3*n*log(x))/(m
 + 3*n + 1) + 3*B*a^2*b*c*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 3
*A*a*b^2*c*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + B*a^3*d*e^m*x*e^
(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 3*A*a^2*b*d*e^m*x*e^(m*log(x) + 2*n
*log(x))/(m + 2*n + 1) + B*a^3*c*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1)
+ 3*A*a^2*b*c*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1) + A*a^3*d*e^m*x*e^(
m*log(x) + n*log(x))/(m + n + 1) + (e*x)^(m + 1)*A*a^3*c/(e*(m + 1))
```

mupad [B] time = 5.64, size = 1089, normalized size = 5.19

[illegible]

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(A + B*x^n)*(a + b*x^n)^3*(c + d*x^n),x)
```

```
[Out] (A*a^3*c*x*(e*x)^m)/(m + 1) + (b^2*x*x^(4*n)*(e*x)^m*(A*b*d + 3*B*a*d + B*b
*c)*(4*m + 11*n + 33*m*n + 82*m*n^2 + 33*m^2*n + 61*m*n^3 + 11*m^3*n + 6*m^
2 + 4*m^3 + m^4 + 41*n^2 + 61*n^3 + 30*n^4 + 41*m^2*n^2 + 1))/(5*m + 15*n +
 60*m*n + 255*m*n^2 + 90*m^2*n + 450*m*n^3 + 60*m^3*n + 274*m*n^4 + 15*m^4*
n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 85*n^2 + 225*n^3 + 274*n^4 + 120*n^5 +
255*m^2*n^2 + 225*m^2*n^3 + 85*m^3*n^2 + 1) + (a*x*x^(2*n)*(e*x)^m*(3*A*b^2
*c + B*a^2*d + 3*A*a*b*d + 3*B*a*b*c)*(4*m + 13*n + 39*m*n + 118*m*n^2 + 39
*m^2*n + 107*m*n^3 + 13*m^3*n + 6*m^2 + 4*m^3 + m^4 + 59*n^2 + 107*n^3 + 60
*n^4 + 59*m^2*n^2 + 1))/(5*m + 15*n + 60*m*n + 255*m*n^2 + 90*m^2*n + 450*m
*n^3 + 60*m^3*n + 274*m*n^4 + 15*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 85
*n^2 + 225*n^3 + 274*n^4 + 120*n^5 + 255*m^2*n^2 + 225*m^2*n^3 + 85*m^3*n^2
 + 1) + (b*x*x^(3*n)*(e*x)^m*(A*b^2*c + 3*B*a^2*d + 3*A*a*b*d + 3*B*a*b*c)*
(4*m + 12*n + 36*m*n + 98*m*n^2 + 36*m^2*n + 78*m*n^3 + 12*m^3*n + 6*m^2 +
4*m^3 + m^4 + 49*n^2 + 78*n^3 + 40*n^4 + 49*m^2*n^2 + 1))/(5*m + 15*n + 60*
m*n + 255*m*n^2 + 90*m^2*n + 450*m*n^3 + 60*m^3*n + 274*m*n^4 + 15*m^4*n +
10*m^2 + 10*m^3 + 5*m^4 + m^5 + 85*n^2 + 225*n^3 + 274*n^4 + 120*n^5 + 255*
m^2*n^2 + 225*m^2*n^3 + 85*m^3*n^2 + 1) + (a^2*x*x^n*(e*x)^m*(A*a*d + 3*A*b
```

```
*c + B*a*c)*(4*m + 14*n + 42*m*n + 142*m*n^2 + 42*m^2*n + 154*m*n^3 + 14*m^
3*n + 6*m^2 + 4*m^3 + m^4 + 71*n^2 + 154*n^3 + 120*n^4 + 71*m^2*n^2 + 1))/(
5*m + 15*n + 60*m*n + 255*m*n^2 + 90*m^2*n + 450*m*n^3 + 60*m^3*n + 274*m*n
^4 + 15*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 85*n^2 + 225*n^3 + 274*n^4
+ 120*n^5 + 255*m^2*n^2 + 225*m^2*n^3 + 85*m^3*n^2 + 1) + (B*b^3*d*x*x^(5*n
)*(e*x)^m*(4*m + 10*n + 30*m*n + 70*m*n^2 + 30*m^2*n + 50*m*n^3 + 10*m^3*n
+ 6*m^2 + 4*m^3 + m^4 + 35*n^2 + 50*n^3 + 24*n^4 + 35*m^2*n^2 + 1))/(5*m +
15*n + 60*m*n + 255*m*n^2 + 90*m^2*n + 450*m*n^3 + 60*m^3*n + 274*m*n^4 + 1
5*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 85*n^2 + 225*n^3 + 274*n^4 + 120*
n^5 + 255*m^2*n^2 + 225*m^2*n^3 + 85*m^3*n^2 + 1)
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)**m*(a+b*x**n)**3*(A+B*x**n)*(c+d*x**n),x)
```

```
[Out] Timed out
```

3.2 $\int (ex)^m (a+bx^n)^2 (A+Bx^n)(c+dx^n)\,dx$

Optimal. Leaf size=160

$$\frac{a^2Ac(ex)^{m+1}}{e(m+1)}+\frac{ax^{n+1}(ex)^m(aAd+aBc+2Abc)}{m+n+1}+\frac{x^{2n+1}(ex)^m(Ab(2ad+bc)+aB(ad+2bc))}{m+2n+1}+\frac{bx^{3n+1}(ex)^m(2aB\ldots}{m+3\ldots}$$

Rubi [A] time = 0.18, antiderivative size = 160, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 3, integrand size = 29, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.103, Rules used = {570, 20, 30}

$$\frac{a^2Ac(ex)^{m+1}}{e(m+1)}+\frac{ax^{n+1}(ex)^m(aAd+aBc+2Abc)}{m+n+1}+\frac{x^{2n+1}(ex)^m(Ab(2ad+bc)+aB(ad+2bc))}{m+2n+1}+\frac{bx^{3n+1}(ex)^m(2aBd+Abd+bBc)}{m+3n+1}+\frac{b^2Bdx^{4n+1}(ex)^m}{m+4n+1}$$

Antiderivative was successfully verified.

```
[In] Int[(e*x)^m*(a + b*x^n)^2*(A + B*x^n)*(c + d*x^n),x]
```

```
[Out] (a*(2*A*b*c + a*B*c + a*A*d)*x^(1 + n)*(e*x)^m)/(1 + m + n) + ((a*B*(2*b*c
+ a*d) + A*b*(b*c + 2*a*d))*x^(1 + 2*n)*(e*x)^m)/(1 + m + 2*n) + (b*(b*B*c
+ A*b*d + 2*a*B*d)*x^(1 + 3*n)*(e*x)^m)/(1 + m + 3*n) + (b^2*B*d*x^(1 + 4*n
)*(e*x)^m)/(1 + m + 4*n) + (a^2*A*c*(e*x)^(1 + m))/(e*(1 + m))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 30

```
Int[(x_)^(m_.), x_Symbol] :> Simp[x^(m + 1)/(m + 1), x] /; FreeQ[m, x] && N
eQ[m, -1]
```

Rule 570

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

Rubi steps

$$\begin{aligned}\int (ex)^m (a+bx^n)^2 (A+Bx^n)(c+dx^n)\,dx &= \int \left(a^2Ac(ex)^m+a(2Abc+aBc+aAd)x^n(ex)^m+(aB(2bc+ad)+\ldots\right.\\ &= \frac{a^2Ac(ex)^{1+m}}{e(1+m)}+\left(b^2Bd\right)\int x^{4n}(ex)^m\,dx+(a(2Abc+aBc+aAd))\ldots\\ &= \frac{a^2Ac(ex)^{1+m}}{e(1+m)}+\left(b^2Bdx^{-m}(ex)^m\right)\int x^{m+4n}\,dx+(a(2Abc+aBc+\ldots\\ &= \frac{a(2Abc+aBc+aAd)x^{1+n}(ex)^m}{1+m+n}+\frac{(aB(2bc+ad)+Ab(bc+2ad)\ldots}{1+m+2n}\end{aligned}$$

Mathematica [A] time = 0.53, size = 129, normalized size = 0.81

$$x(ex)^m\left(\frac{a^2Ac}{m+1}+\frac{x^{2n}(Ab(2ad+bc)+aB(ad+2bc))}{m+2n+1}+\frac{bx^{3n}(2aBd+Abd+bBc)}{m+3n+1}+\frac{ax^n(aAd+aBc+2Abc)}{m+n+1}+\frac{b^2Bdx^{4n}}{m+4n+1}\right)$$

Antiderivative was successfully verified.

```
[In] Integrate[(e*x)^m*(a + b*x^n)^2*(A + B*x^n)*(c + d*x^n),x]
```

```
[Out] x*(e*x)^m*((a^2*A*c)/(1 + m) + (a*(2*A*b*c + a*B*c + a*A*d)*x^n)/(1 + m + n
) + ((a*B*(2*b*c + a*d) + A*b*(b*c + 2*a*d))*x^(2*n))/(1 + m + 2*n) + (b*(b
*B*c + A*b*d + 2*a*B*d)*x^(3*n))/(1 + m + 3*n) + (b^2*B*d*x^(4*n))/(1 + m +
 4*n))
```

IntegrateAlgebraic [F] time = 0.17, size = 0, normalized size = 0.00

$$\int (ex)^m \left(a+bx^n\right)^2 \left(A+Bx^n\right) \left(c+dx^n\right) dx$$

Verification is not applicable to the result.

```
[In] IntegrateAlgebraic[(e*x)^m*(a + b*x^n)^2*(A + B*x^n)*(c + d*x^n),x]
```

```
[Out] Defer[IntegrateAlgebraic][(e*x)^m*(a + b*x^n)^2*(A + B*x^n)*(c + d*x^n), x]
```

fricas [B] time = 0.49, size = 1524, normalized size = 9.52

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)^2*(A+B*x^n)*(c+d*x^n),x, algorithm="fricas")
```

```
[Out] ((B*b^2*d*m^4 + 4*B*b^2*d*m^3 + 6*B*b^2*d*m^2 + 4*B*b^2*d*m + B*b^2*d + 6*(
B*b^2*d*m + B*b^2*d)*n^3 + 11*(B*b^2*d*m^2 + 2*B*b^2*d*m + B*b^2*d)*n^2 + 6
*(B*b^2*d*m^3 + 3*B*b^2*d*m^2 + 3*B*b^2*d*m + B*b^2*d)*n)*x*x^(4*n)*e^(m*lo
g(e) + m*log(x)) + ((B*b^2*c + (2*B*a*b + A*b^2)*d)*m^4 + B*b^2*c + 4*(B*b^
2*c + (2*B*a*b + A*b^2)*d)*m^3 + 8*(B*b^2*c + (2*B*a*b + A*b^2)*d + (B*b^2*
c + (2*B*a*b + A*b^2)*d)*m)*n^3 + 6*(B*b^2*c + (2*B*a*b + A*b^2)*d)*m^2 + 1
4*(B*b^2*c + (B*b^2*c + (2*B*a*b + A*b^2)*d)*m^2 + (2*B*a*b + A*b^2)*d + 2*
(B*b^2*c + (2*B*a*b + A*b^2)*d)*m)*n^2 + (2*B*a*b + A*b^2)*d + 4*(B*b^2*c +
 (2*B*a*b + A*b^2)*d)*m + 7*(B*b^2*c + (B*b^2*c + (2*B*a*b + A*b^2)*d)*m^3
+ 3*(B*b^2*c + (2*B*a*b + A*b^2)*d)*m^2 + (2*B*a*b + A*b^2)*d + 3*(B*b^2*c
+ (2*B*a*b + A*b^2)*d)*m)*n)*x*x^(3*n)*e^(m*log(e) + m*log(x)) + (((2*B*a*b
 + A*b^2)*c + (B*a^2 + 2*A*a*b)*d)*m^4 + 4*((2*B*a*b + A*b^2)*c + (B*a^2 +
2*A*a*b)*d)*m^3 + 12*((2*B*a*b + A*b^2)*c + (B*a^2 + 2*A*a*b)*d + ((2*B*a*b
 + A*b^2)*c + (B*a^2 + 2*A*a*b)*d)*m)*n^3 + 6*((2*B*a*b + A*b^2)*c + (B*a^2
 + 2*A*a*b)*d)*m^2 + 19*(((2*B*a*b + A*b^2)*c + (B*a^2 + 2*A*a*b)*d)*m^2 +
(2*B*a*b + A*b^2)*c + (B*a^2 + 2*A*a*b)*d + 2*((2*B*a*b + A*b^2)*c + (B*a^2
 + 2*A*a*b)*d)*m)*n^2 + (2*B*a*b + A*b^2)*c + (B*a^2 + 2*A*a*b)*d + 4*((2*B
*a*b + A*b^2)*c + (B*a^2 + 2*A*a*b)*d)*m + 8*(((2*B*a*b + A*b^2)*c + (B*a^2
 + 2*A*a*b)*d)*m^3 + 3*((2*B*a*b + A*b^2)*c + (B*a^2 + 2*A*a*b)*d)*m^2 + (2
*B*a*b + A*b^2)*c + (B*a^2 + 2*A*a*b)*d + 3*((2*B*a*b + A*b^2)*c + (B*a^2 +
 2*A*a*b)*d)*m)*n)*x*x^(2*n)*e^(m*log(e) + m*log(x)) + ((A*a^2*d + (B*a^2 +
 2*A*a*b)*c)*m^4 + A*a^2*d + 4*(A*a^2*d + (B*a^2 + 2*A*a*b)*c)*m^3 + 24*(A*
a^2*d + (B*a^2 + 2*A*a*b)*c + (A*a^2*d + (B*a^2 + 2*A*a*b)*c)*m)*n^3 + 6*(A
*a^2*d + (B*a^2 + 2*A*a*b)*c)*m^2 + 26*(A*a^2*d + (A*a^2*d + (B*a^2 + 2*A*a
*b)*c)*m^2 + (B*a^2 + 2*A*a*b)*c + 2*(A*a^2*d + (B*a^2 + 2*A*a*b)*c)*m)*n^2
 + (B*a^2 + 2*A*a*b)*c + 4*(A*a^2*d + (B*a^2 + 2*A*a*b)*c)*m + 9*(A*a^2*d +
 (A*a^2*d + (B*a^2 + 2*A*a*b)*c)*m^3 + 3*(A*a^2*d + (B*a^2 + 2*A*a*b)*c)*m^
2 + (B*a^2 + 2*A*a*b)*c + 3*(A*a^2*d + (B*a^2 + 2*A*a*b)*c)*m)*n)*x*x^n*e^(
m*log(e) + m*log(x)) + (A*a^2*c*m^4 + 24*A*a^2*c*n^4 + 4*A*a^2*c*m^3 + 6*A*
a^2*c*m^2 + 4*A*a^2*c*m + A*a^2*c + 50*(A*a^2*c*m + A*a^2*c)*n^3 + 35*(A*a^
2*c*m^2 + 2*A*a^2*c*m + A*a^2*c)*n^2 + 10*(A*a^2*c*m^3 + 3*A*a^2*c*m^2 + 3*
A*a^2*c*m + A*a^2*c)*n)*x*e^(m*log(e) + m*log(x)))/(m^5 + 24*(m + 1)*n^4 +
5*m^4 + 50*(m^2 + 2*m + 1)*n^3 + 10*m^3 + 35*(m^3 + 3*m^2 + 3*m + 1)*n^2 +
10*m^2 + 10*(m^4 + 4*m^3 + 6*m^2 + 4*m + 1)*n + 5*m + 1)
```

giac [B] time = 0.81, size = 3415, normalized size = 21.34

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)^2*(A+B*x^n)*(c+d*x^n),x, algorithm="giac")
```

```
[Out] (B*b^2*d*m^4*x*x^m*x^(4*n)*e^m + 6*B*b^2*d*m^3*n*x*x^m*x^(4*n)*e^m + 11*B*b
^2*d*m^2*n^2*x*x^m*x^(4*n)*e^m + 6*B*b^2*d*m*n^3*x*x^m*x^(4*n)*e^m + B*b^2*
c*m^4*x*x^m*x^(3*n)*e^m + 2*B*a*b*d*m^4*x*x^m*x^(3*n)*e^m + A*b^2*d*m^4*x*x
^m*x^(3*n)*e^m + 7*B*b^2*c*m^3*n*x*x^m*x^(3*n)*e^m + 14*B*a*b*d*m^3*n*x*x^m
*x^(3*n)*e^m + 7*A*b^2*d*m^3*n*x*x^m*x^(3*n)*e^m + 14*B*b^2*c*m^2*n^2*x*x^m
*x^(3*n)*e^m + 28*B*a*b*d*m^2*n^2*x*x^m*x^(3*n)*e^m + 14*A*b^2*d*m^2*n^2*x*
x^m*x^(3*n)*e^m + 8*B*b^2*c*m*n^3*x*x^m*x^(3*n)*e^m + 16*B*a*b*d*m*n^3*x*x^
m*x^(3*n)*e^m + 8*A*b^2*d*m*n^3*x*x^m*x^(3*n)*e^m + 2*B*a*b*c*m^4*x*x^m*x^(
2*n)*e^m + A*b^2*c*m^4*x*x^m*x^(2*n)*e^m + B*a^2*d*m^4*x*x^m*x^(2*n)*e^m +
2*A*a*b*d*m^4*x*x^m*x^(2*n)*e^m + 16*B*a*b*c*m^3*n*x*x^m*x^(2*n)*e^m + 8*A*
b^2*c*m^3*n*x*x^m*x^(2*n)*e^m + 8*B*a^2*d*m^3*n*x*x^m*x^(2*n)*e^m + 16*A*a*
b*d*m^3*n*x*x^m*x^(2*n)*e^m + 38*B*a*b*c*m^2*n^2*x*x^m*x^(2*n)*e^m + 19*A*b
^2*c*m^2*n^2*x*x^m*x^(2*n)*e^m + 19*B*a^2*d*m^2*n^2*x*x^m*x^(2*n)*e^m + 38*
A*a*b*d*m^2*n^2*x*x^m*x^(2*n)*e^m + 24*B*a*b*c*m*n^3*x*x^m*x^(2*n)*e^m + 12
*A*b^2*c*m*n^3*x*x^m*x^(2*n)*e^m + 12*B*a^2*d*m*n^3*x*x^m*x^(2*n)*e^m + 24*
A*a*b*d*m*n^3*x*x^m*x^(2*n)*e^m + B*a^2*c*m^4*x*x^m*x^n*e^m + 2*A*a*b*c*m^4
*x*x^m*x^n*e^m + A*a^2*d*m^4*x*x^m*x^n*e^m + 9*B*a^2*c*m^3*n*x*x^m*x^n*e^m
+ 18*A*a*b*c*m^3*n*x*x^m*x^n*e^m + 9*A*a^2*d*m^3*n*x*x^m*x^n*e^m + 26*B*a^2
*c*m^2*n^2*x*x^m*x^n*e^m + 52*A*a*b*c*m^2*n^2*x*x^m*x^n*e^m + 26*A*a^2*d*m^
2*n^2*x*x^m*x^n*e^m + 24*B*a^2*c*m*n^3*x*x^m*x^n*e^m + 48*A*a*b*c*m*n^3*x*x
^m*x^n*e^m + 24*A*a^2*d*m*n^3*x*x^m*x^n*e^m + A*a^2*c*m^4*x*x^m*e^m + 10*A*
a^2*c*m^3*n*x*x^m*e^m + 35*A*a^2*c*m^2*n^2*x*x^m*e^m + 50*A*a^2*c*m*n^3*x*x
^m*e^m + 24*A*a^2*c*n^4*x*x^m*e^m + 4*B*b^2*d*m^3*x*x^m*x^(4*n)*e^m + 18*B*
b^2*d*m^2*n*x*x^m*x^(4*n)*e^m + 22*B*b^2*d*m*n^2*x*x^m*x^(4*n)*e^m + 6*B*b^
2*d*n^3*x*x^m*x^(4*n)*e^m + 4*B*b^2*c*m^3*x*x^m*x^(3*n)*e^m + 8*B*a*b*d*m^3
*x*x^m*x^(3*n)*e^m + 4*A*b^2*d*m^3*x*x^m*x^(3*n)*e^m + 21*B*b^2*c*m^2*n*x*x
^m*x^(3*n)*e^m + 42*B*a*b*d*m^2*n*x*x^m*x^(3*n)*e^m + 21*A*b^2*d*m^2*n*x*x^
m*x^(3*n)*e^m + 28*B*b^2*c*m*n^2*x*x^m*x^(3*n)*e^m + 56*B*a*b*d*m*n^2*x*x^m
*x^(3*n)*e^m + 28*A*b^2*d*m*n^2*x*x^m*x^(3*n)*e^m + 8*B*b^2*c*n^3*x*x^m*x^(
3*n)*e^m + 16*B*a*b*d*n^3*x*x^m*x^(3*n)*e^m + 8*A*b^2*d*n^3*x*x^m*x^(3*n)*e
^m + 8*B*a*b*c*m^3*x*x^m*x^(2*n)*e^m + 4*A*b^2*c*m^3*x*x^m*x^(2*n)*e^m + 4*
B*a^2*d*m^3*x*x^m*x^(2*n)*e^m + 8*A*a*b*d*m^3*x*x^m*x^(2*n)*e^m + 48*B*a*b*
c*m^2*n*x*x^m*x^(2*n)*e^m + 24*A*b^2*c*m^2*n*x*x^m*x^(2*n)*e^m + 24*B*a^2*d
*m^2*n*x*x^m*x^(2*n)*e^m + 48*A*a*b*d*m^2*n*x*x^m*x^(2*n)*e^m + 76*B*a*b*c*
m*n^2*x*x^m*x^(2*n)*e^m + 38*A*b^2*c*m*n^2*x*x^m*x^(2*n)*e^m + 38*B*a^2*d*m
*n^2*x*x^m*x^(2*n)*e^m + 76*A*a*b*d*m*n^2*x*x^m*x^(2*n)*e^m + 24*B*a*b*c*n^
3*x*x^m*x^(2*n)*e^m + 12*A*b^2*c*n^3*x*x^m*x^(2*n)*e^m + 12*B*a^2*d*n^3*x*x
^m*x^(2*n)*e^m + 24*A*a*b*d*n^3*x*x^m*x^(2*n)*e^m + 4*B*a^2*c*m^3*x*x^m*x^n
*e^m + 8*A*a*b*c*m^3*x*x^m*x^n*e^m + 4*A*a^2*d*m^3*x*x^m*x^n*e^m + 27*B*a^2
*c*m^2*n*x*x^m*x^n*e^m + 54*A*a*b*c*m^2*n*x*x^m*x^n*e^m + 27*A*a^2*d*m^2*n*
x*x^m*x^n*e^m + 52*B*a^2*c*m*n^2*x*x^m*x^n*e^m + 104*A*a*b*c*m*n^2*x*x^m*x^
n*e^m + 52*A*a^2*d*m*n^2*x*x^m*x^n*e^m + 24*B*a^2*c*n^3*x*x^m*x^n*e^m + 48*
A*a*b*c*n^3*x*x^m*x^n*e^m + 24*A*a^2*d*n^3*x*x^m*x^n*e^m + 4*A*a^2*c*m^3*x*
x^m*e^m + 30*A*a^2*c*m^2*n*x*x^m*e^m + 70*A*a^2*c*m*n^2*x*x^m*e^m + 50*A*a^
2*c*n^3*x*x^m*e^m + 6*B*b^2*d*m^2*x*x^m*x^(4*n)*e^m + 18*B*b^2*d*m*n*x*x^m*
x^(4*n)*e^m + 11*B*b^2*d*n^2*x*x^m*x^(4*n)*e^m + 6*B*b^2*c*m^2*x*x^m*x^(3*n
)*e^m + 12*B*a*b*d*m^2*x*x^m*x^(3*n)*e^m + 6*A*b^2*d*m^2*x*x^m*x^(3*n)*e^m
+ 21*B*b^2*c*m*n*x*x^m*x^(3*n)*e^m + 42*B*a*b*d*m*n*x*x^m*x^(3*n)*e^m + 21*
A*b^2*d*m*n*x*x^m*x^(3*n)*e^m + 14*B*b^2*c*n^2*x*x^m*x^(3*n)*e^m + 28*B*a*b
*d*n^2*x*x^m*x^(3*n)*e^m + 14*A*b^2*d*n^2*x*x^m*x^(3*n)*e^m + 12*B*a*b*c*m^
2*x*x^m*x^(2*n)*e^m + 6*A*b^2*c*m^2*x*x^m*x^(2*n)*e^m + 6*B*a^2*d*m^2*x*x^m
*x^(2*n)*e^m + 12*A*a*b*d*m^2*x*x^m*x^(2*n)*e^m + 48*B*a*b*c*m*n*x*x^m*x^(2
*n)*e^m + 24*A*b^2*c*m*n*x*x^m*x^(2*n)*e^m + 24*B*a^2*d*m*n*x*x^m*x^(2*n)*e
^m + 48*A*a*b*d*m*n*x*x^m*x^(2*n)*e^m + 38*B*a*b*c*n^2*x*x^m*x^(2*n)*e^m +
```

```
19*A*b^2*c*n^2*x*x^m*x^(2*n)*e^m + 19*B*a^2*d*n^2*x*x^m*x^(2*n)*e^m + 38*A*
a*b*d*n^2*x*x^m*x^(2*n)*e^m + 6*B*a^2*c*m^2*x*x^m*x^n*e^m + 12*A*a*b*c*m^2*
x*x^m*x^n*e^m + 6*A*a^2*d*m^2*x*x^m*x^n*e^m + 27*B*a^2*c*m*n*x*x^m*x^n*e^m
+ 54*A*a*b*c*m*n*x*x^m*x^n*e^m + 27*A*a^2*d*m*n*x*x^m*x^n*e^m + 26*B*a^2*c*
n^2*x*x^m*x^n*e^m + 52*A*a*b*c*n^2*x*x^m*x^n*e^m + 26*A*a^2*d*n^2*x*x^m*x^n
*e^m + 6*A*a^2*c*m^2*x*x^m*e^m + 30*A*a^2*c*m*n*x*x^m*e^m + 35*A*a^2*c*n^2*
x*x^m*e^m + 4*B*b^2*d*m*x*x^m*x^(4*n)*e^m + 6*B*b^2*d*n*x*x^m*x^(4*n)*e^m +
 4*B*b^2*c*m*x*x^m*x^(3*n)*e^m + 8*B*a*b*d*m*x*x^m*x^(3*n)*e^m + 4*A*b^2*d*
m*x*x^m*x^(3*n)*e^m + 7*B*b^2*c*n*x*x^m*x^(3*n)*e^m + 14*B*a*b*d*n*x*x^m*x^
(3*n)*e^m + 7*A*b^2*d*n*x*x^m*x^(3*n)*e^m + 8*B*a*b*c*m*x*x^m*x^(2*n)*e^m +
 4*A*b^2*c*m*x*x^m*x^(2*n)*e^m + 4*B*a^2*d*m*x*x^m*x^(2*n)*e^m + 8*A*a*b*d*
m*x*x^m*x^(2*n)*e^m + 16*B*a*b*c*n*x*x^m*x^(2*n)*e^m + 8*A*b^2*c*n*x*x^m*x^
(2*n)*e^m + 8*B*a^2*d*n*x*x^m*x^(2*n)*e^m + 16*A*a*b*d*n*x*x^m*x^(2*n)*e^m
+ 4*B*a^2*c*m*x*x^m*x^n*e^m + 8*A*a*b*c*m*x*x^m*x^n*e^m + 4*A*a^2*d*m*x*x^m
*x^n*e^m + 9*B*a^2*c*n*x*x^m*x^n*e^m + 18*A*a*b*c*n*x*x^m*x^n*e^m + 9*A*a^2
*d*n*x*x^m*x^n*e^m + 4*A*a^2*c*m*x*x^m*e^m + 10*A*a^2*c*n*x*x^m*e^m + B*b^2
*d*x*x^m*x^(4*n)*e^m + B*b^2*c*x*x^m*x^(3*n)*e^m + 2*B*a*b*d*x*x^m*x^(3*n)*
e^m + A*b^2*d*x*x^m*x^(3*n)*e^m + 2*B*a*b*c*x*x^m*x^(2*n)*e^m + A*b^2*c*x*x
^m*x^(2*n)*e^m + B*a^2*d*x*x^m*x^(2*n)*e^m + 2*A*a*b*d*x*x^m*x^(2*n)*e^m +
B*a^2*c*x*x^m*x^n*e^m + 2*A*a*b*c*x*x^m*x^n*e^m + A*a^2*d*x*x^m*x^n*e^m + A
*a^2*c*x*x^m*e^m)/(m^5 + 10*m^4*n + 35*m^3*n^2 + 50*m^2*n^3 + 24*m*n^4 + 5*
m^4 + 40*m^3*n + 105*m^2*n^2 + 100*m*n^3 + 24*n^4 + 10*m^3 + 60*m^2*n + 105
*m*n^2 + 50*n^3 + 10*m^2 + 40*m*n + 35*n^2 + 5*m + 10*n + 1)
```

maple [C] time = 0.14, size = 2410, normalized size = 15.06

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(b*x^n+a)^2*(A+B*x^n)*(d*x^n+c),x)
```

```
[Out] x*(2*B*a*b*d*m^4*(x^n)^3+7*B*b^2*c*m^3*n*(x^n)^3+14*B*b^2*c*m^2*n^2*(x^n)^3
+8*B*b^2*c*m*n^3*(x^n)^3+18*B*b^2*d*m^2*n*(x^n)^4+22*B*b^2*d*m*n^2*(x^n)^4+
2*A*a*b*d*m^4*(x^n)^2+2*A*a*b*c*m^4*x^n+6*B*b^2*d*m^3*n*(x^n)^4+11*B*b^2*d*
m^2*n^2*(x^n)^4+6*B*b^2*d*m*n^3*(x^n)^4+7*A*b^2*d*m^3*n*(x^n)^3+14*A*b^2*d*
m^2*n^2*(x^n)^3+8*A*b^2*d*m*n^3*(x^n)^3+9*A*a^2*d*m^3*n*x^n+26*A*a^2*d*m^2*
n^2*x^n+24*A*a^2*d*m*n^3*x^n+8*B*a*b*d*m^3*(x^n)^3+16*B*a*b*d*n^3*(x^n)^3+2
1*B*b^2*c*m^2*n*(x^n)^3+28*B*b^2*c*m*n^2*(x^n)^3+18*B*b^2*d*m*n*(x^n)^4+a^2
*A*c+A*b^2*d*(x^n)^3+B*b^2*c*(x^n)^3+A*b^2*c*(x^n)^2+B*a^2*d*(x^n)^2+A*a^2*
d*x^n+B*a^2*c*x^n+b^2*B*d*(x^n)^4+35*A*a^2*c*n^2+A*a^2*c*m^4+4*A*a^2*c*m^3+
50*A*a^2*c*n^3+6*A*a^2*c*m^2+24*A*a^2*c*n^4+12*B*a^2*d*m*n^3*(x^n)^2+2*B*a*
b*c*m^4*(x^n)^2+19*B*a^2*d*m^2*n^2*(x^n)^2+19*A*b^2*c*m^2*n^2*(x^n)^2+12*A*
b^2*c*m*n^3*(x^n)^2+21*A*b^2*d*m^2*n*(x^n)^3+28*A*b^2*d*m*n^2*(x^n)^3+8*B*a
^2*d*m^3*n*(x^n)^2+24*B*a^2*c*m*n^3*x^n+24*B*a^2*d*m^2*n*(x^n)^2+38*B*a^2*d
*m*n^2*(x^n)^2+8*A*b^2*c*m^3*n*(x^n)^2+24*A*b^2*c*m^2*n*(x^n)^2+38*A*b^2*c*
m*n^2*(x^n)^2+21*A*b^2*d*m*n*(x^n)^3+9*B*a^2*c*m^3*n*x^n+26*B*a^2*c*m^2*n^2
*x^n+8*A*a*b*d*m^3*(x^n)^2+24*A*a*b*d*n^3*(x^n)^2+12*B*a*b*d*m^2*(x^n)^3+28
*B*a*b*d*n^2*(x^n)^3+27*B*a^2*c*m^2*n*x^n+8*B*a*b*c*m^3*(x^n)^2+24*B*a*b*c*
n^3*(x^n)^2+52*A*a^2*d*m*n^2*x^n+8*A*a*b*c*m^3*x^n+48*A*a*b*c*n^3*x^n+12*A*
a*b*d*m^2*(x^n)^2+38*A*a*b*d*n^2*(x^n)^2+24*A*b^2*c*m*n*(x^n)^2+27*A*a^2*d*
m*n*x^n+52*B*a^2*c*m*n^2*x^n+24*B*a^2*d*m*n*(x^n)^2+12*B*a*b*c*m^2*(x^n)^2+
21*B*b^2*c*m*n*(x^n)^3+27*A*a^2*d*m^2*n*x^n+38*B*a*b*c*n^2*(x^n)^2+8*B*a*b*
d*(x^n)^3*m+14*B*a*b*d*(x^n)^3*n+8*A*a*b*d*(x^n)^2*m+16*A*a*b*d*(x^n)^2*n+2
7*B*a^2*c*m*n*x^n+8*B*a*b*c*(x^n)^2*m+18*A*a*b*c*x^n*n+12*A*a*b*c*m^2*x^n+5
2*A*a*b*c*n^2*x^n+8*A*a*b*c*x^n*m+16*B*a*b*c*(x^n)^2*n+4*a^2*A*c*m+10*a^2*A
*c*n+4*A*b^2*d*(x^n)^3*m+7*A*b^2*d*(x^n)^3*n+4*B*a^2*c*m^3*x^n+24*B*a^2*c*n
^3*x^n+6*B*a^2*d*m^2*(x^n)^2+19*B*a^2*d*n^2*(x^n)^2+4*B*b^2*c*(x^n)^3*m+7*B
*b^2*c*(x^n)^3*n+6*A*a^2*d*m^2*x^n+4*A*b^2*d*m^3*(x^n)^3+8*A*b^2*d*n^3*(x^n
)^3+4*A*a^2*d*x^n*m+9*A*a^2*d*x^n*n+4*B*a^2*c*x^n*m+9*B*a^2*c*x^n*n+54*A*a*
b*c*m*n*x^n+38*A*a*b*d*m^2*n^2*(x^n)^2+24*A*a*b*d*m*n^3*(x^n)^2+16*B*a*b*c*
```

```
m^3*n*(x^n)^2+38*B*a*b*c*m^2*n^2*(x^n)^2+24*B*a*b*c*m*n^3*(x^n)^2+42*B*a*b*
d*m^2*n*(x^n)^3+76*A*a*b*d*m*n^2*(x^n)^2+48*B*a*b*c*m^2*n*(x^n)^2+76*B*a*b*
c*m*n^2*(x^n)^2+42*B*a*b*d*m*n*(x^n)^3+54*A*a*b*c*m^2*n*x^n+104*A*a*b*c*m*n
^2*x^n+48*A*a*b*d*m*n*(x^n)^2+48*B*a*b*c*m*n*(x^n)^2+14*B*a*b*d*m^3*n*(x^n)
^3+28*B*a*b*d*m^2*n^2*(x^n)^3+16*B*a*b*d*m*n^3*(x^n)^3+16*A*a*b*d*m^3*n*(x^
n)^2+56*B*a*b*d*m*n^2*(x^n)^3+18*A*a*b*c*m^3*n*x^n+52*A*a*b*c*m^2*n^2*x^n+4
8*A*a*b*c*m*n^3*x^n+48*A*a*b*d*m^2*n*(x^n)^2+B*b^2*d*m^4*(x^n)^4+A*b^2*d*m^
4*(x^n)^3+B*b^2*c*m^4*(x^n)^3+4*B*b^2*d*m^3*(x^n)^4+6*B*b^2*d*n^3*(x^n)^4+A
*b^2*c*m^4*(x^n)^2+2*(x^n)^2*B*a*b*c+2*x^n*c*a*b*A+2*(x^n)^2*A*a*b*d+30*A*a
^2*c*m^2*n+70*A*a^2*c*m*n^2+30*A*a^2*c*m*n+2*(x^n)^3*B*a*b*d+10*A*a^2*c*m^3
*n+35*A*a^2*c*m^2*n^2+50*A*a^2*c*m*n^3+14*B*b^2*c*n^2*(x^n)^3+4*m*b^2*B*d*(
x^n)^4+6*b^2*B*d*(x^n)^4*n+4*A*a^2*d*m^3*x^n+24*A*a^2*d*n^3*x^n+6*A*b^2*c*m
^2*(x^n)^2+19*A*b^2*c*n^2*(x^n)^2+6*A*b^2*d*m^2*(x^n)^3+14*A*b^2*d*n^2*(x^n
)^3+B*a^2*c*m^4*x^n+4*B*a^2*d*m^3*(x^n)^2+12*B*a^2*d*n^3*(x^n)^2+6*B*b^2*c*
m^2*(x^n)^3+B*a^2*d*m^4*(x^n)^2+4*B*b^2*c*m^3*(x^n)^3+8*B*b^2*c*n^3*(x^n)^3
+6*B*b^2*d*m^2*(x^n)^4+11*B*b^2*d*n^2*(x^n)^4+A*a^2*d*m^4*x^n+4*A*b^2*c*m^3
*(x^n)^2+12*A*b^2*c*n^3*(x^n)^2+26*A*a^2*d*n^2*x^n+4*A*b^2*c*(x^n)^2*m+8*A*
b^2*c*(x^n)^2*n+6*B*a^2*c*m^2*x^n+26*B*a^2*c*n^2*x^n+4*B*a^2*d*(x^n)^2*m+8*
B*a^2*d*(x^n)^2*n)/(m+1)/(m+n+1)/(m+2*n+1)/(m+3*n+1)/(1+m+4*n)*exp(1/2*m*(-
I*Pi*csgn(I*e*x)^3+I*Pi*csgn(I*e*x)^2*csgn(I*e)+I*Pi*csgn(I*e*x)^2*csgn(I*x
)-I*Pi*csgn(I*e*x)*csgn(I*e)*csgn(I*x)+2*ln(e)+2*ln(x)))
```

maxima [B] time = 0.86, size = 332, normalized size = 2.08

$$\frac{Bb^2de^mxe^{\left(m\log(x)+4n\log(x)\right)}}{m+4n+1}+\frac{Bb^2ce^mxe^{\left(m\log(x)+3n\log(x)\right)}}{m+3n+1}+\frac{2\,Babde^mxe^{\left(m\log(x)+3n\log(x)\right)}}{m+3n+1}+\frac{Ab^2de^mxe^{\left(m\log(x)+3n\log(x)\right)}}{m+3n+1}+\frac{2\,Babce^mxe^{\left(m\log(x)+2n\log(x)\right)}}{m+2n+1}+\frac{Ab^2ce^mxe^{\left(m\log(x)+2n\log(x)\right)}}{m+2n+1}+\frac{Ba^2de^mxe^{\left(m\log(x)+2n\log(x)\right)}}{m+2n+1}+\frac{2\,Aabde^mxe^{\left(m\log(x)+2n\log(x)\right)}}{m+2n+1}+\frac{Ba^2ce^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{2\,Aabce^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{Aa^2de^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{(ex)^{m+1}Aa^2c}{e(m+1)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)^2*(A+B*x^n)*(c+d*x^n),x, algorithm="maxima")
```

```
[Out] B*b^2*d*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) + B*b^2*c*e^m*x*e^(m*
log(x) + 3*n*log(x))/(m + 3*n + 1) + 2*B*a*b*d*e^m*x*e^(m*log(x) + 3*n*log(
x))/(m + 3*n + 1) + A*b^2*d*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) +
 2*B*a*b*c*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + A*b^2*c*e^m*x*e^
(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + B*a^2*d*e^m*x*e^(m*log(x) + 2*n*log
(x))/(m + 2*n + 1) + 2*A*a*b*d*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1
) + B*a^2*c*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1) + 2*A*a*b*c*e^m*x*e^(
m*log(x) + n*log(x))/(m + n + 1) + A*a^2*d*e^m*x*e^(m*log(x) + n*log(x))/(m
 + n + 1) + (e*x)^(m + 1)*A*a^2*c/(e*(m + 1))
```

mupad [B] time = 5.23, size = 588, normalized size = 3.68

$$\frac{x\,x^{2n}\,(e\,x)^m\,\left(A\,b^2\,c+B\,a^2\,d+2\,A\,a\,b\,d+2\,B\,a\,b\,c\right)\,\left(m^3+8\,m^2\,n+3\,m^2+19\,m\,n^2+16\,m\,n+3\,m+12\,n^3+19\,n^2+8\,n+1\right)}{m^4+10\,m^3\,n+4\,m^3+35\,m^2\,n^2+30\,m^2\,n+6\,m^2+50\,m\,n^3+70\,m\,n^2+30\,m\,n+4\,m+24\,n^4+50\,n^3+35\,n^2+10\,n+1}+\frac{A\,a^2\,c\,x\,(e\,x)^m}{m+1}+\frac{a\,x\,x^n\,(e\,x)^m\,(A\,a\,d+2\,A\,b\,c+B\,a\,c)\,\left(m^3+9\,m^2\,n+3\,m^2+26\,m\,n^2+18\,m\,n+3\,m+24\,n^3+26\,n^2+9\,n+1\right)}{m^4+10\,m^3\,n+4\,m^3+35\,m^2\,n^2+30\,m^2\,n+6\,m^2+50\,m\,n^3+70\,m\,n^2+30\,m\,n+4\,m+24\,n^4+50\,n^3+35\,n^2+10\,n+1}+\frac{b\,x\,x^{3n}\,(e\,x)^m\,(A\,b\,d+2\,B\,a\,d+B\,b\,c)\,\left(m^3+7\,m^2\,n+3\,m^2+14\,m\,n^2+14\,m\,n+3\,m+8\,n^3+14\,n^2+7\,n+1\right)}{m^4+10\,m^3\,n+4\,m^3+35\,m^2\,n^2+30\,m^2\,n+6\,m^2+50\,m\,n^3+70\,m\,n^2+30\,m\,n+4\,m+24\,n^4+50\,n^3+35\,n^2+10\,n+1}+\frac{B\,b^2\,d\,x\,x^{4n}\,(e\,x)^m\,\left(m^3+6\,m^2\,n+3\,m^2+11\,m\,n^2+12\,m\,n+3\,m+6\,n^3+11\,n^2+6\,n+1\right)}{m^4+10\,m^3\,n+4\,m^3+35\,m^2\,n^2+30\,m^2\,n+6\,m^2+50\,m\,n^3+70\,m\,n^2+30\,m\,n+4\,m+24\,n^4+50\,n^3+35\,n^2+10\,n+1}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(A + B*x^n)*(a + b*x^n)^2*(c + d*x^n),x)
```

```
[Out] (x*x^(2*n)*(e*x)^m*(A*b^2*c + B*a^2*d + 2*A*a*b*d + 2*B*a*b*c)*(3*m + 8*n +
 16*m*n + 19*m*n^2 + 8*m^2*n + 3*m^2 + m^3 + 19*n^2 + 12*n^3 + 1))/(4*m + 1
0*n + 30*m*n + 70*m*n^2 + 30*m^2*n + 50*m*n^3 + 10*m^3*n + 6*m^2 + 4*m^3 +
m^4 + 35*n^2 + 50*n^3 + 24*n^4 + 35*m^2*n^2 + 1) + (A*a^2*c*x*(e*x)^m)/(m +
 1) + (a*x*x^n*(e*x)^m*(A*a*d + 2*A*b*c + B*a*c)*(3*m + 9*n + 18*m*n + 26*m
*n^2 + 9*m^2*n + 3*m^2 + m^3 + 26*n^2 + 24*n^3 + 1))/(4*m + 10*n + 30*m*n +
 70*m*n^2 + 30*m^2*n + 50*m*n^3 + 10*m^3*n + 6*m^2 + 4*m^3 + m^4 + 35*n^2 +
 50*n^3 + 24*n^4 + 35*m^2*n^2 + 1) + (b*x*x^(3*n)*(e*x)^m*(A*b*d + 2*B*a*d
+ B*b*c)*(3*m + 7*n + 14*m*n + 14*m*n^2 + 7*m^2*n + 3*m^2 + m^3 + 14*n^2 +
8*n^3 + 1))/(4*m + 10*n + 30*m*n + 70*m*n^2 + 30*m^2*n + 50*m*n^3 + 10*m^3*
n + 6*m^2 + 4*m^3 + m^4 + 35*n^2 + 50*n^3 + 24*n^4 + 35*m^2*n^2 + 1) + (B*b
^2*d*x*x^(4*n)*(e*x)^m*(3*m + 6*n + 12*m*n + 11*m*n^2 + 6*m^2*n + 3*m^2 + m
^3 + 11*n^2 + 6*n^3 + 1))/(4*m + 10*n + 30*m*n + 70*m*n^2 + 30*m^2*n + 50*m
```

```
*n^3 + 10*m^3*n + 6*m^2 + 4*m^3 + m^4 + 35*n^2 + 50*n^3 + 24*n^4 + 35*m^2*n
^2 + 1)
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)**m*(a+b*x**n)**2*(A+B*x**n)*(c+d*x**n),x)

[Out] Timed out
```

3.3 $\int (ex)^m (a + bx^n) (A + Bx^n) (c + dx^n) \, dx$

Optimal. Leaf size=108

$$\frac{x^{n+1}(ex)^m(aAd+aBc+Abc)}{m+n+1}+\frac{x^{2n+1}(ex)^m(aBd+Abd+bBc)}{m+2n+1}+\frac{aAc(ex)^{m+1}}{e(m+1)}+\frac{bBdx^{3n+1}(ex)^m}{m+3n+1}$$

Rubi [A] time = 0.08, antiderivative size = 108, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 3, integrand size = 27, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.111, Rules used = {570, 20, 30}

$$\frac{x^{n+1}(ex)^m(aAd+aBc+Abc)}{m+n+1}+\frac{x^{2n+1}(ex)^m(aBd+Abd+bBc)}{m+2n+1}+\frac{aAc(ex)^{m+1}}{e(m+1)}+\frac{bBdx^{3n+1}(ex)^m}{m+3n+1}$$

Antiderivative was successfully verified.

```
[In] Int[(e*x)^m*(a + b*x^n)*(A + B*x^n)*(c + d*x^n),x]
```

```
[Out] ((A*b*c + a*B*c + a*A*d)*x^(1 + n)*(e*x)^m)/(1 + m + n) + ((b*B*c + A*b*d +
 a*B*d)*x^(1 + 2*n)*(e*x)^m)/(1 + m + 2*n) + (b*B*d*x^(1 + 3*n)*(e*x)^m)/(1
 + m + 3*n) + (a*A*c*(e*x)^(1 + m))/(e*(1 + m))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 30

```
Int[(x_)^(m_.), x_Symbol] :> Simp[x^(m + 1)/(m + 1), x] /; FreeQ[m, x] && N
eQ[m, -1]
```

Rule 570

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

Rubi steps

$$\begin{aligned}\int (ex)^m (a+bx^n)(A+Bx^n)(c+dx^n)\,dx &= \int \left(aAc(ex)^m+(Abc+aBc+aAd)x^n(ex)^m+(bBc+Abd+aBd)x\right.\\ &= \frac{aAc(ex)^{1+m}}{e(1+m)}+(bBd)\int x^{3n}(ex)^m\,dx+(Abc+aBc+aAd)\int x^n(ex\\ &= \frac{aAc(ex)^{1+m}}{e(1+m)}+\left(bBdx^{-m}(ex)^m\right)\int x^{m+3n}\,dx+((Abc+aBc+aAd)x\\ &= \frac{(Abc+aBc+aAd)x^{1+n}(ex)^m}{1+m+n}+\frac{(bBc+Abd+aBd)x^{1+2n}(ex)^m}{1+m+2n}+\end{aligned}$$

Mathematica [A] time = 0.25, size = 84, normalized size = 0.78

$$x(ex)^m\left(\frac{x^{2n}(aBd+Abd+bBc)}{m+2n+1}+\frac{x^n(aAd+aBc+Abc)}{m+n+1}+\frac{aAc}{m+1}+\frac{bBdx^{3n}}{m+3n+1}\right)$$

Antiderivative was successfully verified.

```
[In] Integrate[(e*x)^m*(a + b*x^n)*(A + B*x^n)*(c + d*x^n),x]
```

```
[Out] x*(e*x)^m*((a*A*c)/(1 + m) + ((A*b*c + a*B*c + a*A*d)*x^n)/(1 + m + n) + ((
b*B*c + A*b*d + a*B*d)*x^(2*n))/(1 + m + 2*n) + (b*B*d*x^(3*n))/(1 + m + 3*
n))
```

IntegrateAlgebraic [F] time = 0.11, size = 0, normalized size = 0.00

$$\int (ex)^m (a+bx^n)(A+Bx^n)(c+dx^n)\, dx$$

Verification is not applicable to the result.

```
[In] IntegrateAlgebraic[(e*x)^m*(a + b*x^n)*(A + B*x^n)*(c + d*x^n),x]
```

```
[Out] Defer[IntegrateAlgebraic][(e*x)^m*(a + b*x^n)*(A + B*x^n)*(c + d*x^n), x]
```

fricas [B] time = 0.45, size = 562, normalized size = 5.20

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)*(A+B*x^n)*(c+d*x^n),x, algorithm="fricas")
```

```
[Out] ((B*b*d*m^3 + 3*B*b*d*m^2 + 3*B*b*d*m + B*b*d + 2*(B*b*d*m + B*b*d)*n^2 + 3
*(B*b*d*m^2 + 2*B*b*d*m + B*b*d)*n)*x*x^(3*n)*e^(m*log(e) + m*log(x)) + ((B
*b*c + (B*a + A*b)*d)*m^3 + B*b*c + 3*(B*b*c + (B*a + A*b)*d)*m^2 + 3*(B*b*
c + (B*a + A*b)*d + (B*b*c + (B*a + A*b)*d)*m)*n^2 + (B*a + A*b)*d + 3*(B*b
*c + (B*a + A*b)*d)*m + 4*(B*b*c + (B*b*c + (B*a + A*b)*d)*m^2 + (B*a + A*b
)*d + 2*(B*b*c + (B*a + A*b)*d)*m)*n)*x*x^(2*n)*e^(m*log(e) + m*log(x)) + (
(A*a*d + (B*a + A*b)*c)*m^3 + A*a*d + 3*(A*a*d + (B*a + A*b)*c)*m^2 + 6*(A*
a*d + (B*a + A*b)*c + (A*a*d + (B*a + A*b)*c)*m)*n^2 + (B*a + A*b)*c + 3*(A
*a*d + (B*a + A*b)*c)*m + 5*(A*a*d + (A*a*d + (B*a + A*b)*c)*m^2 + (B*a + A
*b)*c + 2*(A*a*d + (B*a + A*b)*c)*m)*n)*x*x^n*e^(m*log(e) + m*log(x)) + (A*
a*c*m^3 + 6*A*a*c*n^3 + 3*A*a*c*m^2 + 3*A*a*c*m + A*a*c + 11*(A*a*c*m + A*a
*c)*n^2 + 6*(A*a*c*m^2 + 2*A*a*c*m + A*a*c)*n)*x*e^(m*log(e) + m*log(x)))/(
m^4 + 6*(m + 1)*n^3 + 4*m^3 + 11*(m^2 + 2*m + 1)*n^2 + 6*m^2 + 6*(m^3 + 3*m
^2 + 3*m + 1)*n + 4*m + 1)
```

giac [B] time = 1.80, size = 1290, normalized size = 11.94

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)*(A+B*x^n)*(c+d*x^n),x, algorithm="giac")
```

```
[Out] (B*b*d*m^3*x*x^m*x^(3*n)*e^m + 3*B*b*d*m^2*n*x*x^m*x^(3*n)*e^m + 2*B*b*d*m*
n^2*x*x^m*x^(3*n)*e^m + B*b*c*m^3*x*x^m*x^(2*n)*e^m + B*a*d*m^3*x*x^m*x^(2*
n)*e^m + A*b*d*m^3*x*x^m*x^(2*n)*e^m + 4*B*b*c*m^2*n*x*x^m*x^(2*n)*e^m + 4*
B*a*d*m^2*n*x*x^m*x^(2*n)*e^m + 4*A*b*d*m^2*n*x*x^m*x^(2*n)*e^m + 3*B*b*c*m
*n^2*x*x^m*x^(2*n)*e^m + 3*B*a*d*m*n^2*x*x^m*x^(2*n)*e^m + 3*A*b*d*m*n^2*x*
x^m*x^(2*n)*e^m + B*a*c*m^3*x*x^m*x^n*e^m + A*b*c*m^3*x*x^m*x^n*e^m + A*a*d
*m^3*x*x^m*x^n*e^m + 5*B*a*c*m^2*n*x*x^m*x^n*e^m + 5*A*b*c*m^2*n*x*x^m*x^n*
e^m + 5*A*a*d*m^2*n*x*x^m*x^n*e^m + 6*B*a*c*m*n^2*x*x^m*x^n*e^m + 6*A*b*c*m
*n^2*x*x^m*x^n*e^m + 6*A*a*d*m*n^2*x*x^m*x^n*e^m + A*a*c*m^3*x*x^m*e^m + 6*
A*a*c*m^2*n*x*x^m*e^m + 11*A*a*c*m*n^2*x*x^m*e^m + 6*A*a*c*n^3*x*x^m*e^m +
3*B*b*d*m^2*x*x^m*x^(3*n)*e^m + 6*B*b*d*m*n*x*x^m*x^(3*n)*e^m + 2*B*b*d*n^2
*x*x^m*x^(3*n)*e^m + 3*B*b*c*m^2*x*x^m*x^(2*n)*e^m + 3*B*a*d*m^2*x*x^m*x^(2
*n)*e^m + 3*A*b*d*m^2*x*x^m*x^(2*n)*e^m + 8*B*b*c*m*n*x*x^m*x^(2*n)*e^m + 8
```

```
*B*a*d*m*n*x*x^m*x^(2*n)*e^m + 8*A*b*d*m*n*x*x^m*x^(2*n)*e^m + 3*B*b*c*n^2*
x*x^m*x^(2*n)*e^m + 3*B*a*d*n^2*x*x^m*x^(2*n)*e^m + 3*A*b*d*n^2*x*x^m*x^(2*
n)*e^m + 3*B*a*c*m^2*x*x^m*x^n*e^m + 3*A*b*c*m^2*x*x^m*x^n*e^m + 3*A*a*d*m^
2*x*x^m*x^n*e^m + 10*B*a*c*m*n*x*x^m*x^n*e^m + 10*A*b*c*m*n*x*x^m*x^n*e^m +
 10*A*a*d*m*n*x*x^m*x^n*e^m + 6*B*a*c*n^2*x*x^m*x^n*e^m + 6*A*b*c*n^2*x*x^m
*x^n*e^m + 6*A*a*d*n^2*x*x^m*x^n*e^m + 3*A*a*c*m^2*x*x^m*e^m + 12*A*a*c*m*n
*x*x^m*e^m + 11*A*a*c*n^2*x*x^m*e^m + 3*B*b*d*m*x*x^m*x^(3*n)*e^m + 3*B*b*d
*n*x*x^m*x^(3*n)*e^m + 3*B*b*c*m*x*x^m*x^(2*n)*e^m + 3*B*a*d*m*x*x^m*x^(2*n
)*e^m + 3*A*b*d*m*x*x^m*x^(2*n)*e^m + 4*B*b*c*n*x*x^m*x^(2*n)*e^m + 4*B*a*d
*n*x*x^m*x^(2*n)*e^m + 4*A*b*d*n*x*x^m*x^(2*n)*e^m + 3*B*a*c*m*x*x^m*x^n*e^
m + 3*A*b*c*m*x*x^m*x^n*e^m + 3*A*a*d*m*x*x^m*x^n*e^m + 5*B*a*c*n*x*x^m*x^n
*e^m + 5*A*b*c*n*x*x^m*x^n*e^m + 5*A*a*d*n*x*x^m*x^n*e^m + 3*A*a*c*m*x*x^m*
e^m + 6*A*a*c*n*x*x^m*e^m + B*b*d*x*x^m*x^(3*n)*e^m + B*b*c*x*x^m*x^(2*n)*e
^m + B*a*d*x*x^m*x^(2*n)*e^m + A*b*d*x*x^m*x^(2*n)*e^m + B*a*c*x*x^m*x^n*e^
m + A*b*c*x*x^m*x^n*e^m + A*a*d*x*x^m*x^n*e^m + A*a*c*x*x^m*e^m)/(m^4 + 6*m
^3*n + 11*m^2*n^2 + 6*m*n^3 + 4*m^3 + 18*m^2*n + 22*m*n^2 + 6*n^3 + 6*m^2 +
 18*m*n + 11*n^2 + 4*m + 6*n + 1)
```

maple [C] time = 0.11, size = 891, normalized size = 8.25

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(b*x^n+a)*(A+B*x^n)*(d*x^n+c),x)
```

```
[Out] x*(a*A*c+B*b*c*(x^n)^2+B*a*d*(x^n)^2+A*b*d*(x^n)^2+b*B*d*(x^n)^3+B*a*c*x^n+
A*a*d*x^n+A*b*c*x^n+A*a*c*m^3+3*A*a*c*m^2+11*A*a*c*n^2+6*A*a*c*n^3+6*a*A*c*
n+10*A*b*c*m*n*x^n+10*B*a*c*m*n*x^n+10*A*a*d*m*n*x^n+6*B*a*c*m*n^2*x^n+8*B*
a*d*m*n*(x^n)^2+8*B*b*c*m*n*(x^n)^2+8*A*b*d*m*n*(x^n)^2+5*B*a*c*m^2*n*x^n+5
*A*b*c*m^2*n*x^n+6*A*b*c*m*n^2*x^n+6*A*a*d*m*n^2*x^n+5*A*a*d*m^2*n*x^n+3*B*
b*c*m*n^2*(x^n)^2+6*B*b*d*m*n*(x^n)^3+4*B*a*d*m^2*n*(x^n)^2+3*B*a*d*m*n^2*(
x^n)^2+4*B*b*c*m^2*n*(x^n)^2+3*A*b*d*m*n^2*(x^n)^2+4*A*b*d*m^2*n*(x^n)^2+3*
B*b*d*m^2*n*(x^n)^3+2*B*b*d*m*n^2*(x^n)^3+4*A*(x^n)^2*b*d*n+3*B*(x^n)^2*a*d
*m+4*B*(x^n)^2*a*d*n+3*B*(x^n)^2*b*c*m+4*B*(x^n)^2*b*c*n+3*A*x^n*a*d*m+5*A*
x^n*a*d*n+3*A*x^n*b*c*m+5*A*x^n*b*c*n+3*B*x^n*a*c*m+5*B*x^n*a*c*n+3*B*(x^n)
^3*b*d*m+3*B*(x^n)^3*b*d*n+3*A*(x^n)^2*b*d*m+3*A*a*c*m+6*A*a*c*m^2*n+11*A*a
*c*m*n^2+12*A*a*c*m*n+6*A*b*c*n^2*x^n+3*B*a*c*m^2*x^n+6*B*a*c*n^2*x^n+A*a*d
*m^3*x^n+A*b*c*m^3*x^n+3*A*b*d*m^2*(x^n)^2+3*A*b*d*n^2*(x^n)^2+B*a*c*m^3*x^
n+3*B*a*d*m^2*(x^n)^2+3*B*a*d*n^2*(x^n)^2+3*B*b*c*m^2*(x^n)^2+3*B*b*c*n^2*(
x^n)^2+3*A*a*d*m^2*x^n+6*A*a*d*n^2*x^n+3*A*b*c*m^2*x^n+B*b*d*m^3*(x^n)^3+A*
b*d*m^3*(x^n)^2+B*a*d*m^3*(x^n)^2+B*b*c*m^3*(x^n)^2+3*B*b*d*m^2*(x^n)^3+2*B
*b*d*n^2*(x^n)^3)/(m+1)/(m+n+1)/(m+2*n+1)/(m+3*n+1)*exp(1/2*(-I*Pi*csgn(I*e
)*csgn(I*x)*csgn(I*e*x)+I*Pi*csgn(I*e)*csgn(I*e*x)^2+I*Pi*csgn(I*x)*csgn(I*
e*x)^2-I*Pi*csgn(I*e*x)^3+2*ln(e)+2*ln(x))*m)
```

maxima [A] time = 0.70, size = 200, normalized size = 1.85

$$\frac{Bbde^m x e^{(m\log(x)+3n\log(x))}}{m+3n+1}+\frac{Bbce^m x e^{(m\log(x)+2n\log(x))}}{m+2n+1}+\frac{Bade^m x e^{(m\log(x)+2n\log(x))}}{m+2n+1}+\frac{Abde^m x e^{(m\log(x)+2n\log(x))}}{m+2n+1}+\frac{Bace^m x e^{(m\log(x)+n\log(x))}}{m+n+1}+\frac{Abce^m x e^{(m\log(x)+n\log(x))}}{m+n+1}+\frac{Aade^m x e^{(m\log(x)+n\log(x))}}{m+n+1}+\frac{(ex)^{m+1}Aac}{e(m+1)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)*(A+B*x^n)*(c+d*x^n),x, algorithm="maxima")
```

```
[Out] B*b*d*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + B*b*c*e^m*x*e^(m*log(
x) + 2*n*log(x))/(m + 2*n + 1) + B*a*d*e^m*x*e^(m*log(x) + 2*n*log(x))/(m +
 2*n + 1) + A*b*d*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + B*a*c*e^m
*x*e^(m*log(x) + n*log(x))/(m + n + 1) + A*b*c*e^m*x*e^(m*log(x) + n*log(x)
)/(m + n + 1) + A*a*d*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1) + (e*x)^(m
+ 1)*A*a*c/(e*(m + 1))
```

mupad [B] time = 4.96, size = 271, normalized size = 2.51

$$\frac{A\,a\,c\,x\,(e\,x)^m}{m+1}+\frac{x\,x^{2\,n}\,(e\,x)^m\,\left(A\,b\,d+B\,a\,d+B\,b\,c\right)\left(m^2+4\,m\,n+2\,m+3\,n^2+4\,n+1\right)}{m^3+6\,m^2\,n+3\,m^2+11\,m\,n^2+12\,m\,n+3\,m+6\,n^3+11\,n^2+6\,n+1}+\frac{x\,x^n\,(e\,x)^m\,\left(A\,a\,d+A\,b\,c+B\,a\,c\right)\left(m^2+5\,m\,n+2\,m+6\,n^2+5\,n+1\right)}{m^3+6\,m^2\,n+3\,m^2+11\,m\,n^2+12\,m\,n+3\,m+6\,n^3+11\,n^2+6\,n+1}+\frac{B\,b\,d\,x\,x^{3\,n}\,(e\,x)^m\,\left(m^2+3\,m\,n+2\,m+2\,n^2+3\,n+1\right)}{m^3+6\,m^2\,n+3\,m^2+11\,m\,n^2+12\,m\,n+3\,m+6\,n^3+11\,n^2+6\,n+1}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(A + B*x^n)*(a + b*x^n)*(c + d*x^n),x)
```

```
[Out] (A*a*c*x*(e*x)^m)/(m + 1) + (x*x^(2*n)*(e*x)^m*(A*b*d + B*a*d + B*b*c)*(2*m
 + 4*n + 4*m*n + m^2 + 3*n^2 + 1))/(3*m + 6*n + 12*m*n + 11*m*n^2 + 6*m^2*n
 + 3*m^2 + m^3 + 11*n^2 + 6*n^3 + 1) + (x*x^n*(e*x)^m*(A*a*d + A*b*c + B*a*
c)*(2*m + 5*n + 5*m*n + m^2 + 6*n^2 + 1))/(3*m + 6*n + 12*m*n + 11*m*n^2 +
6*m^2*n + 3*m^2 + m^3 + 11*n^2 + 6*n^3 + 1) + (B*b*d*x*x^(3*n)*(e*x)^m*(2*m
 + 3*n + 3*m*n + m^2 + 2*n^2 + 1))/(3*m + 6*n + 12*m*n + 11*m*n^2 + 6*m^2*n
 + 3*m^2 + m^3 + 11*n^2 + 6*n^3 + 1)
```

sympy [A] time = 88.27, size = 8500, normalized size = 78.70

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)**m*(a+b*x**n)*(A+B*x**n)*(c+d*x**n),x)
```

```
[Out] Piecewise(((A + B)*(a + b)*(c + d)*log(x)/e, Eq(m, -1) & Eq(n, 0)), ((A*a*c
*log(x) + A*a*d*x**n/n + A*b*c*x**n/n + A*b*d*x**(2*n)/(2*n) + B*a*c*x**n/n
 + B*a*d*x**(2*n)/(2*n) + B*b*c*x**(2*n)/(2*n) + B*b*d*x**(3*n)/(3*n))/e, E
q(m, -1)), (A*a*c*Piecewise((log(x), Eq(n, 0)), (-x**(-3*n)*(0**(1/n))**(-3
*n)/(3*n), Eq(e, 0**(1/n))), (-e**(-3*n)*x**(-3*n)/(3*n), True))/e + A*a*d*
Piecewise((log(x), Eq(n, 0)), (-x**n/(3*0**(1/n)*zoo**(1/n)*n*x**(3*n)*(0**
(1/n))**(3*n) - n*x**(3*n)*(0**(1/n))**(3*n)), Eq(e, 0**(1/n))), (-e**(-3*n
)*x**(-2*n)/(2*n), True))/e + A*b*c*Piecewise((log(x), Eq(n, 0)), (-x**n/(3
*0**(1/n)*zoo**(1/n)*n*x**(3*n)*(0**(1/n))**(3*n) - n*x**(3*n)*(0**(1/n))**
(3*n)), Eq(e, 0**(1/n))), (-e**(-3*n)*x**(-2*n)/(2*n), True))/e + A*b*d*Pie
cewise((log(x), Eq(n, 0)), (-x**(2*n)/(3*0**(1/n)*zoo**(1/n)*n*x**(3*n)*(0*
*(1/n))**(3*n) - 2*n*x**(3*n)*(0**(1/n))**(3*n)), Eq(e, 0**(1/n))), (-e**(-
3*n)*x**(-n)/n, True))/e + B*a*c*Piecewise((log(x), Eq(n, 0)), (-x**n/(3*0*
*(1/n)*zoo**(1/n)*n*x**(3*n)*(0**(1/n))**(3*n) - n*x**(3*n)*(0**(1/n))**(3*
n)), Eq(e, 0**(1/n))), (-e**(-3*n)*x**(-2*n)/(2*n), True))/e + B*a*d*Piecew
ise((log(x), Eq(n, 0)), (-x**(2*n)/(3*0**(1/n)*zoo**(1/n)*n*x**(3*n)*(0**(1
/n))**(3*n) - 2*n*x**(3*n)*(0**(1/n))**(3*n)), Eq(e, 0**(1/n))), (-e**(-3*n
)*x**(-n)/n, True))/e + B*b*c*Piecewise((log(x), Eq(n, 0)), (-x**(2*n)/(3*0
**(1/n)*zoo**(1/n)*n*x**(3*n)*(0**(1/n))**(3*n) - 2*n*x**(3*n)*(0**(1/n))**
(3*n)), Eq(e, 0**(1/n))), (-e**(-3*n)*x**(-n)/n, True))/e + B*b*d*Piecewise
((e**(-3*n)*log(x), Abs(x) < 1), (-e**(-3*n)*log(1/x), 1/Abs(x) < 1), (-e**
(-3*n)*meijerg(((), (1, 1)), ((0, 0), ()), x) + e**(-3*n)*meijerg(((1, 1),
()), ((), (0, 0)), x), True))/e, Eq(m, -3*n - 1)), (A*a*c*Piecewise((log(x)
, Eq(n, 0)), (-x**(-2*n)*(0**(1/n))**(-2*n)/(2*n), Eq(e, 0**(1/n))), (-e**(
-2*n)*x**(-2*n)/(2*n), True))/e + A*a*d*Piecewise((log(x), Eq(n, 0)), (-x**
n/(2*0**(1/n)*zoo**(1/n)*n*x**(2*n)*(0**(1/n))**(2*n) - n*x**(2*n)*(0**(1/n
))**(2*n)), Eq(e, 0**(1/n))), (-e**(-2*n)*x**(-n)/n, True))/e + A*b*c*Piece
wise((log(x), Eq(n, 0)), (-x**n/(2*0**(1/n)*zoo**(1/n)*n*x**(2*n)*(0**(1/n)
)**(2*n) - n*x**(2*n)*(0**(1/n))**(2*n)), Eq(e, 0**(1/n))), (-e**(-2*n)*x**
(-n)/n, True))/e + A*b*d*Piecewise((e**(-2*n)*log(x), Abs(x) < 1), (-e**(-2
*n)*log(1/x), 1/Abs(x) < 1), (-e**(-2*n)*meijerg(((), (1, 1)), ((0, 0), ())
, x) + e**(-2*n)*meijerg(((1, 1), ()), ((), (0, 0)), x), True))/e + B*a*c*P
iecewise((log(x), Eq(n, 0)), (-x**n/(2*0**(1/n)*zoo**(1/n)*n*x**(2*n)*(0**(
1/n))**(2*n) - n*x**(2*n)*(0**(1/n))**(2*n)), Eq(e, 0**(1/n))), (-e**(-2*n)
*x**(-n)/n, True))/e + B*a*d*Piecewise((e**(-2*n)*log(x), Abs(x) < 1), (-e*
*(-2*n)*log(1/x), 1/Abs(x) < 1), (-e**(-2*n)*meijerg(((), (1, 1)), ((0, 0),
 ()), x) + e**(-2*n)*meijerg(((1, 1), ()), ((), (0, 0)), x), True))/e + B*b
*c*Piecewise((e**(-2*n)*log(x), Abs(x) < 1), (-e**(-2*n)*log(1/x), 1/Abs(x)
```

```
 < 1), (-e**(-2*n)*meijerg(((), (1, 1)), ((0, 0), ()), x) + e**(-2*n)*meije
rg(((1, 1), ()), ((), (0, 0)), x), True))/e + B*b*d*Piecewise((log(x), Eq(n
, 0)), (-x**(3*n)/(2*0**(1/n)*zoo**(1/n)*n*x**(2*n)*(0**(1/n))**(2*n) - 3*n
*x**(2*n)*(0**(1/n))**(2*n)), Eq(e, 0**(1/n))), (e**(-2*n)*x**n/n, True))/e
, Eq(m, -2*n - 1)), (A*a*c*Piecewise((log(x), Eq(n, 0)), (-x**(-n)*(0**(1/n
))**(-n)/n, Eq(e, 0**(1/n))), (-e**(-n)*x**(-n)/n, True))/e + A*a*d*Piecewi
se((e**(-n)*log(x), Abs(x) < 1), (-e**(-n)*log(1/x), 1/Abs(x) < 1), (-e**(-
n)*meijerg(((), (1, 1)), ((0, 0), ()), x) + e**(-n)*meijerg(((1, 1), ()), (
(), (0, 0)), x), True))/e + A*b*c*Piecewise((e**(-n)*log(x), Abs(x) < 1), (
-e**(-n)*log(1/x), 1/Abs(x) < 1), (-e**(-n)*meijerg(((), (1, 1)), ((0, 0),
()), x) + e**(-n)*meijerg(((1, 1), ()), ((), (0, 0)), x), True))/e + A*b*d*
Piecewise((log(x), Eq(n, 0)), (-x**(2*n)/(0**(1/n)*zoo**(1/n)*n*x**n*(0**(1
/n))**n - 2*n*x**n*(0**(1/n))**n), Eq(e, 0**(1/n))), (e**(-n)*x**n/n, True)
)/e + B*a*c*Piecewise((e**(-n)*log(x), Abs(x) < 1), (-e**(-n)*log(1/x), 1/A
bs(x) < 1), (-e**(-n)*meijerg(((), (1, 1)), ((0, 0), ()), x) + e**(-n)*meij
erg(((1, 1), ()), ((), (0, 0)), x), True))/e + B*a*d*Piecewise((log(x), Eq(
n, 0)), (-x**(2*n)/(0**(1/n)*zoo**(1/n)*n*x**n*(0**(1/n))**n - 2*n*x**n*(0*
*(1/n))**n), Eq(e, 0**(1/n))), (e**(-n)*x**n/n, True))/e + B*b*c*Piecewise(
(log(x), Eq(n, 0)), (-x**(2*n)/(0**(1/n)*zoo**(1/n)*n*x**n*(0**(1/n))**n -
2*n*x**n*(0**(1/n))**n), Eq(e, 0**(1/n))), (e**(-n)*x**n/n, True))/e + B*b*
d*Piecewise((log(x), Eq(n, 0)), (-x**(3*n)/(0**(1/n)*zoo**(1/n)*n*x**n*(0**
(1/n))**n - 3*n*x**n*(0**(1/n))**n), Eq(e, 0**(1/n))), (e**(-n)*x**(2*n)/(2
*n), True))/e, Eq(m, -n - 1)), (A*a*c*e**m*m**3*x*x**m/(m**4 + 6*m**3*n + 4
*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n +
 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 6*A*a*c*e**m*m**2*n*x*x**m/(m**4 + 6*m
**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 +
 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*A*a*c*e**m*m**2*x*x**m/(m**
4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m
*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 11*A*a*c*e**m*m*n**2*x
*x**m/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n
**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 12*A*a*c*e**
m*m*n*x*x**m/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2
+ 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*A*a
*c*e**m*m*x*x**m/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m
**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 6
*A*a*c*e**m*n**3*x*x**m/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*
n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n +
 1) + 11*A*a*c*e**m*n**2*x*x**m/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 +
18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2
 + 6*n + 1) + 6*A*a*c*e**m*n*x*x**m/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**
2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*
n**2 + 6*n + 1) + A*a*c*e**m*x*x**m/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**
2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*
n**2 + 6*n + 1) + A*a*d*e**m*m**3*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 1
1*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*
n**3 + 11*n**2 + 6*n + 1) + 5*A*a*d*e**m*m**2*n*x*x**m*x**n/(m**4 + 6*m**3*
n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*
m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*A*a*d*e**m*m**2*x*x**m*x**n/(m*
*4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*
m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 6*A*a*d*e**m*m*n**2*x
*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 +
6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 10*A*a*
d*e**m*m*n*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n
 + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n +
1) + 3*A*a*d*e**m*m*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 +
18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2
 + 6*n + 1) + 6*A*a*d*e**m*n**2*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*
m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n*
*3 + 11*n**2 + 6*n + 1) + 5*A*a*d*e**m*n*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m
```

```
**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4
*m + 6*n**3 + 11*n**2 + 6*n + 1) + A*a*d*e**m*x*x**m*x**n/(m**4 + 6*m**3*n
+ 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*
n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + A*b*c*e**m*m**3*x*x**m*x**n/(m**4 +
 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n*
*2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 5*A*b*c*e**m*m**2*n*x*x**
m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*
n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*A*b*c*e**
m*m**2*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6
*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) +
 6*A*b*c*e**m*m*n**2*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 +
 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**
2 + 6*n + 1) + 10*A*b*c*e**m*m*n*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11
*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n
**3 + 11*n**2 + 6*n + 1) + 3*A*b*c*e**m*m*x*x**m*x**n/(m**4 + 6*m**3*n + 4*
m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n +
4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 6*A*b*c*e**m*n**2*x*x**m*x**n/(m**4 + 6
*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2
 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 5*A*b*c*e**m*n*x*x**m*x**n/
(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 +
22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + A*b*c*e**m*x*x**m*
x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n*
*3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + A*b*d*e**m*m*
*3*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6
*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) +
 4*A*b*d*e**m*m**2*n*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n*
*2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11
*n**2 + 6*n + 1) + 3*A*b*d*e**m*m**2*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m
**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4
*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*A*b*d*e**m*m*n**2*x*x**m*x**(2*n)/(m**
4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m
*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 8*A*b*d*e**m*m*n*x*x**
m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 +
6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*A*b*d
*e**m*m*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*
n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n +
 1) + 3*A*b*d*e**m*n**2*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2
*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 +
 11*n**2 + 6*n + 1) + 4*A*b*d*e**m*n*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m
**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4
*m + 6*n**3 + 11*n**2 + 6*n + 1) + A*b*d*e**m*x*x**m*x**(2*n)/(m**4 + 6*m**
3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 1
8*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + B*a*c*e**m*m**3*x*x**m*x**n/(m*
*4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*
m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 5*B*a*c*e**m*m**2*n*x
*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 +
6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*B*a*c
*e**m*m**2*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n
 + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n +
1) + 6*B*a*c*e**m*m*n**2*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n*
*2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11
*n**2 + 6*n + 1) + 10*B*a*c*e**m*m*n*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3
+ 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m +
 6*n**3 + 11*n**2 + 6*n + 1) + 3*B*a*c*e**m*m*x*x**m*x**n/(m**4 + 6*m**3*n
+ 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*
n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 6*B*a*c*e**m*n**2*x*x**m*x**n/(m**4
 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*
n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 5*B*a*c*e**m*n*x*x**m*x
**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**
```

```
3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + B*a*c*e**m*x*x
**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*
m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + B*a*d*e**
m*m**3*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n
 + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n +
1) + 4*B*a*d*e**m*m**2*n*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**
2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3
+ 11*n**2 + 6*n + 1) + 3*B*a*d*e**m*m**2*x*x**m*x**(2*n)/(m**4 + 6*m**3*n +
 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n
 + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*B*a*d*e**m*m*n**2*x*x**m*x**(2*n)/
(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 +
22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 8*B*a*d*e**m*m*n*x
*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**
2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*B
*a*d*e**m*m*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m
**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6
*n + 1) + 3*B*a*d*e**m*n**2*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*
m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n*
*3 + 11*n**2 + 6*n + 1) + 4*B*a*d*e**m*n*x*x**m*x**(2*n)/(m**4 + 6*m**3*n +
 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n
 + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + B*a*d*e**m*x*x**m*x**(2*n)/(m**4 + 6
*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2
 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + B*b*c*e**m*m**3*x*x**m*x**(
2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n*
*3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 4*B*b*c*e**m*
m**2*n*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n
 + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n +
1) + 3*B*b*c*e**m*m**2*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*
n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 +
11*n**2 + 6*n + 1) + 3*B*b*c*e**m*m*n**2*x*x**m*x**(2*n)/(m**4 + 6*m**3*n +
 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n
 + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 8*B*b*c*e**m*m*n*x*x**m*x**(2*n)/(m*
*4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*
m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*B*b*c*e**m*m*x*x**m
*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6
*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*B*b*c*
e**m*n**2*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**
2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n
 + 1) + 4*B*b*c*e**m*n*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*
n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 +
11*n**2 + 6*n + 1) + B*b*c*e**m*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 +
 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m +
6*n**3 + 11*n**2 + 6*n + 1) + B*b*d*e**m*m**3*x*x**m*x**(3*n)/(m**4 + 6*m**
3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 1
8*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*B*b*d*e**m*m**2*n*x*x**m*x**(
3*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n*
*3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*B*b*d*e**m*
m**2*x*x**m*x**(3*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n +
 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1)
 + 2*B*b*d*e**m*m*n**2*x*x**m*x**(3*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*
n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 +
11*n**2 + 6*n + 1) + 6*B*b*d*e**m*m*n*x*x**m*x**(3*n)/(m**4 + 6*m**3*n + 4*
m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n +
4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*B*b*d*e**m*m*x*x**m*x**(3*n)/(m**4 +
6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**
2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 2*B*b*d*e**m*n**2*x*x**m*x
**(3*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m
*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*B*b*d*e*
*m*n*x*x**m*x**(3*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n +
```

```
 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1)
 + B*b*d*e**m*x*x**m*x**(3*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18
*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 +
 6*n + 1), True))
```

3.4 $\int (ex)^m (A+Bx^n)(c+dx^n)\,dx$

Optimal. Leaf size=66

$$\frac{x^{n+1}(ex)^m(Ad+Bc)}{m+n+1}+\frac{Ac(ex)^{m+1}}{e(m+1)}+\frac{Bdx^{2n+1}(ex)^m}{m+2n+1}$$

Rubi [A] time = 0.04, antiderivative size = 66, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 3, integrand size = 20, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {448, 20, 30}

$$\frac{x^{n+1}(ex)^m(Ad+Bc)}{m+n+1}+\frac{Ac(ex)^{m+1}}{e(m+1)}+\frac{Bdx^{2n+1}(ex)^m}{m+2n+1}$$

Antiderivative was successfully verified.

```
[In] Int[(e*x)^m*(A + B*x^n)*(c + d*x^n),x]
```

```
[Out] ((B*c + A*d)*x^(1 + n)*(e*x)^m)/(1 + m + n) + (B*d*x^(1 + 2*n)*(e*x)^m)/(1
+ m + 2*n) + (A*c*(e*x)^(1 + m))/(e*(1 + m))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 30

```
Int[(x_)^(m_.), x_Symbol] :> Simp[x^(m + 1)/(m + 1), x] /; FreeQ[m, x] && N
eQ[m, -1]
```

Rule 448

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

Rubi steps

$$\begin{aligned}
\int (ex)^m (A+Bx^n)(c+dx^n)\,dx &= \int \left(Ac(ex)^m+(Bc+Ad)x^n(ex)^m+Bdx^{2n}(ex)^m\right)dx\\
&= \frac{Ac(ex)^{1+m}}{e(1+m)}+(Bd)\int x^{2n}(ex)^m\,dx+(Bc+Ad)\int x^n(ex)^m\,dx\\
&= \frac{Ac(ex)^{1+m}}{e(1+m)}+\left(Bdx^{-m}(ex)^m\right)\int x^{m+2n}\,dx+\left((Bc+Ad)x^{-m}(ex)^m\right)\int x^{m+n}\,dx\\
&= \frac{(Bc+Ad)x^{1+n}(ex)^m}{1+m+n}+\frac{Bdx^{1+2n}(ex)^m}{1+m+2n}+\frac{Ac(ex)^{1+m}}{e(1+m)}
\end{aligned}$$

Mathematica [A] time = 0.07, size = 49, normalized size = 0.74

$$x(ex)^m\left(\frac{x^n(Ad+Bc)}{m+n+1}+\frac{Ac}{m+1}+\frac{Bdx^{2n}}{m+2n+1}\right)$$

Antiderivative was successfully verified.

```
[In] Integrate[(e*x)^m*(A + B*x^n)*(c + d*x^n),x]
```

```
[Out] x*(e*x)^m*((A*c)/(1 + m) + ((B*c + A*d)*x^n)/(1 + m + n) + (B*d*x^(2*n))/(1
 + m + 2*n))
```

IntegrateAlgebraic [F] time = 0.07, size = 0, normalized size = 0.00

$$\int (ex)^m (A + Bx^n) (c + dx^n) \, dx$$

Verification is not applicable to the result.

```
[In] IntegrateAlgebraic[(e*x)^m*(A + B*x^n)*(c + d*x^n),x]
```

```
[Out] Defer[IntegrateAlgebraic][(e*x)^m*(A + B*x^n)*(c + d*x^n), x]
```

fricas [B] time = 0.44, size = 185, normalized size = 2.80

$$\frac{\left(Bdm^2 + 2\,Bdm + Bd + (Bdm + Bd)n\right)xx^{2\,n}e^{\left(m\log(e) + m\log(x)\right)} + \left((Bc + Ad)m^2 + Bc + Ad + 2\,(Bc + Ad)m + 2\,(Bc + Ad + (Bc + Ad)m)n\right)xx^n e^{\left(m\log(e) + m\log(x)\right)} + \left(Acm^2 + 2\,Acn^2 + 2\,Acm + Ac + 3\,(Acm + Ac)n\right)xe^{\left(m\log(e) + m\log(x)\right)}}{m^3 + 2\,(m + 1)n^2 + 3\,m^2 + 3\,\left(m^2 + 2\,m + 1\right)n + 3\,m + 1}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(A+B*x^n)*(c+d*x^n),x, algorithm="fricas")
```

```
[Out] ((B*d*m^2 + 2*B*d*m + B*d + (B*d*m + B*d)*n)*x*x^(2*n)*e^(m*log(e) + m*log(
x)) + ((B*c + A*d)*m^2 + B*c + A*d + 2*(B*c + A*d)*m + 2*(B*c + A*d + (B*c
+ A*d)*m)*n)*x*x^n*e^(m*log(e) + m*log(x)) + (A*c*m^2 + 2*A*c*n^2 + 2*A*c*m
 + A*c + 3*(A*c*m + A*c)*n)*x*e^(m*log(e) + m*log(x)))/(m^3 + 2*(m + 1)*n^2
 + 3*m^2 + 3*(m^2 + 2*m + 1)*n + 3*m + 1)
```

giac [B] time = 0.45, size = 327, normalized size = 4.95

$$\frac{Bdm^2xx^mx^{2\,n}e^m + Bdmnxx^mx^{2\,n}e^m + Bcm^2xx^mx^ne^m + Adm^2xx^mx^ne^m + 2\,Bcmnxx^mx^ne^m + 2\,Admnxx^mx^ne^m + Acm^2xx^me^m + 3\,Acmnxx^me^m + 2\,Acn^2xx^me^m + 2\,Bdmxx^mx^{2\,n}e^m + Bdnxx^mx^{2\,n}e^m + 2\,Bcmxx^mx^ne^m + 2\,Admxx^mx^ne^m + 2\,Bcnxx^mx^ne^m + 2\,Adnxx^mx^ne^m + 2\,Acmxx^me^m + 3\,Acnxx^me^m + Bdxx^mx^{2\,n}e^m + Bcxx^mx^ne^m + Adxx^mx^ne^m + Acxx^me^m}{m^3 + 3\,m^2n + 2\,mn^2 + 3\,m^2 + 6\,mn + 2\,n^2 + 3\,m + 3\,n + 1}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(A+B*x^n)*(c+d*x^n),x, algorithm="giac")
```

```
[Out] (B*d*m^2*x*x^m*x^(2*n)*e^m + B*d*m*n*x*x^m*x^(2*n)*e^m + B*c*m^2*x*x^m*x^n*
e^m + A*d*m^2*x*x^m*x^n*e^m + 2*B*c*m*n*x*x^m*x^n*e^m + 2*A*d*m*n*x*x^m*x^n
*e^m + A*c*m^2*x*x^m*e^m + 3*A*c*m*n*x*x^m*e^m + 2*A*c*n^2*x*x^m*e^m + 2*B*
d*m*x*x^m*x^(2*n)*e^m + B*d*n*x*x^m*x^(2*n)*e^m + 2*B*c*m*x*x^m*x^n*e^m + 2
*A*d*m*x*x^m*x^n*e^m + 2*B*c*n*x*x^m*x^n*e^m + 2*A*d*n*x*x^m*x^n*e^m + 2*A*
c*m*x*x^m*e^m + 3*A*c*n*x*x^m*e^m + B*d*x*x^m*x^(2*n)*e^m + B*c*x*x^m*x^n*e
^m + A*d*x*x^m*x^n*e^m + A*c*x*x^m*e^m)/(m^3 + 3*m^2*n + 2*m*n^2 + 3*m^2 +
6*m*n + 2*n^2 + 3*m + 3*n + 1)
```

maple [C] time = 0.12, size = 262, normalized size = 3.97

$$\frac{\left(Ad\,m^2x^n + 2Admn\,x^n + Bc\,m^2x^n + 2Bcmn\,x^n + Bd\,m^2x^{2n} + Bdmn\,x^{2n} + Ac\,m^2 + 3Acmn + 2Ac\,n^2 + 2Adm\,x^n + 2Adn\,x^n + 2Bcm\,x^n + 2Bcn\,x^n + 2Bdm\,x^{2n} + Bdn\,x^{2n} + 2Acm + 3Acn + Ad\,x^n + Bc\,x^n + Bd\,x^{2n} + Ac\right)x\,\mathrm{e}^{\frac{\left(-i\pi\,\mathrm{csgn}(ie)\mathrm{csgn}(ix)\mathrm{csgn}(iex)+i\pi\,\mathrm{csgn}(ie)\mathrm{csgn}(iex)^2+i\pi\,\mathrm{csgn}(ix)\mathrm{csgn}(iex)^2-i\pi\mathrm{csgn}(iex)^3+2\ln(e)+2\ln(x)\right)m}{2}}}{(m + 1)\,(m + n + 1)\,(m + 2n + 1)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(A+B*x^n)*(d*x^n+c),x)
```

```
[Out] x*(B*d*m^2*(x^n)^2+B*d*m*n*(x^n)^2+A*d*m^2*x^n+2*A*d*m*n*x^n+B*c*m^2*x^n+2*
B*c*m*n*x^n+2*B*(x^n)^2*d*m+B*(x^n)^2*d*n+A*c*m^2+3*A*c*m*n+2*A*c*n^2+2*A*x
^n*d*m+2*A*x^n*d*n+2*B*x^n*c*m+2*B*x^n*c*n+d*(x^n)^2*B+2*A*c*m+3*A*c*n+d*x^
n*A+c*B*x^n+A*c)/(m+1)/(m+n+1)/(m+2*n+1)*exp(1/2*(-I*Pi*csgn(I*e)*csgn(I*x)
*csgn(I*e*x)+I*Pi*csgn(I*e)*csgn(I*e*x)^2+I*Pi*csgn(I*x)*csgn(I*e*x)^2-I*Pi
*csgn(I*e*x)^3+2*ln(e)+2*ln(x))*m)
```

maxima [A] time = 0.60, size = 91, normalized size = 1.38

$$\frac{Bde^{m}xe^{\left(m\log(x)+2n\log(x)\right)}}{m+2n+1}+\frac{Bce^{m}xe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{Ade^{m}xe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{(ex)^{m+1}Ac}{e(m+1)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(A+B*x^n)*(c+d*x^n),x, algorithm="maxima")
```

```
[Out] B*d*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + B*c*e^m*x*e^(m*log(x) +
 n*log(x))/(m + n + 1) + A*d*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1) + (e
*x)^(m + 1)*A*c/(e*(m + 1))
```

mupad [B] time = 4.83, size = 91, normalized size = 1.38

$$(e\,x)^m\,\left(\frac{A\,c\,x}{m+1}+\frac{x\,x^n\,\left(A\,d+B\,c\right)\,\left(m+2\,n+1\right)}{m^2+3\,m\,n+2\,m+2\,n^2+3\,n+1}+\frac{B\,d\,x\,x^{2\,n}\,\left(m+n+1\right)}{m^2+3\,m\,n+2\,m+2\,n^2+3\,n+1}\right)$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(A + B*x^n)*(c + d*x^n),x)
```

```
[Out] (e*x)^m*((A*c*x)/(m + 1) + (x*x^n*(A*d + B*c)*(m + 2*n + 1))/(2*m + 3*n + 3
*m*n + m^2 + 2*n^2 + 1) + (B*d*x*x^(2*n)*(m + n + 1))/(2*m + 3*n + 3*m*n +
m^2 + 2*n^2 + 1))
```

sympy [A] time = 29.33, size = 1698, normalized size = 25.73

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)**m*(A+B*x**n)*(c+d*x**n),x)
```

```
[Out] Piecewise(((A + B)*(c + d)*log(x)/e, Eq(m, -1) & Eq(n, 0)), ((A*c*log(x) +
A*d*x**n/n + B*c*x**n/n + B*d*x**(2*n)/(2*n))/e, Eq(m, -1)), (A*c*Piecewise
((log(x), Eq(n, 0)), (-x**(-2*n)*(0**(1/n))**(-2*n)/(2*n), Eq(e, 0**(1/n)))
, (-e**(-2*n)*x**(-2*n)/(2*n), True))/e + A*d*Piecewise((log(x), Eq(n, 0)),
 (-x**n/(2*0**(1/n)*zoo**(1/n)*n*x**(2*n)*(0**(1/n))**(2*n) - n*x**(2*n)*(0
**(1/n))**(2*n)), Eq(e, 0**(1/n))), (-e**(-2*n)*x**(-n)/n, True))/e + B*c*P
iecewise((log(x), Eq(n, 0)), (-x**n/(2*0**(1/n)*zoo**(1/n)*n*x**(2*n)*(0**(
1/n))**(2*n) - n*x**(2*n)*(0**(1/n))**(2*n)), Eq(e, 0**(1/n))), (-e**(-2*n)
*x**(-n)/n, True))/e + B*d*Piecewise((e**(-2*n)*log(x), Abs(x) < 1), (-e**(
-2*n)*log(1/x), 1/Abs(x) < 1), (-e**(-2*n)*meijerg(((), (1, 1)), ((0, 0), (
)), x) + e**(-2*n)*meijerg(((1, 1), ()), ((), (0, 0)), x), True))/e, Eq(m,
-2*n - 1)), (A*c*Piecewise((log(x), Eq(n, 0)), (-x**(-n)*(0**(1/n))**(-n)/n
, Eq(e, 0**(1/n))), (-e**(-n)*x**(-n)/n, True))/e + A*d*Piecewise((e**(-n)*
log(x), Abs(x) < 1), (-e**(-n)*log(1/x), 1/Abs(x) < 1), (-e**(-n)*meijerg((
(), (1, 1)), ((0, 0), ()), x) + e**(-n)*meijerg(((1, 1), ()), ((), (0, 0)),
 x), True))/e + B*c*Piecewise((e**(-n)*log(x), Abs(x) < 1), (-e**(-n)*log(1
/x), 1/Abs(x) < 1), (-e**(-n)*meijerg(((), (1, 1)), ((0, 0), ()), x) + e**(
-n)*meijerg(((1, 1), ()), ((), (0, 0)), x), True))/e + B*d*Piecewise((log(x
), Eq(n, 0)), (-x**(2*n)/(0**(1/n)*zoo**(1/n)*n*x**n*(0**(1/n))**n - 2*n*x*
*n*(0**(1/n))**n), Eq(e, 0**(1/n))), (e**(-n)*x**n/n, True))/e, Eq(m, -n -
1)), (A*c*e**m*m**2*x*x**m/(m**3 + 3*m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n + 3
*m + 2*n**2 + 3*n + 1) + 3*A*c*e**m*m*n*x*x**m/(m**3 + 3*m**2*n + 3*m**2 +
2*m*n**2 + 6*m*n + 3*m + 2*n**2 + 3*n + 1) + 2*A*c*e**m*m*x*x**m/(m**3 + 3*
m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n + 3*m + 2*n**2 + 3*n + 1) + 2*A*c*e**m*n
**2*x*x**m/(m**3 + 3*m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n + 3*m + 2*n**2 + 3*
n + 1) + 3*A*c*e**m*n*x*x**m/(m**3 + 3*m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n +
 3*m + 2*n**2 + 3*n + 1) + A*c*e**m*x*x**m/(m**3 + 3*m**2*n + 3*m**2 + 2*m*
n**2 + 6*m*n + 3*m + 2*n**2 + 3*n + 1) + A*d*e**m*m**2*x*x**m*x**n/(m**3 +
```

```
3*m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n + 3*m + 2*n**2 + 3*n + 1) + 2*A*d*e**m
*m*n*x*x**m*x**n/(m**3 + 3*m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n + 3*m + 2*n**
2 + 3*n + 1) + 2*A*d*e**m*m*x*x**m*x**n/(m**3 + 3*m**2*n + 3*m**2 + 2*m*n**
2 + 6*m*n + 3*m + 2*n**2 + 3*n + 1) + 2*A*d*e**m*n*x*x**m*x**n/(m**3 + 3*m*
*2*n + 3*m**2 + 2*m*n**2 + 6*m*n + 3*m + 2*n**2 + 3*n + 1) + A*d*e**m*x*x**
m*x**n/(m**3 + 3*m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n + 3*m + 2*n**2 + 3*n +
1) + B*c*e**m*m**2*x*x**m*x**n/(m**3 + 3*m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n
 + 3*m + 2*n**2 + 3*n + 1) + 2*B*c*e**m*m*n*x*x**m*x**n/(m**3 + 3*m**2*n +
3*m**2 + 2*m*n**2 + 6*m*n + 3*m + 2*n**2 + 3*n + 1) + 2*B*c*e**m*m*x*x**m*x
**n/(m**3 + 3*m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n + 3*m + 2*n**2 + 3*n + 1)
+ 2*B*c*e**m*n*x*x**m*x**n/(m**3 + 3*m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n + 3
*m + 2*n**2 + 3*n + 1) + B*c*e**m*x*x**m*x**n/(m**3 + 3*m**2*n + 3*m**2 + 2
*m*n**2 + 6*m*n + 3*m + 2*n**2 + 3*n + 1) + B*d*e**m*m**2*x*x**m*x**(2*n)/(
m**3 + 3*m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n + 3*m + 2*n**2 + 3*n + 1) + B*d
*e**m*m*n*x*x**m*x**(2*n)/(m**3 + 3*m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n + 3*
m + 2*n**2 + 3*n + 1) + 2*B*d*e**m*m*x*x**m*x**(2*n)/(m**3 + 3*m**2*n + 3*m
**2 + 2*m*n**2 + 6*m*n + 3*m + 2*n**2 + 3*n + 1) + B*d*e**m*n*x*x**m*x**(2*
n)/(m**3 + 3*m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n + 3*m + 2*n**2 + 3*n + 1) +
 B*d*e**m*x*x**m*x**(2*n)/(m**3 + 3*m**2*n + 3*m**2 + 2*m*n**2 + 6*m*n + 3*
m + 2*n**2 + 3*n + 1), True))
```

3.5 $\int (ex)^m (a+bx^n)^3 (A+Bx^n)(c+dx^n)^2\,dx$

Optimal. Leaf size=318

$$\frac{a^3Ac^2(ex)^{m+1}}{e(m+1)}+\frac{ax^{2n+1}(ex)^m\left(A\left(a^2d^2+6abcd+3b^2c^2\right)+aBc(2ad+3bc)\right)}{m+2n+1}+\frac{x^{3n+1}(ex)^m\left(Ab\left(3a^2d^2+6abcd+\right.\right.}{m+}$$

Rubi [A] time = 0.41, antiderivative size = 318, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 3, integrand size = 31, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.097, Rules used = {570, 20, 30}

$$\frac{ax^{2n+1}(ex)^m\left(A\left(a^2d^2+6abcd+3b^2c^2\right)+aBc(2ad+3bc)\right)}{m+2n+1}+\frac{x^{3n+1}(ex)^m\left(Ab\left(3a^2d^2+6abcd+b^2c^2\right)+aB\left(a^2d^2+6abcd+3b^2c^2\right)\right)}{m+3n+1}+\frac{bx^{4n+1}(ex)^m\left(3a^2Bd^2+3abd(Ad+2Bc)+b^2c(2Ad+Bc)\right)}{m+4n+1}+\frac{a^2cx^{n+1}(ex)^m(2aAd+aBc+3Abc)}{m+n+1}+\frac{a^3Ac^2(ex)^{m+1}}{e(m+1)}+\frac{b^2dx^{5n+1}(ex)^m(3aBd+Abd+2bBc)}{m+5n+1}+\frac{b^3Bd^2x^{6n+1}(ex)^m}{m+6n+1}$$

Antiderivative was successfully verified.

```
[In] Int[(e*x)^m*(a + b*x^n)^3*(A + B*x^n)*(c + d*x^n)^2,x]
```

```
[Out] (a^2*c*(3*A*b*c + a*B*c + 2*a*A*d)*x^(1 + n)*(e*x)^m)/(1 + m + n) + (a*(a*B
*c*(3*b*c + 2*a*d) + A*(3*b^2*c^2 + 6*a*b*c*d + a^2*d^2))*x^(1 + 2*n)*(e*x)
^m)/(1 + m + 2*n) + ((a*B*(3*b^2*c^2 + 6*a*b*c*d + a^2*d^2) + A*b*(b^2*c^2
+ 6*a*b*c*d + 3*a^2*d^2))*x^(1 + 3*n)*(e*x)^m)/(1 + m + 3*n) + (b*(3*a^2*B*
d^2 + 3*a*b*d*(2*B*c + A*d) + b^2*c*(B*c + 2*A*d))*x^(1 + 4*n)*(e*x)^m)/(1
+ m + 4*n) + (b^2*d*(2*b*B*c + A*b*d + 3*a*B*d)*x^(1 + 5*n)*(e*x)^m)/(1 + m
 + 5*n) + (b^3*B*d^2*x^(1 + 6*n)*(e*x)^m)/(1 + m + 6*n) + (a^3*A*c^2*(e*x)^
(1 + m))/(e*(1 + m))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 30

```
Int[(x_)^(m_.), x_Symbol] :> Simp[x^(m + 1)/(m + 1), x] /; FreeQ[m, x] && N
eQ[m, -1]
```

Rule 570

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

Rubi steps

$$\begin{aligned}
\int (ex)^m (a+bx^n)^3 (A+Bx^n)(c+dx^n)^2\,dx &= \int \left(a^3Ac^2(ex)^m+a^2c(3Abc+aBc+2aAd)x^n(ex)^m+a\left(aBc(3bc\right.\right.\\
&= \frac{a^3Ac^2(ex)^{1+m}}{e(1+m)}+\left(b^3Bd^2\right)\int x^{6n}(ex)^m\,dx+\left(a^2c(3Abc+aBc+2a\right.\\
&= \frac{a^3Ac^2(ex)^{1+m}}{e(1+m)}+\left(b^3Bd^2x^{-m}(ex)^m\right)\int x^{m+6n}\,dx+\left(a^2c(3Abc+a\right.\\
&= \frac{a^2c(3Abc+aBc+2aAd)x^{1+n}(ex)^m}{1+m+n}+\frac{a\left(aBc(3bc+2ad)+A\left(3\right.\right.}{1}
\end{aligned}$$

Mathematica [A] time = 1.48, size = 273, normalized size = 0.86

$$x(ex)^m\left(\frac{a^3Ac^2}{m+1}+\frac{ax^{2n}\left(A\left(a^2d^2+6abcd+3b^2c^2\right)+aBc(2ad+3bc)\right)}{m+2n+1}+\frac{x^{3n}\left(Ab\left(3a^2d^2+6abcd+b^2c^2\right)+aB\left(a^2d^2+6abcd+3b^2c^2\right)\right)}{m+3n+1}+\frac{bx^{4n}\left(3a^2Bd^2+3abd(Ad+2Bc)+b^2c(2Ad+Bc)\right)}{m+4n+1}+\frac{a^2cx^n(2aAd+aBc+3Abc)}{m+n+1}+\frac{b^2dx^{5n}(3aBd+Abd+2bBc)}{m+5n+1}+\frac{b^3Bd^2x^{6n}}{m+6n+1}\right)$$

Antiderivative was successfully verified.

```
[In] Integrate[(e*x)^m*(a + b*x^n)^3*(A + B*x^n)*(c + d*x^n)^2,x]
```

```
[Out] x*(e*x)^m*((a^3*A*c^2)/(1 + m) + (a^2*c*(3*A*b*c + a*B*c + 2*a*A*d)*x^n)/(1
 + m + n) + (a*(a*B*c*(3*b*c + 2*a*d) + A*(3*b^2*c^2 + 6*a*b*c*d + a^2*d^2)
)*x^(2*n))/(1 + m + 2*n) + ((a*B*(3*b^2*c^2 + 6*a*b*c*d + a^2*d^2) + A*b*(b
^2*c^2 + 6*a*b*c*d + 3*a^2*d^2))*x^(3*n))/(1 + m + 3*n) + (b*(3*a^2*B*d^2 +
 3*a*b*d*(2*B*c + A*d) + b^2*c*(B*c + 2*A*d))*x^(4*n))/(1 + m + 4*n) + (b^2
*d*(2*b*B*c + A*b*d + 3*a*B*d)*x^(5*n))/(1 + m + 5*n) + (b^3*B*d^2*x^(6*n))
/(1 + m + 6*n))
```

IntegrateAlgebraic [F] time = 0.86, size = 0, normalized size = 0.00

$$\int (ex)^m\,(a+bx^n)^3\,(A+Bx^n)\,(c+dx^n)^2\;dx$$

Verification is not applicable to the result.

```
[In] IntegrateAlgebraic[(e*x)^m*(a + b*x^n)^3*(A + B*x^n)*(c + d*x^n)^2,x]
```

```
[Out] Defer[IntegrateAlgebraic][(e*x)^m*(a + b*x^n)^3*(A + B*x^n)*(c + d*x^n)^2,
x]
```

fricas [B] time = 0.58, size = 6638, normalized size = 20.87

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

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

```
[Out] ((B*b^3*d^2*m^6 + 6*B*b^3*d^2*m^5 + 15*B*b^3*d^2*m^4 + 20*B*b^3*d^2*m^3 + 1
5*B*b^3*d^2*m^2 + 6*B*b^3*d^2*m + B*b^3*d^2 + 120*(B*b^3*d^2*m + B*b^3*d^2)
*n^5 + 274*(B*b^3*d^2*m^2 + 2*B*b^3*d^2*m + B*b^3*d^2)*n^4 + 225*(B*b^3*d^2
*m^3 + 3*B*b^3*d^2*m^2 + 3*B*b^3*d^2*m + B*b^3*d^2)*n^3 + 85*(B*b^3*d^2*m^4
 + 4*B*b^3*d^2*m^3 + 6*B*b^3*d^2*m^2 + 4*B*b^3*d^2*m + B*b^3*d^2)*n^2 + 15*
(B*b^3*d^2*m^5 + 5*B*b^3*d^2*m^4 + 10*B*b^3*d^2*m^3 + 10*B*b^3*d^2*m^2 + 5*
B*b^3*d^2*m + B*b^3*d^2)*n)*x*x^(6*n)*e^(m*log(e) + m*log(x)) + ((2*B*b^3*c
*d + (3*B*a*b^2 + A*b^3)*d^2)*m^6 + 2*B*b^3*c*d + 6*(2*B*b^3*c*d + (3*B*a*b
^2 + A*b^3)*d^2)*m^5 + 144*(2*B*b^3*c*d + (3*B*a*b^2 + A*b^3)*d^2 + (2*B*b^
3*c*d + (3*B*a*b^2 + A*b^3)*d^2)*m)*n^5 + 15*(2*B*b^3*c*d + (3*B*a*b^2 + A*
b^3)*d^2)*m^4 + 324*(2*B*b^3*c*d + (3*B*a*b^2 + A*b^3)*d^2 + (2*B*b^3*c*d +
 (3*B*a*b^2 + A*b^3)*d^2)*m^2 + 2*(2*B*b^3*c*d + (3*B*a*b^2 + A*b^3)*d^2)*m
)*n^4 + 20*(2*B*b^3*c*d + (3*B*a*b^2 + A*b^3)*d^2)*m^3 + 260*(2*B*b^3*c*d +
 (2*B*b^3*c*d + (3*B*a*b^2 + A*b^3)*d^2)*m^3 + (3*B*a*b^2 + A*b^3)*d^2 + 3*
(2*B*b^3*c*d + (3*B*a*b^2 + A*b^3)*d^2)*m^2 + 3*(2*B*b^3*c*d + (3*B*a*b^2 +
 A*b^3)*d^2)*m)*n^3 + (3*B*a*b^2 + A*b^3)*d^2 + 15*(2*B*b^3*c*d + (3*B*a*b^
2 + A*b^3)*d^2)*m^2 + 95*(2*B*b^3*c*d + (2*B*b^3*c*d + (3*B*a*b^2 + A*b^3)*
d^2)*m^4 + 4*(2*B*b^3*c*d + (3*B*a*b^2 + A*b^3)*d^2)*m^3 + (3*B*a*b^2 + A*b
^3)*d^2 + 6*(2*B*b^3*c*d + (3*B*a*b^2 + A*b^3)*d^2)*m^2 + 4*(2*B*b^3*c*d +
(3*B*a*b^2 + A*b^3)*d^2)*m)*n^2 + 6*(2*B*b^3*c*d + (3*B*a*b^2 + A*b^3)*d^2)
*m + 16*(2*B*b^3*c*d + (2*B*b^3*c*d + (3*B*a*b^2 + A*b^3)*d^2)*m^5 + 5*(2*B
*b^3*c*d + (3*B*a*b^2 + A*b^3)*d^2)*m^4 + 10*(2*B*b^3*c*d + (3*B*a*b^2 + A*
b^3)*d^2)*m^3 + (3*B*a*b^2 + A*b^3)*d^2 + 10*(2*B*b^3*c*d + (3*B*a*b^2 + A*
b^3)*d^2)*m^2 + 5*(2*B*b^3*c*d + (3*B*a*b^2 + A*b^3)*d^2)*m)*n)*x*x^(5*n)*e
^(m*log(e) + m*log(x)) + ((B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3)*c*d + 3*(B*a^2
*b + A*a*b^2)*d^2)*m^6 + B*b^3*c^2 + 6*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3)*c
*d + 3*(B*a^2*b + A*a*b^2)*d^2)*m^5 + 180*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3
```

```
)*c*d + 3*(B*a^2*b + A*a*b^2)*d^2 + (B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3)*c*d
+ 3*(B*a^2*b + A*a*b^2)*d^2)*m)*n^5 + 15*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3)
*c*d + 3*(B*a^2*b + A*a*b^2)*d^2)*m^4 + 396*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b
^3)*c*d + 3*(B*a^2*b + A*a*b^2)*d^2 + (B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3)*c*
d + 3*(B*a^2*b + A*a*b^2)*d^2)*m^2 + 2*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3)*c
*d + 3*(B*a^2*b + A*a*b^2)*d^2)*m)*n^4 + 20*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b
^3)*c*d + 3*(B*a^2*b + A*a*b^2)*d^2)*m^3 + 307*(B*b^3*c^2 + (B*b^3*c^2 + 2*
(3*B*a*b^2 + A*b^3)*c*d + 3*(B*a^2*b + A*a*b^2)*d^2)*m^3 + 2*(3*B*a*b^2 + A
*b^3)*c*d + 3*(B*a^2*b + A*a*b^2)*d^2 + 3*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3
)*c*d + 3*(B*a^2*b + A*a*b^2)*d^2)*m^2 + 3*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^
3)*c*d + 3*(B*a^2*b + A*a*b^2)*d^2)*m)*n^3 + 2*(3*B*a*b^2 + A*b^3)*c*d + 3*
(B*a^2*b + A*a*b^2)*d^2 + 15*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3)*c*d + 3*(B*
a^2*b + A*a*b^2)*d^2)*m^2 + 107*(B*b^3*c^2 + (B*b^3*c^2 + 2*(3*B*a*b^2 + A*
b^3)*c*d + 3*(B*a^2*b + A*a*b^2)*d^2)*m^4 + 4*(B*b^3*c^2 + 2*(3*B*a*b^2 + A
*b^3)*c*d + 3*(B*a^2*b + A*a*b^2)*d^2)*m^3 + 2*(3*B*a*b^2 + A*b^3)*c*d + 3*
(B*a^2*b + A*a*b^2)*d^2 + 6*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3)*c*d + 3*(B*a
^2*b + A*a*b^2)*d^2)*m^2 + 4*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3)*c*d + 3*(B*
a^2*b + A*a*b^2)*d^2)*m)*n^2 + 6*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3)*c*d + 3
*(B*a^2*b + A*a*b^2)*d^2)*m + 17*(B*b^3*c^2 + (B*b^3*c^2 + 2*(3*B*a*b^2 + A
*b^3)*c*d + 3*(B*a^2*b + A*a*b^2)*d^2)*m^5 + 5*(B*b^3*c^2 + 2*(3*B*a*b^2 +
A*b^3)*c*d + 3*(B*a^2*b + A*a*b^2)*d^2)*m^4 + 10*(B*b^3*c^2 + 2*(3*B*a*b^2
+ A*b^3)*c*d + 3*(B*a^2*b + A*a*b^2)*d^2)*m^3 + 2*(3*B*a*b^2 + A*b^3)*c*d +
 3*(B*a^2*b + A*a*b^2)*d^2 + 10*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3)*c*d + 3*
(B*a^2*b + A*a*b^2)*d^2)*m^2 + 5*(B*b^3*c^2 + 2*(3*B*a*b^2 + A*b^3)*c*d + 3
*(B*a^2*b + A*a*b^2)*d^2)*m)*n)*x*x^(4*n)*e^(m*log(e) + m*log(x)) + (((3*B*
a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2)*m
^6 + 6*((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*
a^2*b)*d^2)*m^5 + 240*((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d
+ (B*a^3 + 3*A*a^2*b)*d^2 + ((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2
)*c*d + (B*a^3 + 3*A*a^2*b)*d^2)*m)*n^5 + 15*((3*B*a*b^2 + A*b^3)*c^2 + 6*(
B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2)*m^4 + 508*((3*B*a*b^2 + A
*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2 + ((3*B*a*b
^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2)*m^2
+ 2*((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2
*b)*d^2)*m)*n^4 + 20*((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d +
 (B*a^3 + 3*A*a^2*b)*d^2)*m^3 + 372*(((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b
+ A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2)*m^3 + (3*B*a*b^2 + A*b^3)*c^2 + 6
*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2 + 3*((3*B*a*b^2 + A*b^3)
*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2)*m^2 + 3*((3*B*a
*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2)*m)
*n^3 + (3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a
^2*b)*d^2 + 15*((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^
3 + 3*A*a^2*b)*d^2)*m^2 + 121*(((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*
b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2)*m^4 + 4*((3*B*a*b^2 + A*b^3)*c^2 + 6*(B
*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2)*m^3 + (3*B*a*b^2 + A*b^3)*
c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2 + 6*((3*B*a*b^2 +
 A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2)*m^2 + 4*
((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*
d^2)*m)*n^2 + 6*((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a
^3 + 3*A*a^2*b)*d^2)*m + 18*(((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^
2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2)*m^5 + 5*((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a
^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2)*m^4 + 10*((3*B*a*b^2 + A*b^3
)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2)*m^3 + (3*B*a*b
^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*d^2 + 10*
((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3 + 3*A*a^2*b)*
d^2)*m^2 + 5*((3*B*a*b^2 + A*b^3)*c^2 + 6*(B*a^2*b + A*a*b^2)*c*d + (B*a^3
+ 3*A*a^2*b)*d^2)*m)*n)*x*x^(3*n)*e^(m*log(e) + m*log(x)) + ((A*a^3*d^2 + 3
*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m^6 + A*a^3*d^2 + 6*(
A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m^5 + 36
```

```
0*(A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d + (A*a
^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m)*n^5 + 15
*(A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m^4 +
702*(A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d + (A
*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m^2 + 2*(
A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m)*n^4 +
 20*(A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m^3
 + 461*(A*a^3*d^2 + (A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A
*a^2*b)*c*d)*m^3 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d +
3*(A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m^2 +
 3*(A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m)*n
^3 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d + 15*(A*a^3*d^2
+ 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m^2 + 137*(A*a^3*d
^2 + (A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m^
4 + 4*(A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m
^3 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d + 6*(A*a^3*d^2 +
 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m^2 + 4*(A*a^3*d^2
+ 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m)*n^2 + 6*(A*a^3*
d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m + 19*(A*a^3*
d^2 + (A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m
^5 + 5*(A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*
m^4 + 10*(A*a^3*d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d
)*m^3 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d + 10*(A*a^3*d
^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m^2 + 5*(A*a^3*
d^2 + 3*(B*a^2*b + A*a*b^2)*c^2 + 2*(B*a^3 + 3*A*a^2*b)*c*d)*m)*n)*x*x^(2*n
)*e^(m*log(e) + m*log(x)) + ((2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2)*m^6 +
2*A*a^3*c*d + 6*(2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2)*m^5 + 720*(2*A*a^3*
c*d + (B*a^3 + 3*A*a^2*b)*c^2 + (2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2)*m)*
n^5 + 15*(2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2)*m^4 + 1044*(2*A*a^3*c*d +
(B*a^3 + 3*A*a^2*b)*c^2 + (2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2)*m^2 + 2*(
2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2)*m)*n^4 + 20*(2*A*a^3*c*d + (B*a^3 +
3*A*a^2*b)*c^2)*m^3 + 580*(2*A*a^3*c*d + (2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)
*c^2)*m^3 + (B*a^3 + 3*A*a^2*b)*c^2 + 3*(2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*
c^2)*m^2 + 3*(2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2)*m)*n^3 + (B*a^3 + 3*A*
a^2*b)*c^2 + 15*(2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2)*m^2 + 155*(2*A*a^3*
c*d + (2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2)*m^4 + 4*(2*A*a^3*c*d + (B*a^3
 + 3*A*a^2*b)*c^2)*m^3 + (B*a^3 + 3*A*a^2*b)*c^2 + 6*(2*A*a^3*c*d + (B*a^3
+ 3*A*a^2*b)*c^2)*m^2 + 4*(2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2)*m)*n^2 +
6*(2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2)*m + 20*(2*A*a^3*c*d + (2*A*a^3*c*
d + (B*a^3 + 3*A*a^2*b)*c^2)*m^5 + 5*(2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2
)*m^4 + 10*(2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2)*m^3 + (B*a^3 + 3*A*a^2*b
)*c^2 + 10*(2*A*a^3*c*d + (B*a^3 + 3*A*a^2*b)*c^2)*m^2 + 5*(2*A*a^3*c*d + (
B*a^3 + 3*A*a^2*b)*c^2)*m)*n)*x*x^n*e^(m*log(e) + m*log(x)) + (A*a^3*c^2*m^
6 + 720*A*a^3*c^2*n^6 + 6*A*a^3*c^2*m^5 + 15*A*a^3*c^2*m^4 + 20*A*a^3*c^2*m
^3 + 15*A*a^3*c^2*m^2 + 6*A*a^3*c^2*m + A*a^3*c^2 + 1764*(A*a^3*c^2*m + A*a
^3*c^2)*n^5 + 1624*(A*a^3*c^2*m^2 + 2*A*a^3*c^2*m + A*a^3*c^2)*n^4 + 735*(A
*a^3*c^2*m^3 + 3*A*a^3*c^2*m^2 + 3*A*a^3*c^2*m + A*a^3*c^2)*n^3 + 175*(A*a^
3*c^2*m^4 + 4*A*a^3*c^2*m^3 + 6*A*a^3*c^2*m^2 + 4*A*a^3*c^2*m + A*a^3*c^2)*
n^2 + 21*(A*a^3*c^2*m^5 + 5*A*a^3*c^2*m^4 + 10*A*a^3*c^2*m^3 + 10*A*a^3*c^2
*m^2 + 5*A*a^3*c^2*m + A*a^3*c^2)*n)*x*e^(m*log(e) + m*log(x)))/(m^7 + 720*
(m + 1)*n^6 + 7*m^6 + 1764*(m^2 + 2*m + 1)*n^5 + 21*m^5 + 1624*(m^3 + 3*m^2
 + 3*m + 1)*n^4 + 35*m^4 + 735*(m^4 + 4*m^3 + 6*m^2 + 4*m + 1)*n^3 + 35*m^3
 + 175*(m^5 + 5*m^4 + 10*m^3 + 10*m^2 + 5*m + 1)*n^2 + 21*m^2 + 21*(m^6 + 6
*m^5 + 15*m^4 + 20*m^3 + 15*m^2 + 6*m + 1)*n + 7*m + 1)
```

giac [B] time = 2.01, size = 15358, normalized size = 48.30

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

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

```
[Out] (B*b^3*d^2*m^6*x*x^m*x^(6*n)*e^m + 15*B*b^3*d^2*m^5*n*x*x^m*x^(6*n)*e^m + 8
5*B*b^3*d^2*m^4*n^2*x*x^m*x^(6*n)*e^m + 225*B*b^3*d^2*m^3*n^3*x*x^m*x^(6*n)
*e^m + 274*B*b^3*d^2*m^2*n^4*x*x^m*x^(6*n)*e^m + 120*B*b^3*d^2*m*n^5*x*x^m*
x^(6*n)*e^m + 2*B*b^3*c*d*m^6*x*x^m*x^(5*n)*e^m + 3*B*a*b^2*d^2*m^6*x*x^m*x
^(5*n)*e^m + A*b^3*d^2*m^6*x*x^m*x^(5*n)*e^m + 32*B*b^3*c*d*m^5*n*x*x^m*x^(
5*n)*e^m + 48*B*a*b^2*d^2*m^5*n*x*x^m*x^(5*n)*e^m + 16*A*b^3*d^2*m^5*n*x*x^
m*x^(5*n)*e^m + 190*B*b^3*c*d*m^4*n^2*x*x^m*x^(5*n)*e^m + 285*B*a*b^2*d^2*m
^4*n^2*x*x^m*x^(5*n)*e^m + 95*A*b^3*d^2*m^4*n^2*x*x^m*x^(5*n)*e^m + 520*B*b
^3*c*d*m^3*n^3*x*x^m*x^(5*n)*e^m + 780*B*a*b^2*d^2*m^3*n^3*x*x^m*x^(5*n)*e^
m + 260*A*b^3*d^2*m^3*n^3*x*x^m*x^(5*n)*e^m + 648*B*b^3*c*d*m^2*n^4*x*x^m*x
^(5*n)*e^m + 972*B*a*b^2*d^2*m^2*n^4*x*x^m*x^(5*n)*e^m + 324*A*b^3*d^2*m^2*
n^4*x*x^m*x^(5*n)*e^m + 288*B*b^3*c*d*m*n^5*x*x^m*x^(5*n)*e^m + 432*B*a*b^2
*d^2*m*n^5*x*x^m*x^(5*n)*e^m + 144*A*b^3*d^2*m*n^5*x*x^m*x^(5*n)*e^m + B*b^
3*c^2*m^6*x*x^m*x^(4*n)*e^m + 6*B*a*b^2*c*d*m^6*x*x^m*x^(4*n)*e^m + 2*A*b^3
*c*d*m^6*x*x^m*x^(4*n)*e^m + 3*B*a^2*b*d^2*m^6*x*x^m*x^(4*n)*e^m + 3*A*a*b^
2*d^2*m^6*x*x^m*x^(4*n)*e^m + 17*B*b^3*c^2*m^5*n*x*x^m*x^(4*n)*e^m + 102*B*
a*b^2*c*d*m^5*n*x*x^m*x^(4*n)*e^m + 34*A*b^3*c*d*m^5*n*x*x^m*x^(4*n)*e^m +
51*B*a^2*b*d^2*m^5*n*x*x^m*x^(4*n)*e^m + 51*A*a*b^2*d^2*m^5*n*x*x^m*x^(4*n)
*e^m + 107*B*b^3*c^2*m^4*n^2*x*x^m*x^(4*n)*e^m + 642*B*a*b^2*c*d*m^4*n^2*x*
x^m*x^(4*n)*e^m + 214*A*b^3*c*d*m^4*n^2*x*x^m*x^(4*n)*e^m + 321*B*a^2*b*d^2
*m^4*n^2*x*x^m*x^(4*n)*e^m + 321*A*a*b^2*d^2*m^4*n^2*x*x^m*x^(4*n)*e^m + 30
7*B*b^3*c^2*m^3*n^3*x*x^m*x^(4*n)*e^m + 1842*B*a*b^2*c*d*m^3*n^3*x*x^m*x^(4
*n)*e^m + 614*A*b^3*c*d*m^3*n^3*x*x^m*x^(4*n)*e^m + 921*B*a^2*b*d^2*m^3*n^3
*x*x^m*x^(4*n)*e^m + 921*A*a*b^2*d^2*m^3*n^3*x*x^m*x^(4*n)*e^m + 396*B*b^3*
c^2*m^2*n^4*x*x^m*x^(4*n)*e^m + 2376*B*a*b^2*c*d*m^2*n^4*x*x^m*x^(4*n)*e^m
+ 792*A*b^3*c*d*m^2*n^4*x*x^m*x^(4*n)*e^m + 1188*B*a^2*b*d^2*m^2*n^4*x*x^m*
x^(4*n)*e^m + 1188*A*a*b^2*d^2*m^2*n^4*x*x^m*x^(4*n)*e^m + 180*B*b^3*c^2*m*
n^5*x*x^m*x^(4*n)*e^m + 1080*B*a*b^2*c*d*m*n^5*x*x^m*x^(4*n)*e^m + 360*A*b^
3*c*d*m*n^5*x*x^m*x^(4*n)*e^m + 540*B*a^2*b*d^2*m*n^5*x*x^m*x^(4*n)*e^m + 5
40*A*a*b^2*d^2*m*n^5*x*x^m*x^(4*n)*e^m + 3*B*a*b^2*c^2*m^6*x*x^m*x^(3*n)*e^
m + A*b^3*c^2*m^6*x*x^m*x^(3*n)*e^m + 6*B*a^2*b*c*d*m^6*x*x^m*x^(3*n)*e^m +
 6*A*a*b^2*c*d*m^6*x*x^m*x^(3*n)*e^m + B*a^3*d^2*m^6*x*x^m*x^(3*n)*e^m + 3*
A*a^2*b*d^2*m^6*x*x^m*x^(3*n)*e^m + 54*B*a*b^2*c^2*m^5*n*x*x^m*x^(3*n)*e^m
+ 18*A*b^3*c^2*m^5*n*x*x^m*x^(3*n)*e^m + 108*B*a^2*b*c*d*m^5*n*x*x^m*x^(3*n
)*e^m + 108*A*a*b^2*c*d*m^5*n*x*x^m*x^(3*n)*e^m + 18*B*a^3*d^2*m^5*n*x*x^m*
x^(3*n)*e^m + 54*A*a^2*b*d^2*m^5*n*x*x^m*x^(3*n)*e^m + 363*B*a*b^2*c^2*m^4*
n^2*x*x^m*x^(3*n)*e^m + 121*A*b^3*c^2*m^4*n^2*x*x^m*x^(3*n)*e^m + 726*B*a^2
*b*c*d*m^4*n^2*x*x^m*x^(3*n)*e^m + 726*A*a*b^2*c*d*m^4*n^2*x*x^m*x^(3*n)*e^
m + 121*B*a^3*d^2*m^4*n^2*x*x^m*x^(3*n)*e^m + 363*A*a^2*b*d^2*m^4*n^2*x*x^m
*x^(3*n)*e^m + 1116*B*a*b^2*c^2*m^3*n^3*x*x^m*x^(3*n)*e^m + 372*A*b^3*c^2*m
^3*n^3*x*x^m*x^(3*n)*e^m + 2232*B*a^2*b*c*d*m^3*n^3*x*x^m*x^(3*n)*e^m + 223
2*A*a*b^2*c*d*m^3*n^3*x*x^m*x^(3*n)*e^m + 372*B*a^3*d^2*m^3*n^3*x*x^m*x^(3*
n)*e^m + 1116*A*a^2*b*d^2*m^3*n^3*x*x^m*x^(3*n)*e^m + 1524*B*a*b^2*c^2*m^2*
n^4*x*x^m*x^(3*n)*e^m + 508*A*b^3*c^2*m^2*n^4*x*x^m*x^(3*n)*e^m + 3048*B*a^
2*b*c*d*m^2*n^4*x*x^m*x^(3*n)*e^m + 3048*A*a*b^2*c*d*m^2*n^4*x*x^m*x^(3*n)*
e^m + 508*B*a^3*d^2*m^2*n^4*x*x^m*x^(3*n)*e^m + 1524*A*a^2*b*d^2*m^2*n^4*x*
x^m*x^(3*n)*e^m + 720*B*a*b^2*c^2*m*n^5*x*x^m*x^(3*n)*e^m + 240*A*b^3*c^2*m
*n^5*x*x^m*x^(3*n)*e^m + 1440*B*a^2*b*c*d*m*n^5*x*x^m*x^(3*n)*e^m + 1440*A*
a*b^2*c*d*m*n^5*x*x^m*x^(3*n)*e^m + 240*B*a^3*d^2*m*n^5*x*x^m*x^(3*n)*e^m +
 720*A*a^2*b*d^2*m*n^5*x*x^m*x^(3*n)*e^m + 3*B*a^2*b*c^2*m^6*x*x^m*x^(2*n)*
e^m + 3*A*a*b^2*c^2*m^6*x*x^m*x^(2*n)*e^m + 2*B*a^3*c*d*m^6*x*x^m*x^(2*n)*e
^m + 6*A*a^2*b*c*d*m^6*x*x^m*x^(2*n)*e^m + A*a^3*d^2*m^6*x*x^m*x^(2*n)*e^m
+ 57*B*a^2*b*c^2*m^5*n*x*x^m*x^(2*n)*e^m + 57*A*a*b^2*c^2*m^5*n*x*x^m*x^(2*
n)*e^m + 38*B*a^3*c*d*m^5*n*x*x^m*x^(2*n)*e^m + 114*A*a^2*b*c*d*m^5*n*x*x^m
*x^(2*n)*e^m + 19*A*a^3*d^2*m^5*n*x*x^m*x^(2*n)*e^m + 411*B*a^2*b*c^2*m^4*n
^2*x*x^m*x^(2*n)*e^m + 411*A*a*b^2*c^2*m^4*n^2*x*x^m*x^(2*n)*e^m + 274*B*a^
3*c*d*m^4*n^2*x*x^m*x^(2*n)*e^m + 822*A*a^2*b*c*d*m^4*n^2*x*x^m*x^(2*n)*e^m
 + 137*A*a^3*d^2*m^4*n^2*x*x^m*x^(2*n)*e^m + 1383*B*a^2*b*c^2*m^3*n^3*x*x^m
```

```
*x^(2*n)*e^m + 1383*A*a*b^2*c^2*m^3*n^3*x*x^m*x^(2*n)*e^m + 922*B*a^3*c*d*m
^3*n^3*x*x^m*x^(2*n)*e^m + 2766*A*a^2*b*c*d*m^3*n^3*x*x^m*x^(2*n)*e^m + 461
*A*a^3*d^2*m^3*n^3*x*x^m*x^(2*n)*e^m + 2106*B*a^2*b*c^2*m^2*n^4*x*x^m*x^(2*
n)*e^m + 2106*A*a*b^2*c^2*m^2*n^4*x*x^m*x^(2*n)*e^m + 1404*B*a^3*c*d*m^2*n^
4*x*x^m*x^(2*n)*e^m + 4212*A*a^2*b*c*d*m^2*n^4*x*x^m*x^(2*n)*e^m + 702*A*a^
3*d^2*m^2*n^4*x*x^m*x^(2*n)*e^m + 1080*B*a^2*b*c^2*m*n^5*x*x^m*x^(2*n)*e^m
 + 1080*A*a*b^2*c^2*m*n^5*x*x^m*x^(2*n)*e^m + 720*B*a^3*c*d*m*n^5*x*x^m*x^(2
*n)*e^m + 2160*A*a^2*b*c*d*m*n^5*x*x^m*x^(2*n)*e^m + 360*A*a^3*d^2*m*n^5*x*
x^m*x^(2*n)*e^m + B*a^3*c^2*m^6*x*x^m*x^n*e^m + 3*A*a^2*b*c^2*m^6*x*x^m*x^n
*e^m + 2*A*a^3*c*d*m^6*x*x^m*x^n*e^m + 20*B*a^3*c^2*m^5*n*x*x^m*x^n*e^m + 6
0*A*a^2*b*c^2*m^5*n*x*x^m*x^n*e^m + 40*A*a^3*c*d*m^5*n*x*x^m*x^n*e^m + 155*
B*a^3*c^2*m^4*n^2*x*x^m*x^n*e^m + 465*A*a^2*b*c^2*m^4*n^2*x*x^m*x^n*e^m + 3
10*A*a^3*c*d*m^4*n^2*x*x^m*x^n*e^m + 580*B*a^3*c^2*m^3*n^3*x*x^m*x^n*e^m +
1740*A*a^2*b*c^2*m^3*n^3*x*x^m*x^n*e^m + 1160*A*a^3*c*d*m^3*n^3*x*x^m*x^n*e
^m + 1044*B*a^3*c^2*m^2*n^4*x*x^m*x^n*e^m + 3132*A*a^2*b*c^2*m^2*n^4*x*x^m*
x^n*e^m + 2088*A*a^3*c*d*m^2*n^4*x*x^m*x^n*e^m + 720*B*a^3*c^2*m*n^5*x*x^m*
x^n*e^m + 2160*A*a^2*b*c^2*m*n^5*x*x^m*x^n*e^m + 1440*A*a^3*c*d*m*n^5*x*x^m
*x^n*e^m + A*a^3*c^2*m^6*x*x^m*e^m + 21*A*a^3*c^2*m^5*n*x*x^m*e^m + 175*A*a
^3*c^2*m^4*n^2*x*x^m*e^m + 735*A*a^3*c^2*m^3*n^3*x*x^m*e^m + 1624*A*a^3*c^2
*m^2*n^4*x*x^m*e^m + 1764*A*a^3*c^2*m*n^5*x*x^m*e^m + 720*A*a^3*c^2*n^6*x*x
^m*e^m + 6*B*b^3*d^2*m^5*x*x^m*x^(6*n)*e^m + 75*B*b^3*d^2*m^4*n*x*x^m*x^(6*
n)*e^m + 340*B*b^3*d^2*m^3*n^2*x*x^m*x^(6*n)*e^m + 675*B*b^3*d^2*m^2*n^3*x*
x^m*x^(6*n)*e^m + 548*B*b^3*d^2*m*n^4*x*x^m*x^(6*n)*e^m + 120*B*b^3*d^2*n^5
*x*x^m*x^(6*n)*e^m + 12*B*b^3*c*d*m^5*x*x^m*x^(5*n)*e^m + 18*B*a*b^2*d^2*m^
5*x*x^m*x^(5*n)*e^m + 6*A*b^3*d^2*m^5*x*x^m*x^(5*n)*e^m + 160*B*b^3*c*d*m^4
*n*x*x^m*x^(5*n)*e^m + 240*B*a*b^2*d^2*m^4*n*x*x^m*x^(5*n)*e^m + 80*A*b^3*d
^2*m^4*n*x*x^m*x^(5*n)*e^m + 760*B*b^3*c*d*m^3*n^2*x*x^m*x^(5*n)*e^m + 1140
*B*a*b^2*d^2*m^3*n^2*x*x^m*x^(5*n)*e^m + 380*A*b^3*d^2*m^3*n^2*x*x^m*x^(5*n
)*e^m + 1560*B*b^3*c*d*m^2*n^3*x*x^m*x^(5*n)*e^m + 2340*B*a*b^2*d^2*m^2*n^3
*x*x^m*x^(5*n)*e^m + 780*A*b^3*d^2*m^2*n^3*x*x^m*x^(5*n)*e^m + 1296*B*b^3*c
*d*m*n^4*x*x^m*x^(5*n)*e^m + 1944*B*a*b^2*d^2*m*n^4*x*x^m*x^(5*n)*e^m + 648
*A*b^3*d^2*m*n^4*x*x^m*x^(5*n)*e^m + 288*B*b^3*c*d*n^5*x*x^m*x^(5*n)*e^m +
432*B*a*b^2*d^2*n^5*x*x^m*x^(5*n)*e^m + 144*A*b^3*d^2*n^5*x*x^m*x^(5*n)*e^m
 + 6*B*b^3*c^2*m^5*x*x^m*x^(4*n)*e^m + 36*B*a*b^2*c*d*m^5*x*x^m*x^(4*n)*e^m
 + 12*A*b^3*c*d*m^5*x*x^m*x^(4*n)*e^m + 18*B*a^2*b*d^2*m^5*x*x^m*x^(4*n)*e^
m + 18*A*a*b^2*d^2*m^5*x*x^m*x^(4*n)*e^m + 85*B*b^3*c^2*m^4*n*x*x^m*x^(4*n)
*e^m + 510*B*a*b^2*c*d*m^4*n*x*x^m*x^(4*n)*e^m + 170*A*b^3*c*d*m^4*n*x*x^m*
x^(4*n)*e^m + 255*B*a^2*b*d^2*m^4*n*x*x^m*x^(4*n)*e^m + 255*A*a*b^2*d^2*m^4
*n*x*x^m*x^(4*n)*e^m + 428*B*b^3*c^2*m^3*n^2*x*x^m*x^(4*n)*e^m + 2568*B*a*b
^2*c*d*m^3*n^2*x*x^m*x^(4*n)*e^m + 856*A*b^3*c*d*m^3*n^2*x*x^m*x^(4*n)*e^m
 + 1284*B*a^2*b*d^2*m^3*n^2*x*x^m*x^(4*n)*e^m + 1284*A*a*b^2*d^2*m^3*n^2*x*x
^m*x^(4*n)*e^m + 921*B*b^3*c^2*m^2*n^3*x*x^m*x^(4*n)*e^m + 5526*B*a*b^2*c*d
*m^2*n^3*x*x^m*x^(4*n)*e^m + 1842*A*b^3*c*d*m^2*n^3*x*x^m*x^(4*n)*e^m + 276
3*B*a^2*b*d^2*m^2*n^3*x*x^m*x^(4*n)*e^m + 2763*A*a*b^2*d^2*m^2*n^3*x*x^m*x^
(4*n)*e^m + 792*B*b^3*c^2*m*n^4*x*x^m*x^(4*n)*e^m + 4752*B*a*b^2*c*d*m*n^4*
x*x^m*x^(4*n)*e^m + 1584*A*b^3*c*d*m*n^4*x*x^m*x^(4*n)*e^m + 2376*B*a^2*b*d
^2*m*n^4*x*x^m*x^(4*n)*e^m + 2376*A*a*b^2*d^2*m*n^4*x*x^m*x^(4*n)*e^m + 180
*B*b^3*c^2*n^5*x*x^m*x^(4*n)*e^m + 1080*B*a*b^2*c*d*n^5*x*x^m*x^(4*n)*e^m +
 360*A*b^3*c*d*n^5*x*x^m*x^(4*n)*e^m + 540*B*a^2*b*d^2*n^5*x*x^m*x^(4*n)*e^
m + 540*A*a*b^2*d^2*n^5*x*x^m*x^(4*n)*e^m + 18*B*a*b^2*c^2*m^5*x*x^m*x^(3*n
)*e^m + 6*A*b^3*c^2*m^5*x*x^m*x^(3*n)*e^m + 36*B*a^2*b*c*d*m^5*x*x^m*x^(3*n
)*e^m + 36*A*a*b^2*c*d*m^5*x*x^m*x^(3*n)*e^m + 6*B*a^3*d^2*m^5*x*x^m*x^(3*n
)*e^m + 18*A*a^2*b*d^2*m^5*x*x^m*x^(3*n)*e^m + 270*B*a*b^2*c^2*m^4*n*x*x^m*
x^(3*n)*e^m + 90*A*b^3*c^2*m^4*n*x*x^m*x^(3*n)*e^m + 540*B*a^2*b*c*d*m^4*n*
x*x^m*x^(3*n)*e^m + 540*A*a*b^2*c*d*m^4*n*x*x^m*x^(3*n)*e^m + 90*B*a^3*d^2*
m^4*n*x*x^m*x^(3*n)*e^m + 270*A*a^2*b*d^2*m^4*n*x*x^m*x^(3*n)*e^m + 1452*B*
a*b^2*c^2*m^3*n^2*x*x^m*x^(3*n)*e^m + 484*A*b^3*c^2*m^3*n^2*x*x^m*x^(3*n)*e
^m + 2904*B*a^2*b*c*d*m^3*n^2*x*x^m*x^(3*n)*e^m + 2904*A*a*b^2*c*d*m^3*n^2*
x*x^m*x^(3*n)*e^m + 484*B*a^3*d^2*m^3*n^2*x*x^m*x^(3*n)*e^m + 1452*A*a^2*b*
```

```
d^2*m^3*n^2*x*x^m*x^(3*n)*e^m + 3348*B*a*b^2*c^2*m^2*n^3*x*x^m*x^(3*n)*e^m
+ 1116*A*b^3*c^2*m^2*n^3*x*x^m*x^(3*n)*e^m + 6696*B*a^2*b*c*d*m^2*n^3*x*x^m
*x^(3*n)*e^m + 6696*A*a*b^2*c*d*m^2*n^3*x*x^m*x^(3*n)*e^m + 1116*B*a^3*d^2*
m^2*n^3*x*x^m*x^(3*n)*e^m + 3348*A*a^2*b*d^2*m^2*n^3*x*x^m*x^(3*n)*e^m + 30
48*B*a*b^2*c^2*m*n^4*x*x^m*x^(3*n)*e^m + 1016*A*b^3*c^2*m*n^4*x*x^m*x^(3*n)
*e^m + 6096*B*a^2*b*c*d*m*n^4*x*x^m*x^(3*n)*e^m + 6096*A*a*b^2*c*d*m*n^4*x*
x^m*x^(3*n)*e^m + 1016*B*a^3*d^2*m*n^4*x*x^m*x^(3*n)*e^m + 3048*A*a^2*b*d^2
*m*n^4*x*x^m*x^(3*n)*e^m + 720*B*a*b^2*c^2*n^5*x*x^m*x^(3*n)*e^m + 240*A*b^
3*c^2*n^5*x*x^m*x^(3*n)*e^m + 1440*B*a^2*b*c*d*n^5*x*x^m*x^(3*n)*e^m + 1440
*A*a*b^2*c*d*n^5*x*x^m*x^(3*n)*e^m + 240*B*a^3*d^2*n^5*x*x^m*x^(3*n)*e^m +
720*A*a^2*b*d^2*n^5*x*x^m*x^(3*n)*e^m + 18*B*a^2*b*c^2*m^5*x*x^m*x^(2*n)*e^
m + 18*A*a*b^2*c^2*m^5*x*x^m*x^(2*n)*e^m + 12*B*a^3*c*d*m^5*x*x^m*x^(2*n)*e
^m + 36*A*a^2*b*c*d*m^5*x*x^m*x^(2*n)*e^m + 6*A*a^3*d^2*m^5*x*x^m*x^(2*n)*e
^m + 285*B*a^2*b*c^2*m^4*n*x*x^m*x^(2*n)*e^m + 285*A*a*b^2*c^2*m^4*n*x*x^m*
x^(2*n)*e^m + 190*B*a^3*c*d*m^4*n*x*x^m*x^(2*n)*e^m + 570*A*a^2*b*c*d*m^4*n
*x*x^m*x^(2*n)*e^m + 95*A*a^3*d^2*m^4*n*x*x^m*x^(2*n)*e^m + 1644*B*a^2*b*c^
2*m^3*n^2*x*x^m*x^(2*n)*e^m + 1644*A*a*b^2*c^2*m^3*n^2*x*x^m*x^(2*n)*e^m +
1096*B*a^3*c*d*m^3*n^2*x*x^m*x^(2*n)*e^m + 3288*A*a^2*b*c*d*m^3*n^2*x*x^m*x
^(2*n)*e^m + 548*A*a^3*d^2*m^3*n^2*x*x^m*x^(2*n)*e^m + 4149*B*a^2*b*c^2*m^2
*n^3*x*x^m*x^(2*n)*e^m + 4149*A*a*b^2*c^2*m^2*n^3*x*x^m*x^(2*n)*e^m + 2766*
B*a^3*c*d*m^2*n^3*x*x^m*x^(2*n)*e^m + 8298*A*a^2*b*c*d*m^2*n^3*x*x^m*x^(2*n
)*e^m + 1383*A*a^3*d^2*m^2*n^3*x*x^m*x^(2*n)*e^m + 4212*B*a^2*b*c^2*m*n^4*x
*x^m*x^(2*n)*e^m + 4212*A*a*b^2*c^2*m*n^4*x*x^m*x^(2*n)*e^m + 2808*B*a^3*c*
d*m*n^4*x*x^m*x^(2*n)*e^m + 8424*A*a^2*b*c*d*m*n^4*x*x^m*x^(2*n)*e^m + 1404
*A*a^3*d^2*m*n^4*x*x^m*x^(2*n)*e^m + 1080*B*a^2*b*c^2*n^5*x*x^m*x^(2*n)*e^m
 + 1080*A*a*b^2*c^2*n^5*x*x^m*x^(2*n)*e^m + 720*B*a^3*c*d*n^5*x*x^m*x^(2*n)
*e^m + 2160*A*a^2*b*c*d*n^5*x*x^m*x^(2*n)*e^m + 360*A*a^3*d^2*n^5*x*x^m*x^(
2*n)*e^m + 6*B*a^3*c^2*m^5*x*x^m*x^n*e^m + 18*A*a^2*b*c^2*m^5*x*x^m*x^n*e^m
 + 12*A*a^3*c*d*m^5*x*x^m*x^n*e^m + 100*B*a^3*c^2*m^4*n*x*x^m*x^n*e^m + 300
*A*a^2*b*c^2*m^4*n*x*x^m*x^n*e^m + 200*A*a^3*c*d*m^4*n*x*x^m*x^n*e^m + 620*
B*a^3*c^2*m^3*n^2*x*x^m*x^n*e^m + 1860*A*a^2*b*c^2*m^3*n^2*x*x^m*x^n*e^m +
1240*A*a^3*c*d*m^3*n^2*x*x^m*x^n*e^m + 1740*B*a^3*c^2*m^2*n^3*x*x^m*x^n*e^m
 + 5220*A*a^2*b*c^2*m^2*n^3*x*x^m*x^n*e^m + 3480*A*a^3*c*d*m^2*n^3*x*x^m*x^
n*e^m + 2088*B*a^3*c^2*m*n^4*x*x^m*x^n*e^m + 6264*A*a^2*b*c^2*m*n^4*x*x^m*x
^n*e^m + 4176*A*a^3*c*d*m*n^4*x*x^m*x^n*e^m + 720*B*a^3*c^2*n^5*x*x^m*x^n*e
^m + 2160*A*a^2*b*c^2*n^5*x*x^m*x^n*e^m + 1440*A*a^3*c*d*n^5*x*x^m*x^n*e^m
+ 6*A*a^3*c^2*m^5*x*x^m*e^m + 105*A*a^3*c^2*m^4*n*x*x^m*e^m + 700*A*a^3*c^2
*m^3*n^2*x*x^m*e^m + 2205*A*a^3*c^2*m^2*n^3*x*x^m*e^m + 3248*A*a^3*c^2*m*n^
4*x*x^m*e^m + 1764*A*a^3*c^2*n^5*x*x^m*e^m + 15*B*b^3*d^2*m^4*x*x^m*x^(6*n)
*e^m + 150*B*b^3*d^2*m^3*n*x*x^m*x^(6*n)*e^m + 510*B*b^3*d^2*m^2*n^2*x*x^m*
x^(6*n)*e^m + 675*B*b^3*d^2*m*n^3*x*x^m*x^(6*n)*e^m + 274*B*b^3*d^2*n^4*x*x
^m*x^(6*n)*e^m + 30*B*b^3*c*d*m^4*x*x^m*x^(5*n)*e^m + 45*B*a*b^2*d^2*m^4*x*
x^m*x^(5*n)*e^m + 15*A*b^3*d^2*m^4*x*x^m*x^(5*n)*e^m + 320*B*b^3*c*d*m^3*n*
x*x^m*x^(5*n)*e^m + 480*B*a*b^2*d^2*m^3*n*x*x^m*x^(5*n)*e^m + 160*A*b^3*d^2
*m^3*n*x*x^m*x^(5*n)*e^m + 1140*B*b^3*c*d*m^2*n^2*x*x^m*x^(5*n)*e^m + 1710*
B*a*b^2*d^2*m^2*n^2*x*x^m*x^(5*n)*e^m + 570*A*b^3*d^2*m^2*n^2*x*x^m*x^(5*n)
*e^m + 1560*B*b^3*c*d*m*n^3*x*x^m*x^(5*n)*e^m + 2340*B*a*b^2*d^2*m*n^3*x*x^
m*x^(5*n)*e^m + 780*A*b^3*d^2*m*n^3*x*x^m*x^(5*n)*e^m + 648*B*b^3*c*d*n^4*x
*x^m*x^(5*n)*e^m + 972*B*a*b^2*d^2*n^4*x*x^m*x^(5*n)*e^m + 324*A*b^3*d^2*n^
4*x*x^m*x^(5*n)*e^m + 15*B*b^3*c^2*m^4*x*x^m*x^(4*n)*e^m + 90*B*a*b^2*c*d*m
^4*x*x^m*x^(4*n)*e^m + 30*A*b^3*c*d*m^4*x*x^m*x^(4*n)*e^m + 45*B*a^2*b*d^2*
m^4*x*x^m*x^(4*n)*e^m + 45*A*a*b^2*d^2*m^4*x*x^m*x^(4*n)*e^m + 170*B*b^3*c^
2*m^3*n*x*x^m*x^(4*n)*e^m + 1020*B*a*b^2*c*d*m^3*n*x*x^m*x^(4*n)*e^m + 340*
A*b^3*c*d*m^3*n*x*x^m*x^(4*n)*e^m + 510*B*a^2*b*d^2*m^3*n*x*x^m*x^(4*n)*e^m
 + 510*A*a*b^2*d^2*m^3*n*x*x^m*x^(4*n)*e^m + 642*B*b^3*c^2*m^2*n^2*x*x^m*x^
(4*n)*e^m + 3852*B*a*b^2*c*d*m^2*n^2*x*x^m*x^(4*n)*e^m + 1284*A*b^3*c*d*m^2
*n^2*x*x^m*x^(4*n)*e^m + 1926*B*a^2*b*d^2*m^2*n^2*x*x^m*x^(4*n)*e^m + 1926*
A*a*b^2*d^2*m^2*n^2*x*x^m*x^(4*n)*e^m + 921*B*b^3*c^2*m*n^3*x*x^m*x^(4*n)*e
^m + 5526*B*a*b^2*c*d*m*n^3*x*x^m*x^(4*n)*e^m + 1842*A*b^3*c*d*m*n^3*x*x^m*
```

x^(4*n)*e^m + 2763*B*a^2*b*d^2*m*n^3*x*x^m*x^(4*n)*e^m + 2763*A*a*b^2*d^2*m
*n^3*x*x^m*x^(4*n)*e^m + 396*B*b^3*c^2*n^4*x*x^m*x^(4*n)*e^m + 2376*B*a*b^2
*c*d*n^4*x*x^m*x^(4*n)*e^m + 792*A*b^3*c*d*n^4*x*x^m*x^(4*n)*e^m + 1188*B*a
^2*b*d^2*n^4*x*x^m*x^(4*n)*e^m + 1188*A*a*b^2*d^2*n^4*x*x^m*x^(4*n)*e^m + 4
5*B*a*b^2*c^2*m^4*x*x^m*x^(3*n)*e^m + 15*A*b^3*c^2*m^4*x*x^m*x^(3*n)*e^m +
90*B*a^2*b*c*d*m^4*x*x^m*x^(3*n)*e^m + 90*A*a*b^2*c*d*m^4*x*x^m*x^(3*n)*e^m
+ 15*B*a^3*d^2*m^4*x*x^m*x^(3*n)*e^m + 45*A*a^2*b*d^2*m^4*x*x^m*x^(3*n)*e^
m + 540*B*a*b^2*c^2*m^3*n*x*x^m*x^(3*n)*e^m + 180*A*b^3*c^2*m^3*n*x*x^m*x^(
3*n)*e^m + 1080*B*a^2*b*c*d*m^3*n*x*x^m*x^(3*n)*e^m + 1080*A*a*b^2*c*d*m^3*
n*x*x^m*x^(3*n)*e^m + 180*B*a^3*d^2*m^3*n*x*x^m*x^(3*n)*e^m + 540*A*a^2*b*d
^2*m^3*n*x*x^m*x^(3*n)*e^m + 2178*B*a*b^2*c^2*m^2*n^2*x*x^m*x^(3*n)*e^m + 7
26*A*b^3*c^2*m^2*n^2*x*x^m*x^(3*n)*e^m + 4356*B*a^2*b*c*d*m^2*n^2*x*x^m*x^(
3*n)*e^m + 4356*A*a*b^2*c*d*m^2*n^2*x*x^m*x^(3*n)*e^m + 726*B*a^3*d^2*m^2*n
^2*x*x^m*x^(3*n)*e^m + 2178*A*a^2*b*d^2*m^2*n^2*x*x^m*x^(3*n)*e^m + 3348*B*
a*b^2*c^2*m*n^3*x*x^m*x^(3*n)*e^m + 1116*A*b^3*c^2*m*n^3*x*x^m*x^(3*n)*e^m
+ 6696*B*a^2*b*c*d*m*n^3*x*x^m*x^(3*n)*e^m + 6696*A*a*b^2*c*d*m*n^3*x*x^m*x
^(3*n)*e^m + 1116*B*a^3*d^2*m*n^3*x*x^m*x^(3*n)*e^m + 3348*A*a^2*b*d^2*m*n^
3*x*x^m*x^(3*n)*e^m + 1524*B*a*b^2*c^2*n^4*x*x^m*x^(3*n)*e^m + 508*A*b^3*c^
2*n^4*x*x^m*x^(3*n)*e^m + 3048*B*a^2*b*c*d*n^4*x*x^m*x^(3*n)*e^m + 3048*A*a
*b^2*c*d*n^4*x*x^m*x^(3*n)*e^m + 508*B*a^3*d^2*n^4*x*x^m*x^(3*n)*e^m + 1524
*A*a^2*b*d^2*n^4*x*x^m*x^(3*n)*e^m + 45*B*a^2*b*c^2*m^4*x*x^m*x^(2*n)*e^m +
45*A*a*b^2*c^2*m^4*x*x^m*x^(2*n)*e^m + 30*B*a^3*c*d*m^4*x*x^m*x^(2*n)*e^m
+ 90*A*a^2*b*c*d*m^4*x*x^m*x^(2*n)*e^m + 15*A*a^3*d^2*m^4*x*x^m*x^(2*n)*e^m
+ 570*B*a^2*b*c^2*m^3*n*x*x^m*x^(2*n)*e^m + 570*A*a*b^2*c^2*m^3*n*x*x^m*x^
(2*n)*e^m + 380*B*a^3*c*d*m^3*n*x*x^m*x^(2*n)*e^m + 1140*A*a^2*b*c*d*m^3*n*
x*x^m*x^(2*n)*e^m + 190*A*a^3*d^2*m^3*n*x*x^m*x^(2*n)*e^m + 2466*B*a^2*b*c^
2*m^2*n^2*x*x^m*x^(2*n)*e^m + 2466*A*a*b^2*c^2*m^2*n^2*x*x^m*x^(2*n)*e^m +
1644*B*a^3*c*d*m^2*n^2*x*x^m*x^(2*n)*e^m + 4932*A*a^2*b*c*d*m^2*n^2*x*x^m*x
^(2*n)*e^m + 822*A*a^3*d^2*m^2*n^2*x*x^m*x^(2*n)*e^m + 4149*B*a^2*b*c^2*m*n
^3*x*x^m*x^(2*n)*e^m + 4149*A*a*b^2*c^2*m*n^3*x*x^m*x^(2*n)*e^m + 2766*B*a^
3*c*d*m*n^3*x*x^m*x^(2*n)*e^m + 8298*A*a^2*b*c*d*m*n^3*x*x^m*x^(2*n)*e^m +
1383*A*a^3*d^2*m*n^3*x*x^m*x^(2*n)*e^m + 2106*B*a^2*b*c^2*n^4*x*x^m*x^(2*n)
*e^m + 2106*A*a*b^2*c^2*n^4*x*x^m*x^(2*n)*e^m + 1404*B*a^3*c*d*n^4*x*x^m*x^
(2*n)*e^m + 4212*A*a^2*b*c*d*n^4*x*x^m*x^(2*n)*e^m + 702*A*a^3*d^2*n^4*x*x^
m*x^(2*n)*e^m + 15*B*a^3*c^2*m^4*x*x^m*x^n*e^m + 45*A*a^2*b*c^2*m^4*x*x^m*x
^n*e^m + 30*A*a^3*c*d*m^4*x*x^m*x^n*e^m + 200*B*a^3*c^2*m^3*n*x*x^m*x^n*e^m
+ 600*A*a^2*b*c^2*m^3*n*x*x^m*x^n*e^m + 400*A*a^3*c*d*m^3*n*x*x^m*x^n*e^m
+ 930*B*a^3*c^2*m^2*n^2*x*x^m*x^n*e^m + 2790*A*a^2*b*c^2*m^2*n^2*x*x^m*x^n*
e^m + 1860*A*a^3*c*d*m^2*n^2*x*x^m*x^n*e^m + 1740*B*a^3*c^2*m*n^3*x*x^m*x^n
*e^m + 5220*A*a^2*b*c^2*m*n^3*x*x^m*x^n*e^m + 3480*A*a^3*c*d*m*n^3*x*x^m*x^
n*e^m + 1044*B*a^3*c^2*n^4*x*x^m*x^n*e^m + 3132*A*a^2*b*c^2*n^4*x*x^m*x^n*e
^m + 2088*A*a^3*c*d*n^4*x*x^m*x^n*e^m + 15*A*a^3*c^2*m^4*x*x^m*e^m + 210*A*
a^3*c^2*m^3*n*x*x^m*e^m + 1050*A*a^3*c^2*m^2*n^2*x*x^m*e^m + 2205*A*a^3*c^2
*m*n^3*x*x^m*e^m + 1624*A*a^3*c^2*n^4*x*x^m*e^m + 20*B*b^3*d^2*m^3*x*x^m*x^
(6*n)*e^m + 150*B*b^3*d^2*m^2*n*x*x^m*x^(6*n)*e^m + 340*B*b^3*d^2*m*n^2*x*x
^m*x^(6*n)*e^m + 225*B*b^3*d^2*n^3*x*x^m*x^(6*n)*e^m + 40*B*b^3*c*d*m^3*x*x
^m*x^(5*n)*e^m + 60*B*a*b^2*d^2*m^3*x*x^m*x^(5*n)*e^m + 20*A*b^3*d^2*m^3*x*
x^m*x^(5*n)*e^m + 320*B*b^3*c*d*m^2*n*x*x^m*x^(5*n)*e^m + 480*B*a*b^2*d^2*m
^2*n*x*x^m*x^(5*n)*e^m + 160*A*b^3*d^2*m^2*n*x*x^m*x^(5*n)*e^m + 760*B*b^3*
c*d*m*n^2*x*x^m*x^(5*n)*e^m + 1140*B*a*b^2*d^2*m*n^2*x*x^m*x^(5*n)*e^m + 38
0*A*b^3*d^2*m*n^2*x*x^m*x^(5*n)*e^m + 520*B*b^3*c*d*n^3*x*x^m*x^(5*n)*e^m +
780*B*a*b^2*d^2*n^3*x*x^m*x^(5*n)*e^m + 260*A*b^3*d^2*n^3*x*x^m*x^(5*n)*e^
m + 20*B*b^3*c^2*m^3*x*x^m*x^(4*n)*e^m + 120*B*a*b^2*c*d*m^3*x*x^m*x^(4*n)*
e^m + 40*A*b^3*c*d*m^3*x*x^m*x^(4*n)*e^m + 60*B*a^2*b*d^2*m^3*x*x^m*x^(4*n)
*e^m + 60*A*a*b^2*d^2*m^3*x*x^m*x^(4*n)*e^m + 170*B*b^3*c^2*m^2*n*x*x^m*x^(
4*n)*e^m + 1020*B*a*b^2*c*d*m^2*n*x*x^m*x^(4*n)*e^m + 340*A*b^3*c*d*m^2*n*x
*x^m*x^(4*n)*e^m + 510*B*a^2*b*d^2*m^2*n*x*x^m*x^(4*n)*e^m + 510*A*a*b^2*d^
2*m^2*n*x*x^m*x^(4*n)*e^m + 428*B*b^3*c^2*m*n^2*x*x^m*x^(4*n)*e^m + 2568*B*
a*b^2*c*d*m*n^2*x*x^m*x^(4*n)*e^m + 856*A*b^3*c*d*m*n^2*x*x^m*x^(4*n)*e^m +

```
 1284*B*a^2*b*d^2*m*n^2*x*x^m*x^(4*n)*e^m + 1284*A*a*b^2*d^2*m*n^2*x*x^m*x^
(4*n)*e^m + 307*B*b^3*c^2*n^3*x*x^m*x^(4*n)*e^m + 1842*B*a*b^2*c*d*n^3*x*x^
m*x^(4*n)*e^m + 614*A*b^3*c*d*n^3*x*x^m*x^(4*n)*e^m + 921*B*a^2*b*d^2*n^3*x
*x^m*x^(4*n)*e^m + 921*A*a*b^2*d^2*n^3*x*x^m*x^(4*n)*e^m + 60*B*a*b^2*c^2*m
^3*x*x^m*x^(3*n)*e^m + 20*A*b^3*c^2*m^3*x*x^m*x^(3*n)*e^m + 120*B*a^2*b*c*d
*m^3*x*x^m*x^(3*n)*e^m + 120*A*a*b^2*c*d*m^3*x*x^m*x^(3*n)*e^m + 20*B*a^3*d
^2*m^3*x*x^m*x^(3*n)*e^m + 60*A*a^2*b*d^2*m^3*x*x^m*x^(3*n)*e^m + 540*B*a*b
^2*c^2*m^2*n*x*x^m*x^(3*n)*e^m + 180*A*b^3*c^2*m^2*n*x*x^m*x^(3*n)*e^m + 10
80*B*a^2*b*c*d*m^2*n*x*x^m*x^(3*n)*e^m + 1080*A*a*b^2*c*d*m^2*n*x*x^m*x^(3*
n)*e^m + 180*B*a^3*d^2*m^2*n*x*x^m*x^(3*n)*e^m + 540*A*a^2*b*d^2*m^2*n*x*x^
m*x^(3*n)*e^m + 1452*B*a*b^2*c^2*m*n^2*x*x^m*x^(3*n)*e^m + 484*A*b^3*c^2*m*
n^2*x*x^m*x^(3*n)*e^m + 2904*B*a^2*b*c*d*m*n^2*x*x^m*x^(3*n)*e^m + 2904*A*a
*b^2*c*d*m*n^2*x*x^m*x^(3*n)*e^m + 484*B*a^3*d^2*m*n^2*x*x^m*x^(3*n)*e^m +
1452*A*a^2*b*d^2*m*n^2*x*x^m*x^(3*n)*e^m + 1116*B*a*b^2*c^2*n^3*x*x^m*x^(3*
n)*e^m + 372*A*b^3*c^2*n^3*x*x^m*x^(3*n)*e^m + 2232*B*a^2*b*c*d*n^3*x*x^m*x
^(3*n)*e^m + 2232*A*a*b^2*c*d*n^3*x*x^m*x^(3*n)*e^m + 372*B*a^3*d^2*n^3*x*x
^m*x^(3*n)*e^m + 1116*A*a^2*b*d^2*n^3*x*x^m*x^(3*n)*e^m + 60*B*a^2*b*c^2*m^
3*x*x^m*x^(2*n)*e^m + 60*A*a*b^2*c^2*m^3*x*x^m*x^(2*n)*e^m + 40*B*a^3*c*d*m
^3*x*x^m*x^(2*n)*e^m + 120*A*a^2*b*c*d*m^3*x*x^m*x^(2*n)*e^m + 20*A*a^3*d^2
*m^3*x*x^m*x^(2*n)*e^m + 570*B*a^2*b*c^2*m^2*n*x*x^m*x^(2*n)*e^m + 570*A*a*
b^2*c^2*m^2*n*x*x^m*x^(2*n)*e^m + 380*B*a^3*c*d*m^2*n*x*x^m*x^(2*n)*e^m + 1
140*A*a^2*b*c*d*m^2*n*x*x^m*x^(2*n)*e^m + 190*A*a^3*d^2*m^2*n*x*x^m*x^(2*n)
*e^m + 1644*B*a^2*b*c^2*m*n^2*x*x^m*x^(2*n)*e^m + 1644*A*a*b^2*c^2*m*n^2*x*
x^m*x^(2*n)*e^m + 1096*B*a^3*c*d*m*n^2*x*x^m*x^(2*n)*e^m + 3288*A*a^2*b*c*d
*m*n^2*x*x^m*x^(2*n)*e^m + 548*A*a^3*d^2*m*n^2*x*x^m*x^(2*n)*e^m + 1383*B*a
^2*b*c^2*n^3*x*x^m*x^(2*n)*e^m + 1383*A*a*b^2*c^2*n^3*x*x^m*x^(2*n)*e^m + 9
22*B*a^3*c*d*n^3*x*x^m*x^(2*n)*e^m + 2766*A*a^2*b*c*d*n^3*x*x^m*x^(2*n)*e^m
 + 461*A*a^3*d^2*n^3*x*x^m*x^(2*n)*e^m + 20*B*a^3*c^2*m^3*x*x^m*x^n*e^m + 6
0*A*a^2*b*c^2*m^3*x*x^m*x^n*e^m + 40*A*a^3*c*d*m^3*x*x^m*x^n*e^m + 200*B*a^
3*c^2*m^2*n*x*x^m*x^n*e^m + 600*A*a^2*b*c^2*m^2*n*x*x^m*x^n*e^m + 400*A*a^3
*c*d*m^2*n*x*x^m*x^n*e^m + 620*B*a^3*c^2*m*n^2*x*x^m*x^n*e^m + 1860*A*a^2*b
*c^2*m*n^2*x*x^m*x^n*e^m + 1240*A*a^3*c*d*m*n^2*x*x^m*x^n*e^m + 580*B*a^3*c
^2*n^3*x*x^m*x^n*e^m + 1740*A*a^2*b*c^2*n^3*x*x^m*x^n*e^m + 1160*A*a^3*c*d*
n^3*x*x^m*x^n*e^m + 20*A*a^3*c^2*m^3*x*x^m*e^m + 210*A*a^3*c^2*m^2*n*x*x^m*
e^m + 700*A*a^3*c^2*m*n^2*x*x^m*e^m + 735*A*a^3*c^2*n^3*x*x^m*e^m + 15*B*b^
3*d^2*m^2*x*x^m*x^(6*n)*e^m + 75*B*b^3*d^2*m*n*x*x^m*x^(6*n)*e^m + 85*B*b^3
*d^2*n^2*x*x^m*x^(6*n)*e^m + 30*B*b^3*c*d*m^2*x*x^m*x^(5*n)*e^m + 45*B*a*b^
2*d^2*m^2*x*x^m*x^(5*n)*e^m + 15*A*b^3*d^2*m^2*x*x^m*x^(5*n)*e^m + 160*B*b^
3*c*d*m*n*x*x^m*x^(5*n)*e^m + 240*B*a*b^2*d^2*m*n*x*x^m*x^(5*n)*e^m + 80*A*
b^3*d^2*m*n*x*x^m*x^(5*n)*e^m + 190*B*b^3*c*d*n^2*x*x^m*x^(5*n)*e^m + 285*B
*a*b^2*d^2*n^2*x*x^m*x^(5*n)*e^m + 95*A*b^3*d^2*n^2*x*x^m*x^(5*n)*e^m + 15*
B*b^3*c^2*m^2*x*x^m*x^(4*n)*e^m + 90*B*a*b^2*c*d*m^2*x*x^m*x^(4*n)*e^m + 30
*A*b^3*c*d*m^2*x*x^m*x^(4*n)*e^m + 45*B*a^2*b*d^2*m^2*x*x^m*x^(4*n)*e^m + 4
5*A*a*b^2*d^2*m^2*x*x^m*x^(4*n)*e^m + 85*B*b^3*c^2*m*n*x*x^m*x^(4*n)*e^m +
510*B*a*b^2*c*d*m*n*x*x^m*x^(4*n)*e^m + 170*A*b^3*c*d*m*n*x*x^m*x^(4*n)*e^m
 + 255*B*a^2*b*d^2*m*n*x*x^m*x^(4*n)*e^m + 255*A*a*b^2*d^2*m*n*x*x^m*x^(4*n
)*e^m + 107*B*b^3*c^2*n^2*x*x^m*x^(4*n)*e^m + 642*B*a*b^2*c*d*n^2*x*x^m*x^(
4*n)*e^m + 214*A*b^3*c*d*n^2*x*x^m*x^(4*n)*e^m + 321*B*a^2*b*d^2*n^2*x*x^m*
x^(4*n)*e^m + 321*A*a*b^2*d^2*n^2*x*x^m*x^(4*n)*e^m + 45*B*a*b^2*c^2*m^2*x*
x^m*x^(3*n)*e^m + 15*A*b^3*c^2*m^2*x*x^m*x^(3*n)*e^m + 90*B*a^2*b*c*d*m^2*x
*x^m*x^(3*n)*e^m + 90*A*a*b^2*c*d*m^2*x*x^m*x^(3*n)*e^m + 15*B*a^3*d^2*m^2*
x*x^m*x^(3*n)*e^m + 45*A*a^2*b*d^2*m^2*x*x^m*x^(3*n)*e^m + 270*B*a*b^2*c^2*
m*n*x*x^m*x^(3*n)*e^m + 90*A*b^3*c^2*m*n*x*x^m*x^(3*n)*e^m + 540*B*a^2*b*c*
d*m*n*x*x^m*x^(3*n)*e^m + 540*A*a*b^2*c*d*m*n*x*x^m*x^(3*n)*e^m + 90*B*a^3*
d^2*m*n*x*x^m*x^(3*n)*e^m + 270*A*a^2*b*d^2*m*n*x*x^m*x^(3*n)*e^m + 363*B*a
*b^2*c^2*n^2*x*x^m*x^(3*n)*e^m + 121*A*b^3*c^2*n^2*x*x^m*x^(3*n)*e^m + 726*
B*a^2*b*c*d*n^2*x*x^m*x^(3*n)*e^m + 726*A*a*b^2*c*d*n^2*x*x^m*x^(3*n)*e^m +
 121*B*a^3*d^2*n^2*x*x^m*x^(3*n)*e^m + 363*A*a^2*b*d^2*n^2*x*x^m*x^(3*n)*e^
m + 45*B*a^2*b*c^2*m^2*x*x^m*x^(2*n)*e^m + 45*A*a*b^2*c^2*m^2*x*x^m*x^(2*n)
```

```
*e^m + 30*B*a^3*c*d*m^2*x*x^m*x^(2*n)*e^m + 90*A*a^2*b*c*d*m^2*x*x^m*x^(2*n
)*e^m + 15*A*a^3*d^2*m^2*x*x^m*x^(2*n)*e^m + 285*B*a^2*b*c^2*m*n*x*x^m*x^(2
*n)*e^m + 285*A*a*b^2*c^2*m*n*x*x^m*x^(2*n)*e^m + 190*B*a^3*c*d*m*n*x*x^m*x
^(2*n)*e^m + 570*A*a^2*b*c*d*m*n*x*x^m*x^(2*n)*e^m + 95*A*a^3*d^2*m*n*x*x^m
*x^(2*n)*e^m + 411*B*a^2*b*c^2*n^2*x*x^m*x^(2*n)*e^m + 411*A*a*b^2*c^2*n^2*
x*x^m*x^(2*n)*e^m + 274*B*a^3*c*d*n^2*x*x^m*x^(2*n)*e^m + 822*A*a^2*b*c*d*n
^2*x*x^m*x^(2*n)*e^m + 137*A*a^3*d^2*n^2*x*x^m*x^(2*n)*e^m + 15*B*a^3*c^2*m
^2*x*x^m*x^n*e^m + 45*A*a^2*b*c^2*m^2*x*x^m*x^n*e^m + 30*A*a^3*c*d*m^2*x*x^
m*x^n*e^m + 100*B*a^3*c^2*m*n*x*x^m*x^n*e^m + 300*A*a^2*b*c^2*m*n*x*x^m*x^n
*e^m + 200*A*a^3*c*d*m*n*x*x^m*x^n*e^m + 155*B*a^3*c^2*n^2*x*x^m*x^n*e^m +
465*A*a^2*b*c^2*n^2*x*x^m*x^n*e^m + 310*A*a^3*c*d*n^2*x*x^m*x^n*e^m + 15*A*
a^3*c^2*m^2*x*x^m*e^m + 105*A*a^3*c^2*m*n*x*x^m*e^m + 175*A*a^3*c^2*n^2*x*x
^m*e^m + 6*B*b^3*d^2*m*x*x^m*x^(6*n)*e^m + 15*B*b^3*d^2*n*x*x^m*x^(6*n)*e^m
 + 12*B*b^3*c*d*m*x*x^m*x^(5*n)*e^m + 18*B*a*b^2*d^2*m*x*x^m*x^(5*n)*e^m +
6*A*b^3*d^2*m*x*x^m*x^(5*n)*e^m + 32*B*b^3*c*d*n*x*x^m*x^(5*n)*e^m + 48*B*a
*b^2*d^2*n*x*x^m*x^(5*n)*e^m + 16*A*b^3*d^2*n*x*x^m*x^(5*n)*e^m + 6*B*b^3*c
^2*m*x*x^m*x^(4*n)*e^m + 36*B*a*b^2*c*d*m*x*x^m*x^(4*n)*e^m + 12*A*b^3*c*d*
m*x*x^m*x^(4*n)*e^m + 18*B*a^2*b*d^2*m*x*x^m*x^(4*n)*e^m + 18*A*a*b^2*d^2*m
*x*x^m*x^(4*n)*e^m + 17*B*b^3*c^2*n*x*x^m*x^(4*n)*e^m + 102*B*a*b^2*c*d*n*x
*x^m*x^(4*n)*e^m + 34*A*b^3*c*d*n*x*x^m*x^(4*n)*e^m + 51*B*a^2*b*d^2*n*x*x^
m*x^(4*n)*e^m + 51*A*a*b^2*d^2*n*x*x^m*x^(4*n)*e^m + 18*B*a*b^2*c^2*m*x*x^m
*x^(3*n)*e^m + 6*A*b^3*c^2*m*x*x^m*x^(3*n)*e^m + 36*B*a^2*b*c*d*m*x*x^m*x^(
3*n)*e^m + 36*A*a*b^2*c*d*m*x*x^m*x^(3*n)*e^m + 6*B*a^3*d^2*m*x*x^m*x^(3*n)
*e^m + 18*A*a^2*b*d^2*m*x*x^m*x^(3*n)*e^m + 54*B*a*b^2*c^2*n*x*x^m*x^(3*n)*
e^m + 18*A*b^3*c^2*n*x*x^m*x^(3*n)*e^m + 108*B*a^2*b*c*d*n*x*x^m*x^(3*n)*e^
m + 108*A*a*b^2*c*d*n*x*x^m*x^(3*n)*e^m + 18*B*a^3*d^2*n*x*x^m*x^(3*n)*e^m
+ 54*A*a^2*b*d^2*n*x*x^m*x^(3*n)*e^m + 18*B*a^2*b*c^2*m*x*x^m*x^(2*n)*e^m +
 18*A*a*b^2*c^2*m*x*x^m*x^(2*n)*e^m + 12*B*a^3*c*d*m*x*x^m*x^(2*n)*e^m + 36
*A*a^2*b*c*d*m*x*x^m*x^(2*n)*e^m + 6*A*a^3*d^2*m*x*x^m*x^(2*n)*e^m + 57*B*a
^2*b*c^2*n*x*x^m*x^(2*n)*e^m + 57*A*a*b^2*c^2*n*x*x^m*x^(2*n)*e^m + 38*B*a^
3*c*d*n*x*x^m*x^(2*n)*e^m + 114*A*a^2*b*c*d*n*x*x^m*x^(2*n)*e^m + 19*A*a^3*
d^2*n*x*x^m*x^(2*n)*e^m + 6*B*a^3*c^2*m*x*x^m*x^n*e^m + 18*A*a^2*b*c^2*m*x*
x^m*x^n*e^m + 12*A*a^3*c*d*m*x*x^m*x^n*e^m + 20*B*a^3*c^2*n*x*x^m*x^n*e^m +
 60*A*a^2*b*c^2*n*x*x^m*x^n*e^m + 40*A*a^3*c*d*n*x*x^m*x^n*e^m + 6*A*a^3*c^
2*m*x*x^m*e^m + 21*A*a^3*c^2*n*x*x^m*e^m + B*b^3*d^2*x*x^m*x^(6*n)*e^m + 2*
B*b^3*c*d*x*x^m*x^(5*n)*e^m + 3*B*a*b^2*d^2*x*x^m*x^(5*n)*e^m + A*b^3*d^2*x
*x^m*x^(5*n)*e^m + B*b^3*c^2*x*x^m*x^(4*n)*e^m + 6*B*a*b^2*c*d*x*x^m*x^(4*n
)*e^m + 2*A*b^3*c*d*x*x^m*x^(4*n)*e^m + 3*B*a^2*b*d^2*x*x^m*x^(4*n)*e^m + 3
*A*a*b^2*d^2*x*x^m*x^(4*n)*e^m + 3*B*a*b^2*c^2*x*x^m*x^(3*n)*e^m + A*b^3*c^
2*x*x^m*x^(3*n)*e^m + 6*B*a^2*b*c*d*x*x^m*x^(3*n)*e^m + 6*A*a*b^2*c*d*x*x^m
*x^(3*n)*e^m + B*a^3*d^2*x*x^m*x^(3*n)*e^m + 3*A*a^2*b*d^2*x*x^m*x^(3*n)*e^
m + 3*B*a^2*b*c^2*x*x^m*x^(2*n)*e^m + 3*A*a*b^2*c^2*x*x^m*x^(2*n)*e^m + 2*B
*a^3*c*d*x*x^m*x^(2*n)*e^m + 6*A*a^2*b*c*d*x*x^m*x^(2*n)*e^m + A*a^3*d^2*x*
x^m*x^(2*n)*e^m + B*a^3*c^2*x*x^m*x^n*e^m + 3*A*a^2*b*c^2*x*x^m*x^n*e^m + 2
*A*a^3*c*d*x*x^m*x^n*e^m + A*a^3*c^2*x*x^m*e^m)/(m^7 + 21*m^6*n + 175*m^5*n
^2 + 735*m^4*n^3 + 1624*m^3*n^4 + 1764*m^2*n^5 + 720*m*n^6 + 7*m^6 + 126*m^
5*n + 875*m^4*n^2 + 2940*m^3*n^3 + 4872*m^2*n^4 + 3528*m*n^5 + 720*n^6 + 21
*m^5 + 315*m^4*n + 1750*m^3*n^2 + 4410*m^2*n^3 + 4872*m*n^4 + 1764*n^5 + 35
*m^4 + 420*m^3*n + 1750*m^2*n^2 + 2940*m*n^3 + 1624*n^4 + 35*m^3 + 315*m^2*
n + 875*m*n^2 + 735*n^3 + 21*m^2 + 126*m*n + 175*n^2 + 7*m + 21*n + 1)
```

maple [C] time = 0.23, size = 11389, normalized size = 35.81

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(b*x^n+a)^3*(B*x^n+A)*(d*x^n+c)^2,x)

[Out] result too large to display
```

maxima [B] time = 1.12, size = 748, normalized size = 2.35

Verification of antiderivative is not currently implemented for this CAS.

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

```
[Out] B*b^3*d^2*e^m*x*e^(m*log(x) + 6*n*log(x))/(m + 6*n + 1) + 2*B*b^3*c*d*e^m*x
*e^(m*log(x) + 5*n*log(x))/(m + 5*n + 1) + 3*B*a*b^2*d^2*e^m*x*e^(m*log(x)
+ 5*n*log(x))/(m + 5*n + 1) + A*b^3*d^2*e^m*x*e^(m*log(x) + 5*n*log(x))/(m
+ 5*n + 1) + B*b^3*c^2*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) + 6*B*
a*b^2*c*d*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) + 2*A*b^3*c*d*e^m*x
*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) + 3*B*a^2*b*d^2*e^m*x*e^(m*log(x)
+ 4*n*log(x))/(m + 4*n + 1) + 3*A*a*b^2*d^2*e^m*x*e^(m*log(x) + 4*n*log(x))
/(m + 4*n + 1) + 3*B*a*b^2*c^2*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1
) + A*b^3*c^2*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + 6*B*a^2*b*c*d
*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + 6*A*a*b^2*c*d*e^m*x*e^(m*l
og(x) + 3*n*log(x))/(m + 3*n + 1) + B*a^3*d^2*e^m*x*e^(m*log(x) + 3*n*log(x
))/(m + 3*n + 1) + 3*A*a^2*b*d^2*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n +
 1) + 3*B*a^2*b*c^2*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 3*A*a*b
^2*c^2*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 2*B*a^3*c*d*e^m*x*e^
(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 6*A*a^2*b*c*d*e^m*x*e^(m*log(x) + 2
*n*log(x))/(m + 2*n + 1) + A*a^3*d^2*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2
*n + 1) + B*a^3*c^2*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1) + 3*A*a^2*b*c
^2*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1) + 2*A*a^3*c*d*e^m*x*e^(m*log(x
) + n*log(x))/(m + n + 1) + (e*x)^(m + 1)*A*a^3*c^2/(e*(m + 1))
```

mupad [B] time = 6.35, size = 1882, normalized size = 5.92

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(A + B*x^n)*(a + b*x^n)^3*(c + d*x^n)^2,x)
```

```
[Out] (x*x^(3*n)*(e*x)^m*(A*b^3*c^2 + B*a^3*d^2 + 3*A*a^2*b*d^2 + 3*B*a*b^2*c^2 +
 6*A*a*b^2*c*d + 6*B*a^2*b*c*d)*(5*m + 18*n + 72*m*n + 363*m*n^2 + 108*m^2*
n + 744*m*n^3 + 72*m^3*n + 508*m*n^4 + 18*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 +
 m^5 + 121*n^2 + 372*n^3 + 508*n^4 + 240*n^5 + 363*m^2*n^2 + 372*m^2*n^3 +
121*m^3*n^2 + 1))/(6*m + 21*n + 105*m*n + 700*m*n^2 + 210*m^2*n + 2205*m*n^
3 + 210*m^3*n + 3248*m*n^4 + 105*m^4*n + 1764*m*n^5 + 21*m^5*n + 15*m^2 + 2
0*m^3 + 15*m^4 + 6*m^5 + m^6 + 175*n^2 + 735*n^3 + 1624*n^4 + 1764*n^5 + 72
0*n^6 + 1050*m^2*n^2 + 2205*m^2*n^3 + 700*m^3*n^2 + 1624*m^2*n^4 + 735*m^3*
n^3 + 175*m^4*n^2 + 1) + (A*a^3*c^2*x*(e*x)^m)/(m + 1) + (a*x*x^(2*n)*(e*x)
^m*(A*a^2*d^2 + 3*A*b^2*c^2 + 3*B*a*b*c^2 + 2*B*a^2*c*d + 6*A*a*b*c*d)*(5*m
 + 19*n + 76*m*n + 411*m*n^2 + 114*m^2*n + 922*m*n^3 + 76*m^3*n + 702*m*n^4
 + 19*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 137*n^2 + 461*n^3 + 702*n^4 +
 360*n^5 + 411*m^2*n^2 + 461*m^2*n^3 + 137*m^3*n^2 + 1))/(6*m + 21*n + 105*
m*n + 700*m*n^2 + 210*m^2*n + 2205*m*n^3 + 210*m^3*n + 3248*m*n^4 + 105*m^4
*n + 1764*m*n^5 + 21*m^5*n + 15*m^2 + 20*m^3 + 15*m^4 + 6*m^5 + m^6 + 175*n
^2 + 735*n^3 + 1624*n^4 + 1764*n^5 + 720*n^6 + 1050*m^2*n^2 + 2205*m^2*n^3
+ 700*m^3*n^2 + 1624*m^2*n^4 + 735*m^3*n^3 + 175*m^4*n^2 + 1) + (b*x*x^(4*n
)*(e*x)^m*(3*B*a^2*d^2 + B*b^2*c^2 + 3*A*a*b*d^2 + 2*A*b^2*c*d + 6*B*a*b*c*
d)*(5*m + 17*n + 68*m*n + 321*m*n^2 + 102*m^2*n + 614*m*n^3 + 68*m^3*n + 39
6*m*n^4 + 17*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 107*n^2 + 307*n^3 + 39
6*n^4 + 180*n^5 + 321*m^2*n^2 + 307*m^2*n^3 + 107*m^3*n^2 + 1))/(6*m + 21*n
 + 105*m*n + 700*m*n^2 + 210*m^2*n + 2205*m*n^3 + 210*m^3*n + 3248*m*n^4 +
105*m^4*n + 1764*m*n^5 + 21*m^5*n + 15*m^2 + 20*m^3 + 15*m^4 + 6*m^5 + m^6
+ 175*n^2 + 735*n^3 + 1624*n^4 + 1764*n^5 + 720*n^6 + 1050*m^2*n^2 + 2205*m
^2*n^3 + 700*m^3*n^2 + 1624*m^2*n^4 + 735*m^3*n^3 + 175*m^4*n^2 + 1) + (a^2
*c*x*x^n*(e*x)^m*(2*A*a*d + 3*A*b*c + B*a*c)*(5*m + 20*n + 80*m*n + 465*m*n
```

```
^2 + 120*m^2*n + 1160*m*n^3 + 80*m^3*n + 1044*m*n^4 + 20*m^4*n + 10*m^2 + 1
0*m^3 + 5*m^4 + m^5 + 155*n^2 + 580*n^3 + 1044*n^4 + 720*n^5 + 465*m^2*n^2
+ 580*m^2*n^3 + 155*m^3*n^2 + 1))/(6*m + 21*n + 105*m*n + 700*m*n^2 + 210*m
^2*n + 2205*m*n^3 + 210*m^3*n + 3248*m*n^4 + 105*m^4*n + 1764*m*n^5 + 21*m^
5*n + 15*m^2 + 20*m^3 + 15*m^4 + 6*m^5 + m^6 + 175*n^2 + 735*n^3 + 1624*n^4
 + 1764*n^5 + 720*n^6 + 1050*m^2*n^2 + 2205*m^2*n^3 + 700*m^3*n^2 + 1624*m^
2*n^4 + 735*m^3*n^3 + 175*m^4*n^2 + 1) + (b^2*d*x*x^(5*n)*(e*x)^m*(A*b*d +
3*B*a*d + 2*B*b*c)*(5*m + 16*n + 64*m*n + 285*m*n^2 + 96*m^2*n + 520*m*n^3
+ 64*m^3*n + 324*m*n^4 + 16*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 95*n^2
+ 260*n^3 + 324*n^4 + 144*n^5 + 285*m^2*n^2 + 260*m^2*n^3 + 95*m^3*n^2 + 1)
)/(6*m + 21*n + 105*m*n + 700*m*n^2 + 210*m^2*n + 2205*m*n^3 + 210*m^3*n +
3248*m*n^4 + 105*m^4*n + 1764*m*n^5 + 21*m^5*n + 15*m^2 + 20*m^3 + 15*m^4 +
 6*m^5 + m^6 + 175*n^2 + 735*n^3 + 1624*n^4 + 1764*n^5 + 720*n^6 + 1050*m^2
*n^2 + 2205*m^2*n^3 + 700*m^3*n^2 + 1624*m^2*n^4 + 735*m^3*n^3 + 175*m^4*n^
2 + 1) + (B*b^3*d^2*x*x^(6*n)*(e*x)^m*(5*m + 15*n + 60*m*n + 255*m*n^2 + 90
*m^2*n + 450*m*n^3 + 60*m^3*n + 274*m*n^4 + 15*m^4*n + 10*m^2 + 10*m^3 + 5*
m^4 + m^5 + 85*n^2 + 225*n^3 + 274*n^4 + 120*n^5 + 255*m^2*n^2 + 225*m^2*n^
3 + 85*m^3*n^2 + 1))/(6*m + 21*n + 105*m*n + 700*m*n^2 + 210*m^2*n + 2205*m
*n^3 + 210*m^3*n + 3248*m*n^4 + 105*m^4*n + 1764*m*n^5 + 21*m^5*n + 15*m^2
+ 20*m^3 + 15*m^4 + 6*m^5 + m^6 + 175*n^2 + 735*n^3 + 1624*n^4 + 1764*n^5 +
 720*n^6 + 1050*m^2*n^2 + 2205*m^2*n^3 + 700*m^3*n^2 + 1624*m^2*n^4 + 735*m
^3*n^3 + 175*m^4*n^2 + 1)
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)**m*(a+b*x**n)**3*(A+B*x**n)*(c+d*x**n)**2,x)
[Out] Timed out
```

3.6 $\int (ex)^m (a+bx^n)^2 (A+Bx^n) (c+dx^n)^2 \, dx$

Optimal. Leaf size=237

$$\frac{x^{2n+1}(ex)^m \left(A \left(a^2d^2+4abcd+b^2c^2\right)+2aBc(ad+bc)\right)}{m+2n+1}+\frac{x^{3n+1}(ex)^m \left(a^2Bd^2+2abd(Ad+2Bc)+b^2c(2Ad+Bc\right.}{m+3n+1}$$

Rubi [A] time = 0.31, antiderivative size = 237, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 3, integrand size = 31, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.097, Rules used = {570, 20, 30}

$$\frac{x^{2n+1}(ex)^m \left(A \left(a^2d^2+4abcd+b^2c^2\right)+2aBc(ad+bc)\right)}{m+2n+1}+\frac{x^{3n+1}(ex)^m \left(a^2Bd^2+2abd(Ad+2Bc)+b^2c(2Ad+Bc)\right)}{m+3n+1}+\frac{a^2Ac^2(ex)^{m+1}}{e(m+1)}+\frac{acx^{n+1}(ex)^m(2A(ad+bc)+aBc)}{m+n+1}+\frac{bdx^{4n+1}(ex)^m(2aBd+Abd+2bBc)}{m+4n+1}+\frac{b^2Bd^2x^{5n+1}(ex)^m}{m+5n+1}$$

Antiderivative was successfully verified.

```
[In] Int[(e*x)^m*(a + b*x^n)^2*(A + B*x^n)*(c + d*x^n)^2,x]
```

```
[Out] (a*c*(a*B*c + 2*A*(b*c + a*d))*x^(1 + n)*(e*x)^m)/(1 + m + n) + ((2*a*B*c*(
b*c + a*d) + A*(b^2*c^2 + 4*a*b*c*d + a^2*d^2))*x^(1 + 2*n)*(e*x)^m)/(1 + m
 + 2*n) + ((a^2*B*d^2 + 2*a*b*d*(2*B*c + A*d) + b^2*c*(B*c + 2*A*d))*x^(1 +
 3*n)*(e*x)^m)/(1 + m + 3*n) + (b*d*(2*b*B*c + A*b*d + 2*a*B*d)*x^(1 + 4*n)
*(e*x)^m)/(1 + m + 4*n) + (b^2*B*d^2*x^(1 + 5*n)*(e*x)^m)/(1 + m + 5*n) + (
a^2*A*c^2*(e*x)^(1 + m))/(e*(1 + m))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] &&  !IntegerQ[m] &&  !IntegerQ[n] &&  !
IntegerQ[m + n]
```

Rule 30

```
Int[(x_)^(m_.), x_Symbol] :> Simp[x^(m + 1)/(m + 1), x] /; FreeQ[m, x] && N
eQ[m, -1]
```

Rule 570

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

Rubi steps

$$\begin{aligned}
\int (ex)^m (a+bx^n)^2 (A+Bx^n) (c+dx^n)^2 \, dx &= \int \left(a^2Ac^2(ex)^m+ac(aBc+2A(bc+ad))x^n(ex)^m+\left(2aBc(bc+a\right.\right. \\
&= \frac{a^2Ac^2(ex)^{1+m}}{e(1+m)}+\left(b^2Bd^2\right)\int x^{5n}(ex)^m \, dx+(bd(2bBc+Abd+2a \\
&= \frac{a^2Ac^2(ex)^{1+m}}{e(1+m)}+\left(b^2Bd^2x^{-m}(ex)^m\right)\int x^{m+5n} \, dx+(bd(2bBc+Ab \\
&= \frac{ac(aBc+2A(bc+ad))x^{1+n}(ex)^m}{1+m+n}+\frac{\left(2aBc(bc+ad)+A\left(b^2c^2+\right.\right.}{1+m}
\end{aligned}$$

Mathematica [A] time = 0.60, size = 199, normalized size = 0.84

$$x(ex)^m \left(\frac{x^{2n} \left(A \left(a^2d^2+4abcd+b^2c^2\right)+2aBc(ad+bc)\right)}{m+2n+1}+\frac{x^{3n} \left(a^2Bd^2+2abd(Ad+2Bc)+b^2c(2Ad+Bc)\right)}{m+3n+1}+\frac{a^2Ac^2}{m+1}+\frac{bdx^{4n}(2aBd+Abd+2bBc)}{m+4n+1}+\frac{acx^n(2A(ad+bc)+aBc)}{m+n+1}+\frac{b^2Bd^2x^{5n}}{m+5n+1}\right)$$

Antiderivative was successfully verified.

```
[In] Integrate[(e*x)^m*(a + b*x^n)^2*(A + B*x^n)*(c + d*x^n)^2,x]
```

```
[Out] x*(e*x)^m*((a^2*A*c^2)/(1 + m) + (a*c*(a*B*c + 2*A*(b*c + a*d))*x^n)/(1 + m
 + n) + ((2*a*B*c*(b*c + a*d) + A*(b^2*c^2 + 4*a*b*c*d + a^2*d^2))*x^(2*n))
/(1 + m + 2*n) + ((a^2*B*d^2 + 2*a*b*d*(2*B*c + A*d) + b^2*c*(B*c + 2*A*d))
*x^(3*n))/(1 + m + 3*n) + (b*d*(2*b*B*c + A*b*d + 2*a*B*d)*x^(4*n))/(1 + m
+ 4*n) + (b^2*B*d^2*x^(5*n))/(1 + m + 5*n))
```

IntegrateAlgebraic [F] time = 0.30, size = 0, normalized size = 0.00

$$\int (ex)^m (a+bx^n)^2 (A+Bx^n) (c+dx^n)^2 \, dx$$

Verification is not applicable to the result.

```
[In] IntegrateAlgebraic[(e*x)^m*(a + b*x^n)^2*(A + B*x^n)*(c + d*x^n)^2,x]
```

```
[Out] Defer[IntegrateAlgebraic][(e*x)^m*(a + b*x^n)^2*(A + B*x^n)*(c + d*x^n)^2,
x]
```

fricas [B] time = 0.51, size = 3515, normalized size = 14.83

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)^2*(A+B*x^n)*(c+d*x^n)^2,x, algorithm="fricas")
```

```
[Out] ((B*b^2*d^2*m^5 + 5*B*b^2*d^2*m^4 + 10*B*b^2*d^2*m^3 + 10*B*b^2*d^2*m^2 + 5
*B*b^2*d^2*m + B*b^2*d^2 + 24*(B*b^2*d^2*m + B*b^2*d^2)*n^4 + 50*(B*b^2*d^2
*m^2 + 2*B*b^2*d^2*m + B*b^2*d^2)*n^3 + 35*(B*b^2*d^2*m^3 + 3*B*b^2*d^2*m^2
 + 3*B*b^2*d^2*m + B*b^2*d^2)*n^2 + 10*(B*b^2*d^2*m^4 + 4*B*b^2*d^2*m^3 + 6
*B*b^2*d^2*m^2 + 4*B*b^2*d^2*m + B*b^2*d^2)*n)*x*x^(5*n)*e^(m*log(e) + m*lo
g(x)) + ((2*B*b^2*c*d + (2*B*a*b + A*b^2)*d^2)*m^5 + 2*B*b^2*c*d + 5*(2*B*b
^2*c*d + (2*B*a*b + A*b^2)*d^2)*m^4 + 30*(2*B*b^2*c*d + (2*B*a*b + A*b^2)*d
^2 + (2*B*b^2*c*d + (2*B*a*b + A*b^2)*d^2)*m)*n^4 + 10*(2*B*b^2*c*d + (2*B*
a*b + A*b^2)*d^2)*m^3 + 61*(2*B*b^2*c*d + (2*B*a*b + A*b^2)*d^2 + (2*B*b^2*
c*d + (2*B*a*b + A*b^2)*d^2)*m^2 + 2*(2*B*b^2*c*d + (2*B*a*b + A*b^2)*d^2)*
m)*n^3 + (2*B*a*b + A*b^2)*d^2 + 10*(2*B*b^2*c*d + (2*B*a*b + A*b^2)*d^2)*m
^2 + 41*(2*B*b^2*c*d + (2*B*b^2*c*d + (2*B*a*b + A*b^2)*d^2)*m^3 + (2*B*a*b
 + A*b^2)*d^2 + 3*(2*B*b^2*c*d + (2*B*a*b + A*b^2)*d^2)*m^2 + 3*(2*B*b^2*c*
d + (2*B*a*b + A*b^2)*d^2)*m)*n^2 + 5*(2*B*b^2*c*d + (2*B*a*b + A*b^2)*d^2)
*m + 11*(2*B*b^2*c*d + (2*B*b^2*c*d + (2*B*a*b + A*b^2)*d^2)*m^4 + 4*(2*B*b
^2*c*d + (2*B*a*b + A*b^2)*d^2)*m^3 + (2*B*a*b + A*b^2)*d^2 + 6*(2*B*b^2*c*
d + (2*B*a*b + A*b^2)*d^2)*m^2 + 4*(2*B*b^2*c*d + (2*B*a*b + A*b^2)*d^2)*m)
*n)*x*x^(4*n)*e^(m*log(e) + m*log(x)) + ((B*b^2*c^2 + 2*(2*B*a*b + A*b^2)*c
*d + (B*a^2 + 2*A*a*b)*d^2)*m^5 + B*b^2*c^2 + 5*(B*b^2*c^2 + 2*(2*B*a*b + A
*b^2)*c*d + (B*a^2 + 2*A*a*b)*d^2)*m^4 + 40*(B*b^2*c^2 + 2*(2*B*a*b + A*b^2
)*c*d + (B*a^2 + 2*A*a*b)*d^2 + (B*b^2*c^2 + 2*(2*B*a*b + A*b^2)*c*d + (B*a
^2 + 2*A*a*b)*d^2)*m)*n^4 + 10*(B*b^2*c^2 + 2*(2*B*a*b + A*b^2)*c*d + (B*a^
2 + 2*A*a*b)*d^2)*m^3 + 78*(B*b^2*c^2 + 2*(2*B*a*b + A*b^2)*c*d + (B*a^2 +
2*A*a*b)*d^2 + (B*b^2*c^2 + 2*(2*B*a*b + A*b^2)*c*d + (B*a^2 + 2*A*a*b)*d^2
)*m^2 + 2*(B*b^2*c^2 + 2*(2*B*a*b + A*b^2)*c*d + (B*a^2 + 2*A*a*b)*d^2)*m)*
n^3 + 2*(2*B*a*b + A*b^2)*c*d + (B*a^2 + 2*A*a*b)*d^2 + 10*(B*b^2*c^2 + 2*(
2*B*a*b + A*b^2)*c*d + (B*a^2 + 2*A*a*b)*d^2)*m^2 + 49*(B*b^2*c^2 + (B*b^2*
c^2 + 2*(2*B*a*b + A*b^2)*c*d + (B*a^2 + 2*A*a*b)*d^2)*m^3 + 2*(2*B*a*b + A
*b^2)*c*d + (B*a^2 + 2*A*a*b)*d^2 + 3*(B*b^2*c^2 + 2*(2*B*a*b + A*b^2)*c*d
+ (B*a^2 + 2*A*a*b)*d^2)*m^2 + 3*(B*b^2*c^2 + 2*(2*B*a*b + A*b^2)*c*d + (B*
a^2 + 2*A*a*b)*d^2)*m)*n^2 + 5*(B*b^2*c^2 + 2*(2*B*a*b + A*b^2)*c*d + (B*a^
2 + 2*A*a*b)*d^2)*m + 12*(B*b^2*c^2 + (B*b^2*c^2 + 2*(2*B*a*b + A*b^2)*c*d
+ (B*a^2 + 2*A*a*b)*d^2)*m^4 + 4*(B*b^2*c^2 + 2*(2*B*a*b + A*b^2)*c*d + (B*
```

```
a^2 + 2*A*a*b)*d^2)*m^3 + 2*(2*B*a*b + A*b^2)*c*d + (B*a^2 + 2*A*a*b)*d^2 +
 6*(B*b^2*c^2 + 2*(2*B*a*b + A*b^2)*c*d + (B*a^2 + 2*A*a*b)*d^2)*m^2 + 4*(B
*b^2*c^2 + 2*(2*B*a*b + A*b^2)*c*d + (B*a^2 + 2*A*a*b)*d^2)*m)*n)*x*x^(3*n)
*e^(m*log(e) + m*log(x)) + ((A*a^2*d^2 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 +
 2*A*a*b)*c*d)*m^5 + A*a^2*d^2 + 5*(A*a^2*d^2 + (2*B*a*b + A*b^2)*c^2 + 2*(
B*a^2 + 2*A*a*b)*c*d)*m^4 + 60*(A*a^2*d^2 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^
2 + 2*A*a*b)*c*d + (A*a^2*d^2 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 + 2*A*a*b)
*c*d)*m)*n^4 + 10*(A*a^2*d^2 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 + 2*A*a*b)*
c*d)*m^3 + 107*(A*a^2*d^2 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 + 2*A*a*b)*c*d
 + (A*a^2*d^2 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 + 2*A*a*b)*c*d)*m^2 + 2*(A
*a^2*d^2 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 + 2*A*a*b)*c*d)*m)*n^3 + (2*B*a
*b + A*b^2)*c^2 + 2*(B*a^2 + 2*A*a*b)*c*d + 10*(A*a^2*d^2 + (2*B*a*b + A*b^
2)*c^2 + 2*(B*a^2 + 2*A*a*b)*c*d)*m^2 + 59*(A*a^2*d^2 + (A*a^2*d^2 + (2*B*a
*b + A*b^2)*c^2 + 2*(B*a^2 + 2*A*a*b)*c*d)*m^3 + (2*B*a*b + A*b^2)*c^2 + 2*
(B*a^2 + 2*A*a*b)*c*d + 3*(A*a^2*d^2 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 + 2
*A*a*b)*c*d)*m^2 + 3*(A*a^2*d^2 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 + 2*A*a*
b)*c*d)*m)*n^2 + 5*(A*a^2*d^2 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 + 2*A*a*b)
*c*d)*m + 13*(A*a^2*d^2 + (A*a^2*d^2 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 + 2
*A*a*b)*c*d)*m^4 + 4*(A*a^2*d^2 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 + 2*A*a*
b)*c*d)*m^3 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 + 2*A*a*b)*c*d + 6*(A*a^2*d^
2 + (2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 + 2*A*a*b)*c*d)*m^2 + 4*(A*a^2*d^2 + (
2*B*a*b + A*b^2)*c^2 + 2*(B*a^2 + 2*A*a*b)*c*d)*m)*n)*x*x^(2*n)*e^(m*log(e)
 + m*log(x)) + ((2*A*a^2*c*d + (B*a^2 + 2*A*a*b)*c^2)*m^5 + 2*A*a^2*c*d + 5
*(2*A*a^2*c*d + (B*a^2 + 2*A*a*b)*c^2)*m^4 + 120*(2*A*a^2*c*d + (B*a^2 + 2*
A*a*b)*c^2 + (2*A*a^2*c*d + (B*a^2 + 2*A*a*b)*c^2)*m)*n^4 + 10*(2*A*a^2*c*d
 + (B*a^2 + 2*A*a*b)*c^2)*m^3 + 154*(2*A*a^2*c*d + (B*a^2 + 2*A*a*b)*c^2 +
(2*A*a^2*c*d + (B*a^2 + 2*A*a*b)*c^2)*m^2 + 2*(2*A*a^2*c*d + (B*a^2 + 2*A*a
*b)*c^2)*m)*n^3 + (B*a^2 + 2*A*a*b)*c^2 + 10*(2*A*a^2*c*d + (B*a^2 + 2*A*a*
b)*c^2)*m^2 + 71*(2*A*a^2*c*d + (2*A*a^2*c*d + (B*a^2 + 2*A*a*b)*c^2)*m^3 +
 (B*a^2 + 2*A*a*b)*c^2 + 3*(2*A*a^2*c*d + (B*a^2 + 2*A*a*b)*c^2)*m^2 + 3*(2
*A*a^2*c*d + (B*a^2 + 2*A*a*b)*c^2)*m)*n^2 + 5*(2*A*a^2*c*d + (B*a^2 + 2*A*
a*b)*c^2)*m + 14*(2*A*a^2*c*d + (2*A*a^2*c*d + (B*a^2 + 2*A*a*b)*c^2)*m^4 +
 4*(2*A*a^2*c*d + (B*a^2 + 2*A*a*b)*c^2)*m^3 + (B*a^2 + 2*A*a*b)*c^2 + 6*(2
*A*a^2*c*d + (B*a^2 + 2*A*a*b)*c^2)*m^2 + 4*(2*A*a^2*c*d + (B*a^2 + 2*A*a*b
)*c^2)*m)*n)*x*x^n*e^(m*log(e) + m*log(x)) + (A*a^2*c^2*m^5 + 120*A*a^2*c^2
*n^5 + 5*A*a^2*c^2*m^4 + 10*A*a^2*c^2*m^3 + 10*A*a^2*c^2*m^2 + 5*A*a^2*c^2*
m + A*a^2*c^2 + 274*(A*a^2*c^2*m + A*a^2*c^2)*n^4 + 225*(A*a^2*c^2*m^2 + 2*
A*a^2*c^2*m + A*a^2*c^2)*n^3 + 85*(A*a^2*c^2*m^3 + 3*A*a^2*c^2*m^2 + 3*A*a^
2*c^2*m + A*a^2*c^2)*n^2 + 15*(A*a^2*c^2*m^4 + 4*A*a^2*c^2*m^3 + 6*A*a^2*c^
2*m^2 + 4*A*a^2*c^2*m + A*a^2*c^2)*n)*x*e^(m*log(e) + m*log(x)))/(m^6 + 120
*(m + 1)*n^5 + 6*m^5 + 274*(m^2 + 2*m + 1)*n^4 + 15*m^4 + 225*(m^3 + 3*m^2
+ 3*m + 1)*n^3 + 20*m^3 + 85*(m^4 + 4*m^3 + 6*m^2 + 4*m + 1)*n^2 + 15*m^2 +
 15*(m^5 + 5*m^4 + 10*m^3 + 10*m^2 + 5*m + 1)*n + 6*m + 1)
```

giac [B] time = 2.01, size = 8103, normalized size = 34.19

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)^2*(A+B*x^n)*(c+d*x^n)^2,x, algorithm="giac")
```

```
[Out] (B*b^2*d^2*m^5*x*x^m*x^(5*n)*e^m + 10*B*b^2*d^2*m^4*n*x*x^m*x^(5*n)*e^m + 3
5*B*b^2*d^2*m^3*n^2*x*x^m*x^(5*n)*e^m + 50*B*b^2*d^2*m^2*n^3*x*x^m*x^(5*n)*
e^m + 24*B*b^2*d^2*m*n^4*x*x^m*x^(5*n)*e^m + 2*B*b^2*c*d*m^5*x*x^m*x^(4*n)*
e^m + 2*B*a*b*d^2*m^5*x*x^m*x^(4*n)*e^m + A*b^2*d^2*m^5*x*x^m*x^(4*n)*e^m +
 22*B*b^2*c*d*m^4*n*x*x^m*x^(4*n)*e^m + 22*B*a*b*d^2*m^4*n*x*x^m*x^(4*n)*e^
m + 11*A*b^2*d^2*m^4*n*x*x^m*x^(4*n)*e^m + 82*B*b^2*c*d*m^3*n^2*x*x^m*x^(4*
n)*e^m + 82*B*a*b*d^2*m^3*n^2*x*x^m*x^(4*n)*e^m + 41*A*b^2*d^2*m^3*n^2*x*x^
m*x^(4*n)*e^m + 122*B*b^2*c*d*m^2*n^3*x*x^m*x^(4*n)*e^m + 122*B*a*b*d^2*m^2
*n^3*x*x^m*x^(4*n)*e^m + 61*A*b^2*d^2*m^2*n^3*x*x^m*x^(4*n)*e^m + 60*B*b^2*
```

```
c*d*m*n^4*x*x^m*x^(4*n)*e^m + 60*B*a*b*d^2*m*n^4*x*x^m*x^(4*n)*e^m + 30*A*b
^2*d^2*m*n^4*x*x^m*x^(4*n)*e^m + B*b^2*c^2*m^5*x*x^m*x^(3*n)*e^m + 4*B*a*b*
c*d*m^5*x*x^m*x^(3*n)*e^m + 2*A*b^2*c*d*m^5*x*x^m*x^(3*n)*e^m + B*a^2*d^2*m
^5*x*x^m*x^(3*n)*e^m + 2*A*a*b*d^2*m^5*x*x^m*x^(3*n)*e^m + 12*B*b^2*c^2*m^4
*n*x*x^m*x^(3*n)*e^m + 48*B*a*b*c*d*m^4*n*x*x^m*x^(3*n)*e^m + 24*A*b^2*c*d*
m^4*n*x*x^m*x^(3*n)*e^m + 12*B*a^2*d^2*m^4*n*x*x^m*x^(3*n)*e^m + 24*A*a*b*d
^2*m^4*n*x*x^m*x^(3*n)*e^m + 49*B*b^2*c^2*m^3*n^2*x*x^m*x^(3*n)*e^m + 196*B
*a*b*c*d*m^3*n^2*x*x^m*x^(3*n)*e^m + 98*A*b^2*c*d*m^3*n^2*x*x^m*x^(3*n)*e^m
 + 49*B*a^2*d^2*m^3*n^2*x*x^m*x^(3*n)*e^m + 98*A*a*b*d^2*m^3*n^2*x*x^m*x^(3
*n)*e^m + 78*B*b^2*c^2*m^2*n^3*x*x^m*x^(3*n)*e^m + 312*B*a*b*c*d*m^2*n^3*x*
x^m*x^(3*n)*e^m + 156*A*b^2*c*d*m^2*n^3*x*x^m*x^(3*n)*e^m + 78*B*a^2*d^2*m^
2*n^3*x*x^m*x^(3*n)*e^m + 156*A*a*b*d^2*m^2*n^3*x*x^m*x^(3*n)*e^m + 40*B*b^
2*c^2*m*n^4*x*x^m*x^(3*n)*e^m + 160*B*a*b*c*d*m*n^4*x*x^m*x^(3*n)*e^m + 80*
A*b^2*c*d*m*n^4*x*x^m*x^(3*n)*e^m + 40*B*a^2*d^2*m*n^4*x*x^m*x^(3*n)*e^m +
80*A*a*b*d^2*m*n^4*x*x^m*x^(3*n)*e^m + 2*B*a*b*c^2*m^5*x*x^m*x^(2*n)*e^m +
A*b^2*c^2*m^5*x*x^m*x^(2*n)*e^m + 2*B*a^2*c*d*m^5*x*x^m*x^(2*n)*e^m + 4*A*a
*b*c*d*m^5*x*x^m*x^(2*n)*e^m + A*a^2*d^2*m^5*x*x^m*x^(2*n)*e^m + 26*B*a*b*c
^2*m^4*n*x*x^m*x^(2*n)*e^m + 13*A*b^2*c^2*m^4*n*x*x^m*x^(2*n)*e^m + 26*B*a^
2*c*d*m^4*n*x*x^m*x^(2*n)*e^m + 52*A*a*b*c*d*m^4*n*x*x^m*x^(2*n)*e^m + 13*A
*a^2*d^2*m^4*n*x*x^m*x^(2*n)*e^m + 118*B*a*b*c^2*m^3*n^2*x*x^m*x^(2*n)*e^m
 + 59*A*b^2*c^2*m^3*n^2*x*x^m*x^(2*n)*e^m + 118*B*a^2*c*d*m^3*n^2*x*x^m*x^(2
*n)*e^m + 236*A*a*b*c*d*m^3*n^2*x*x^m*x^(2*n)*e^m + 59*A*a^2*d^2*m^3*n^2*x*
x^m*x^(2*n)*e^m + 214*B*a*b*c^2*m^2*n^3*x*x^m*x^(2*n)*e^m + 107*A*b^2*c^2*m
^2*n^3*x*x^m*x^(2*n)*e^m + 214*B*a^2*c*d*m^2*n^3*x*x^m*x^(2*n)*e^m + 428*A*
a*b*c*d*m^2*n^3*x*x^m*x^(2*n)*e^m + 107*A*a^2*d^2*m^2*n^3*x*x^m*x^(2*n)*e^m
 + 120*B*a*b*c^2*m*n^4*x*x^m*x^(2*n)*e^m + 60*A*b^2*c^2*m*n^4*x*x^m*x^(2*n)
*e^m + 120*B*a^2*c*d*m*n^4*x*x^m*x^(2*n)*e^m + 240*A*a*b*c*d*m*n^4*x*x^m*x^
(2*n)*e^m + 60*A*a^2*d^2*m*n^4*x*x^m*x^(2*n)*e^m + B*a^2*c^2*m^5*x*x^m*x^n*
e^m + 2*A*a*b*c^2*m^5*x*x^m*x^n*e^m + 2*A*a^2*c*d*m^5*x*x^m*x^n*e^m + 14*B*
a^2*c^2*m^4*n*x*x^m*x^n*e^m + 28*A*a*b*c^2*m^4*n*x*x^m*x^n*e^m + 28*A*a^2*c
*d*m^4*n*x*x^m*x^n*e^m + 71*B*a^2*c^2*m^3*n^2*x*x^m*x^n*e^m + 142*A*a*b*c^2
*m^3*n^2*x*x^m*x^n*e^m + 142*A*a^2*c*d*m^3*n^2*x*x^m*x^n*e^m + 154*B*a^2*c^
2*m^2*n^3*x*x^m*x^n*e^m + 308*A*a*b*c^2*m^2*n^3*x*x^m*x^n*e^m + 308*A*a^2*c
*d*m^2*n^3*x*x^m*x^n*e^m + 120*B*a^2*c^2*m*n^4*x*x^m*x^n*e^m + 240*A*a*b*c^
2*m*n^4*x*x^m*x^n*e^m + 240*A*a^2*c*d*m*n^4*x*x^m*x^n*e^m + A*a^2*c^2*m^5*x
*x^m*e^m + 15*A*a^2*c^2*m^4*n*x*x^m*e^m + 85*A*a^2*c^2*m^3*n^2*x*x^m*e^m +
225*A*a^2*c^2*m^2*n^3*x*x^m*e^m + 274*A*a^2*c^2*m*n^4*x*x^m*e^m + 120*A*a^2
*c^2*n^5*x*x^m*e^m + 5*B*b^2*d^2*m^4*x*x^m*x^(5*n)*e^m + 40*B*b^2*d^2*m^3*n
*x*x^m*x^(5*n)*e^m + 105*B*b^2*d^2*m^2*n^2*x*x^m*x^(5*n)*e^m + 100*B*b^2*d^
2*m*n^3*x*x^m*x^(5*n)*e^m + 24*B*b^2*d^2*n^4*x*x^m*x^(5*n)*e^m + 10*B*b^2*c
*d*m^4*x*x^m*x^(4*n)*e^m + 10*B*a*b*d^2*m^4*x*x^m*x^(4*n)*e^m + 5*A*b^2*d^2
*m^4*x*x^m*x^(4*n)*e^m + 88*B*b^2*c*d*m^3*n*x*x^m*x^(4*n)*e^m + 88*B*a*b*d^
2*m^3*n*x*x^m*x^(4*n)*e^m + 44*A*b^2*d^2*m^3*n*x*x^m*x^(4*n)*e^m + 246*B*b^
2*c*d*m^2*n^2*x*x^m*x^(4*n)*e^m + 246*B*a*b*d^2*m^2*n^2*x*x^m*x^(4*n)*e^m +
 123*A*b^2*d^2*m^2*n^2*x*x^m*x^(4*n)*e^m + 244*B*b^2*c*d*m*n^3*x*x^m*x^(4*n
)*e^m + 244*B*a*b*d^2*m*n^3*x*x^m*x^(4*n)*e^m + 122*A*b^2*d^2*m*n^3*x*x^m*x
^(4*n)*e^m + 60*B*b^2*c*d*n^4*x*x^m*x^(4*n)*e^m + 60*B*a*b*d^2*n^4*x*x^m*x^
(4*n)*e^m + 30*A*b^2*d^2*n^4*x*x^m*x^(4*n)*e^m + 5*B*b^2*c^2*m^4*x*x^m*x^(3
*n)*e^m + 20*B*a*b*c*d*m^4*x*x^m*x^(3*n)*e^m + 10*A*b^2*c*d*m^4*x*x^m*x^(3*
n)*e^m + 5*B*a^2*d^2*m^4*x*x^m*x^(3*n)*e^m + 10*A*a*b*d^2*m^4*x*x^m*x^(3*n)
*e^m + 48*B*b^2*c^2*m^3*n*x*x^m*x^(3*n)*e^m + 192*B*a*b*c*d*m^3*n*x*x^m*x^(
3*n)*e^m + 96*A*b^2*c*d*m^3*n*x*x^m*x^(3*n)*e^m + 48*B*a^2*d^2*m^3*n*x*x^m*
x^(3*n)*e^m + 96*A*a*b*d^2*m^3*n*x*x^m*x^(3*n)*e^m + 147*B*b^2*c^2*m^2*n^2*
x*x^m*x^(3*n)*e^m + 588*B*a*b*c*d*m^2*n^2*x*x^m*x^(3*n)*e^m + 294*A*b^2*c*d
*m^2*n^2*x*x^m*x^(3*n)*e^m + 147*B*a^2*d^2*m^2*n^2*x*x^m*x^(3*n)*e^m + 294*
A*a*b*d^2*m^2*n^2*x*x^m*x^(3*n)*e^m + 156*B*b^2*c^2*m*n^3*x*x^m*x^(3*n)*e^m
 + 624*B*a*b*c*d*m*n^3*x*x^m*x^(3*n)*e^m + 312*A*b^2*c*d*m*n^3*x*x^m*x^(3*n
)*e^m + 156*B*a^2*d^2*m*n^3*x*x^m*x^(3*n)*e^m + 312*A*a*b*d^2*m*n^3*x*x^m*x
^(3*n)*e^m + 40*B*b^2*c^2*n^4*x*x^m*x^(3*n)*e^m + 160*B*a*b*c*d*n^4*x*x^m*x
```

```
^(3*n)*e^m + 80*A*b^2*c*d*n^4*x*x^m*x^(3*n)*e^m + 40*B*a^2*d^2*n^4*x*x^m*x^
(3*n)*e^m + 80*A*a*b*d^2*n^4*x*x^m*x^(3*n)*e^m + 10*B*a*b*c^2*m^4*x*x^m*x^(
2*n)*e^m + 5*A*b^2*c^2*m^4*x*x^m*x^(2*n)*e^m + 10*B*a^2*c*d*m^4*x*x^m*x^(2*
n)*e^m + 20*A*a*b*c*d*m^4*x*x^m*x^(2*n)*e^m + 5*A*a^2*d^2*m^4*x*x^m*x^(2*n)
*e^m + 104*B*a*b*c^2*m^3*n*x*x^m*x^(2*n)*e^m + 52*A*b^2*c^2*m^3*n*x*x^m*x^(
2*n)*e^m + 104*B*a^2*c*d*m^3*n*x*x^m*x^(2*n)*e^m + 208*A*a*b*c*d*m^3*n*x*x^
m*x^(2*n)*e^m + 52*A*a^2*d^2*m^3*n*x*x^m*x^(2*n)*e^m + 354*B*a*b*c^2*m^2*n^
2*x*x^m*x^(2*n)*e^m + 177*A*b^2*c^2*m^2*n^2*x*x^m*x^(2*n)*e^m + 354*B*a^2*c
*d*m^2*n^2*x*x^m*x^(2*n)*e^m + 708*A*a*b*c*d*m^2*n^2*x*x^m*x^(2*n)*e^m + 17
7*A*a^2*d^2*m^2*n^2*x*x^m*x^(2*n)*e^m + 428*B*a*b*c^2*m*n^3*x*x^m*x^(2*n)*e
^m + 214*A*b^2*c^2*m*n^3*x*x^m*x^(2*n)*e^m + 428*B*a^2*c*d*m*n^3*x*x^m*x^(2
*n)*e^m + 856*A*a*b*c*d*m*n^3*x*x^m*x^(2*n)*e^m + 214*A*a^2*d^2*m*n^3*x*x^m
*x^(2*n)*e^m + 120*B*a*b*c^2*n^4*x*x^m*x^(2*n)*e^m + 60*A*b^2*c^2*n^4*x*x^m
*x^(2*n)*e^m + 120*B*a^2*c*d*n^4*x*x^m*x^(2*n)*e^m + 240*A*a*b*c*d*n^4*x*x^
m*x^(2*n)*e^m + 60*A*a^2*d^2*n^4*x*x^m*x^(2*n)*e^m + 5*B*a^2*c^2*m^4*x*x^m*
x^n*e^m + 10*A*a*b*c^2*m^4*x*x^m*x^n*e^m + 10*A*a^2*c*d*m^4*x*x^m*x^n*e^m +
 56*B*a^2*c^2*m^3*n*x*x^m*x^n*e^m + 112*A*a*b*c^2*m^3*n*x*x^m*x^n*e^m + 112
*A*a^2*c*d*m^3*n*x*x^m*x^n*e^m + 213*B*a^2*c^2*m^2*n^2*x*x^m*x^n*e^m + 426*
A*a*b*c^2*m^2*n^2*x*x^m*x^n*e^m + 426*A*a^2*c*d*m^2*n^2*x*x^m*x^n*e^m + 308
*B*a^2*c^2*m*n^3*x*x^m*x^n*e^m + 616*A*a*b*c^2*m*n^3*x*x^m*x^n*e^m + 616*A*
a^2*c*d*m*n^3*x*x^m*x^n*e^m + 120*B*a^2*c^2*n^4*x*x^m*x^n*e^m + 240*A*a*b*c
^2*n^4*x*x^m*x^n*e^m + 240*A*a^2*c*d*n^4*x*x^m*x^n*e^m + 5*A*a^2*c^2*m^4*x*
x^m*e^m + 60*A*a^2*c^2*m^3*n*x*x^m*e^m + 255*A*a^2*c^2*m^2*n^2*x*x^m*e^m +
450*A*a^2*c^2*m*n^3*x*x^m*e^m + 274*A*a^2*c^2*n^4*x*x^m*e^m + 10*B*b^2*d^2*
m^3*x*x^m*x^(5*n)*e^m + 60*B*b^2*d^2*m^2*n*x*x^m*x^(5*n)*e^m + 105*B*b^2*d^
2*m*n^2*x*x^m*x^(5*n)*e^m + 50*B*b^2*d^2*n^3*x*x^m*x^(5*n)*e^m + 20*B*b^2*c
*d*m^3*x*x^m*x^(4*n)*e^m + 20*B*a*b*d^2*m^3*x*x^m*x^(4*n)*e^m + 10*A*b^2*d^
2*m^3*x*x^m*x^(4*n)*e^m + 132*B*b^2*c*d*m^2*n*x*x^m*x^(4*n)*e^m + 132*B*a*b
*d^2*m^2*n*x*x^m*x^(4*n)*e^m + 66*A*b^2*d^2*m^2*n*x*x^m*x^(4*n)*e^m + 246*B
*b^2*c*d*m*n^2*x*x^m*x^(4*n)*e^m + 246*B*a*b*d^2*m*n^2*x*x^m*x^(4*n)*e^m +
123*A*b^2*d^2*m*n^2*x*x^m*x^(4*n)*e^m + 122*B*b^2*c*d*n^3*x*x^m*x^(4*n)*e^m
 + 122*B*a*b*d^2*n^3*x*x^m*x^(4*n)*e^m + 61*A*b^2*d^2*n^3*x*x^m*x^(4*n)*e^m
 + 10*B*b^2*c^2*m^3*x*x^m*x^(3*n)*e^m + 40*B*a*b*c*d*m^3*x*x^m*x^(3*n)*e^m
+ 20*A*b^2*c*d*m^3*x*x^m*x^(3*n)*e^m + 10*B*a^2*d^2*m^3*x*x^m*x^(3*n)*e^m +
 20*A*a*b*d^2*m^3*x*x^m*x^(3*n)*e^m + 72*B*b^2*c^2*m^2*n*x*x^m*x^(3*n)*e^m
+ 288*B*a*b*c*d*m^2*n*x*x^m*x^(3*n)*e^m + 144*A*b^2*c*d*m^2*n*x*x^m*x^(3*n)
*e^m + 72*B*a^2*d^2*m^2*n*x*x^m*x^(3*n)*e^m + 144*A*a*b*d^2*m^2*n*x*x^m*x^(
3*n)*e^m + 147*B*b^2*c^2*m*n^2*x*x^m*x^(3*n)*e^m + 588*B*a*b*c*d*m*n^2*x*x^
m*x^(3*n)*e^m + 294*A*b^2*c*d*m*n^2*x*x^m*x^(3*n)*e^m + 147*B*a^2*d^2*m*n^2
*x*x^m*x^(3*n)*e^m + 294*A*a*b*d^2*m*n^2*x*x^m*x^(3*n)*e^m + 78*B*b^2*c^2*n
^3*x*x^m*x^(3*n)*e^m + 312*B*a*b*c*d*n^3*x*x^m*x^(3*n)*e^m + 156*A*b^2*c*d*
n^3*x*x^m*x^(3*n)*e^m + 78*B*a^2*d^2*n^3*x*x^m*x^(3*n)*e^m + 156*A*a*b*d^2*
n^3*x*x^m*x^(3*n)*e^m + 20*B*a*b*c^2*m^3*x*x^m*x^(2*n)*e^m + 10*A*b^2*c^2*m
^3*x*x^m*x^(2*n)*e^m + 20*B*a^2*c*d*m^3*x*x^m*x^(2*n)*e^m + 40*A*a*b*c*d*m^
3*x*x^m*x^(2*n)*e^m + 10*A*a^2*d^2*m^3*x*x^m*x^(2*n)*e^m + 156*B*a*b*c^2*m^
2*n*x*x^m*x^(2*n)*e^m + 78*A*b^2*c^2*m^2*n*x*x^m*x^(2*n)*e^m + 156*B*a^2*c*
d*m^2*n*x*x^m*x^(2*n)*e^m + 312*A*a*b*c*d*m^2*n*x*x^m*x^(2*n)*e^m + 78*A*a^
2*d^2*m^2*n*x*x^m*x^(2*n)*e^m + 354*B*a*b*c^2*m*n^2*x*x^m*x^(2*n)*e^m + 177
*A*b^2*c^2*m*n^2*x*x^m*x^(2*n)*e^m + 354*B*a^2*c*d*m*n^2*x*x^m*x^(2*n)*e^m
+ 708*A*a*b*c*d*m*n^2*x*x^m*x^(2*n)*e^m + 177*A*a^2*d^2*m*n^2*x*x^m*x^(2*n)
*e^m + 214*B*a*b*c^2*n^3*x*x^m*x^(2*n)*e^m + 107*A*b^2*c^2*n^3*x*x^m*x^(2*n
)*e^m + 214*B*a^2*c*d*n^3*x*x^m*x^(2*n)*e^m + 428*A*a*b*c*d*n^3*x*x^m*x^(2*
n)*e^m + 107*A*a^2*d^2*n^3*x*x^m*x^(2*n)*e^m + 10*B*a^2*c^2*m^3*x*x^m*x^n*e
^m + 20*A*a*b*c^2*m^3*x*x^m*x^n*e^m + 20*A*a^2*c*d*m^3*x*x^m*x^n*e^m + 84*B
*a^2*c^2*m^2*n*x*x^m*x^n*e^m + 168*A*a*b*c^2*m^2*n*x*x^m*x^n*e^m + 168*A*a^
2*c*d*m^2*n*x*x^m*x^n*e^m + 213*B*a^2*c^2*m*n^2*x*x^m*x^n*e^m + 426*A*a*b*c
^2*m*n^2*x*x^m*x^n*e^m + 426*A*a^2*c*d*m*n^2*x*x^m*x^n*e^m + 154*B*a^2*c^2*
n^3*x*x^m*x^n*e^m + 308*A*a*b*c^2*n^3*x*x^m*x^n*e^m + 308*A*a^2*c*d*n^3*x*x
^m*x^n*e^m + 10*A*a^2*c^2*m^3*x*x^m*e^m + 90*A*a^2*c^2*m^2*n*x*x^m*e^m + 25
```

```
5*A*a^2*c^2*m*n^2*x*x^m*e^m + 225*A*a^2*c^2*n^3*x*x^m*e^m + 10*B*b^2*d^2*m^
2*x*x^m*x^(5*n)*e^m + 40*B*b^2*d^2*m*n*x*x^m*x^(5*n)*e^m + 35*B*b^2*d^2*n^2
*x*x^m*x^(5*n)*e^m + 20*B*b^2*c*d*m^2*x*x^m*x^(4*n)*e^m + 20*B*a*b*d^2*m^2*
x*x^m*x^(4*n)*e^m + 10*A*b^2*d^2*m^2*x*x^m*x^(4*n)*e^m + 88*B*b^2*c*d*m*n*x
*x^m*x^(4*n)*e^m + 88*B*a*b*d^2*m*n*x*x^m*x^(4*n)*e^m + 44*A*b^2*d^2*m*n*x*
x^m*x^(4*n)*e^m + 82*B*b^2*c*d*n^2*x*x^m*x^(4*n)*e^m + 82*B*a*b*d^2*n^2*x*x
^m*x^(4*n)*e^m + 41*A*b^2*d^2*n^2*x*x^m*x^(4*n)*e^m + 10*B*b^2*c^2*m^2*x*x^
m*x^(3*n)*e^m + 40*B*a*b*c*d*m^2*x*x^m*x^(3*n)*e^m + 20*A*b^2*c*d*m^2*x*x^m
*x^(3*n)*e^m + 10*B*a^2*d^2*m^2*x*x^m*x^(3*n)*e^m + 20*A*a*b*d^2*m^2*x*x^m*
x^(3*n)*e^m + 48*B*b^2*c^2*m*n*x*x^m*x^(3*n)*e^m + 192*B*a*b*c*d*m*n*x*x^m*
x^(3*n)*e^m + 96*A*b^2*c*d*m*n*x*x^m*x^(3*n)*e^m + 48*B*a^2*d^2*m*n*x*x^m*x
^(3*n)*e^m + 96*A*a*b*d^2*m*n*x*x^m*x^(3*n)*e^m + 49*B*b^2*c^2*n^2*x*x^m*x^
(3*n)*e^m + 196*B*a*b*c*d*n^2*x*x^m*x^(3*n)*e^m + 98*A*b^2*c*d*n^2*x*x^m*x^
(3*n)*e^m + 49*B*a^2*d^2*n^2*x*x^m*x^(3*n)*e^m + 98*A*a*b*d^2*n^2*x*x^m*x^(
3*n)*e^m + 20*B*a*b*c^2*m^2*x*x^m*x^(2*n)*e^m + 10*A*b^2*c^2*m^2*x*x^m*x^(2
*n)*e^m + 20*B*a^2*c*d*m^2*x*x^m*x^(2*n)*e^m + 40*A*a*b*c*d*m^2*x*x^m*x^(2*
n)*e^m + 10*A*a^2*d^2*m^2*x*x^m*x^(2*n)*e^m + 104*B*a*b*c^2*m*n*x*x^m*x^(2*
n)*e^m + 52*A*b^2*c^2*m*n*x*x^m*x^(2*n)*e^m + 104*B*a^2*c*d*m*n*x*x^m*x^(2*
n)*e^m + 208*A*a*b*c*d*m*n*x*x^m*x^(2*n)*e^m + 52*A*a^2*d^2*m*n*x*x^m*x^(2*
n)*e^m + 118*B*a*b*c^2*n^2*x*x^m*x^(2*n)*e^m + 59*A*b^2*c^2*n^2*x*x^m*x^(2*
n)*e^m + 118*B*a^2*c*d*n^2*x*x^m*x^(2*n)*e^m + 236*A*a*b*c*d*n^2*x*x^m*x^(2
*n)*e^m + 59*A*a^2*d^2*n^2*x*x^m*x^(2*n)*e^m + 10*B*a^2*c^2*m^2*x*x^m*x^n*e
^m + 20*A*a*b*c^2*m^2*x*x^m*x^n*e^m + 20*A*a^2*c*d*m^2*x*x^m*x^n*e^m + 56*B
*a^2*c^2*m*n*x*x^m*x^n*e^m + 112*A*a*b*c^2*m*n*x*x^m*x^n*e^m + 112*A*a^2*c*
d*m*n*x*x^m*x^n*e^m + 71*B*a^2*c^2*n^2*x*x^m*x^n*e^m + 142*A*a*b*c^2*n^2*x*
x^m*x^n*e^m + 142*A*a^2*c*d*n^2*x*x^m*x^n*e^m + 10*A*a^2*c^2*m^2*x*x^m*e^m
+ 60*A*a^2*c^2*m*n*x*x^m*e^m + 85*A*a^2*c^2*n^2*x*x^m*e^m + 5*B*b^2*d^2*m*x
*x^m*x^(5*n)*e^m + 10*B*b^2*d^2*n*x*x^m*x^(5*n)*e^m + 10*B*b^2*c*d*m*x*x^m*
x^(4*n)*e^m + 10*B*a*b*d^2*m*x*x^m*x^(4*n)*e^m + 5*A*b^2*d^2*m*x*x^m*x^(4*n
)*e^m + 22*B*b^2*c*d*n*x*x^m*x^(4*n)*e^m + 22*B*a*b*d^2*n*x*x^m*x^(4*n)*e^m
 + 11*A*b^2*d^2*n*x*x^m*x^(4*n)*e^m + 5*B*b^2*c^2*m*x*x^m*x^(3*n)*e^m + 20*
B*a*b*c*d*m*x*x^m*x^(3*n)*e^m + 10*A*b^2*c*d*m*x*x^m*x^(3*n)*e^m + 5*B*a^2*
d^2*m*x*x^m*x^(3*n)*e^m + 10*A*a*b*d^2*m*x*x^m*x^(3*n)*e^m + 12*B*b^2*c^2*n
*x*x^m*x^(3*n)*e^m + 48*B*a*b*c*d*n*x*x^m*x^(3*n)*e^m + 24*A*b^2*c*d*n*x*x^
m*x^(3*n)*e^m + 12*B*a^2*d^2*n*x*x^m*x^(3*n)*e^m + 24*A*a*b*d^2*n*x*x^m*x^(
3*n)*e^m + 10*B*a*b*c^2*m*x*x^m*x^(2*n)*e^m + 5*A*b^2*c^2*m*x*x^m*x^(2*n)*e
^m + 10*B*a^2*c*d*m*x*x^m*x^(2*n)*e^m + 20*A*a*b*c*d*m*x*x^m*x^(2*n)*e^m +
5*A*a^2*d^2*m*x*x^m*x^(2*n)*e^m + 26*B*a*b*c^2*n*x*x^m*x^(2*n)*e^m + 13*A*b
^2*c^2*n*x*x^m*x^(2*n)*e^m + 26*B*a^2*c*d*n*x*x^m*x^(2*n)*e^m + 52*A*a*b*c*
d*n*x*x^m*x^(2*n)*e^m + 13*A*a^2*d^2*n*x*x^m*x^(2*n)*e^m + 5*B*a^2*c^2*m*x*
x^m*x^n*e^m + 10*A*a*b*c^2*m*x*x^m*x^n*e^m + 10*A*a^2*c*d*m*x*x^m*x^n*e^m +
 14*B*a^2*c^2*n*x*x^m*x^n*e^m + 28*A*a*b*c^2*n*x*x^m*x^n*e^m + 28*A*a^2*c*d
*n*x*x^m*x^n*e^m + 5*A*a^2*c^2*m*x*x^m*e^m + 15*A*a^2*c^2*n*x*x^m*e^m + B*b
^2*d^2*x*x^m*x^(5*n)*e^m + 2*B*b^2*c*d*x*x^m*x^(4*n)*e^m + 2*B*a*b*d^2*x*x^
m*x^(4*n)*e^m + A*b^2*d^2*x*x^m*x^(4*n)*e^m + B*b^2*c^2*x*x^m*x^(3*n)*e^m +
 4*B*a*b*c*d*x*x^m*x^(3*n)*e^m + 2*A*b^2*c*d*x*x^m*x^(3*n)*e^m + B*a^2*d^2*
x*x^m*x^(3*n)*e^m + 2*A*a*b*d^2*x*x^m*x^(3*n)*e^m + 2*B*a*b*c^2*x*x^m*x^(2*
n)*e^m + A*b^2*c^2*x*x^m*x^(2*n)*e^m + 2*B*a^2*c*d*x*x^m*x^(2*n)*e^m + 4*A*
a*b*c*d*x*x^m*x^(2*n)*e^m + A*a^2*d^2*x*x^m*x^(2*n)*e^m + B*a^2*c^2*x*x^m*x
^n*e^m + 2*A*a*b*c^2*x*x^m*x^n*e^m + 2*A*a^2*c*d*x*x^m*x^n*e^m + A*a^2*c^2*
x*x^m*e^m)/(m^6 + 15*m^5*n + 85*m^4*n^2 + 225*m^3*n^3 + 274*m^2*n^4 + 120*m
*n^5 + 6*m^5 + 75*m^4*n + 340*m^3*n^2 + 675*m^2*n^3 + 548*m*n^4 + 120*n^5 +
 15*m^4 + 150*m^3*n + 510*m^2*n^2 + 675*m*n^3 + 274*n^4 + 20*m^3 + 150*m^2*
n + 340*m*n^2 + 225*n^3 + 15*m^2 + 75*m*n + 85*n^2 + 6*m + 15*n + 1)
```

maple [C] time = 0.17, size = 5908, normalized size = 24.93

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(b*x^n+a)^2*(B*x^n+A)*(d*x^n+c)^2,x)
```

```
[Out] result too large to display
```

maxima [B] time = 0.91, size = 540, normalized size = 2.28

$$\frac{Bb^2d^2e^m e^{(m\log(x)+5n\log(x))}}{m+5n+1}+\frac{2Bb^2cde^m e^{(m\log(x)+4n\log(x))}}{m+4n+1}+\frac{2Babd^2e^m e^{(m\log(x)+4n\log(x))}}{m+4n+1}+\frac{Ab^2d^2e^m e^{(m\log(x)+4n\log(x))}}{m+4n+1}+\frac{Bb^2c^2e^m e^{(m\log(x)+3n\log(x))}}{m+3n+1}+\frac{4Babcde^m e^{(m\log(x)+3n\log(x))}}{m+3n+1}+\frac{2Ab^2cde^m e^{(m\log(x)+3n\log(x))}}{m+3n+1}+\frac{Ba^2d^2e^m e^{(m\log(x)+3n\log(x))}}{m+3n+1}+\frac{2Aabd^2e^m e^{(m\log(x)+3n\log(x))}}{m+3n+1}+\frac{2Babc^2e^m e^{(m\log(x)+2n\log(x))}}{m+2n+1}+\frac{Ab^2c^2e^m e^{(m\log(x)+2n\log(x))}}{m+2n+1}+\frac{2Ba^2cde^m e^{(m\log(x)+2n\log(x))}}{m+2n+1}+\frac{4Aabcde^m e^{(m\log(x)+2n\log(x))}}{m+2n+1}+\frac{Aa^2d^2e^m e^{(m\log(x)+2n\log(x))}}{m+2n+1}+\frac{Ba^2c^2e^m e^{(m\log(x)+n\log(x))}}{m+n+1}+\frac{2Aabc^2e^m e^{(m\log(x)+n\log(x))}}{m+n+1}+\frac{2Aa^2cde^m e^{(m\log(x)+n\log(x))}}{m+n+1}+\frac{(ex)^{m+1}Aa^2c^2}{e(m+1)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)^2*(A+B*x^n)*(c+d*x^n)^2,x, algorithm="maxima")
```

```
[Out] B*b^2*d^2*e^m*x*e^(m*log(x) + 5*n*log(x))/(m + 5*n + 1) + 2*B*b^2*c*d*e^m*x
*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) + 2*B*a*b*d^2*e^m*x*e^(m*log(x) +
4*n*log(x))/(m + 4*n + 1) + A*b^2*d^2*e^m*x*e^(m*log(x) + 4*n*log(x))/(m +
4*n + 1) + B*b^2*c^2*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + 4*B*a*
b*c*d*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + 2*A*b^2*c*d*e^m*x*e^(
m*log(x) + 3*n*log(x))/(m + 3*n + 1) + B*a^2*d^2*e^m*x*e^(m*log(x) + 3*n*lo
g(x))/(m + 3*n + 1) + 2*A*a*b*d^2*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n
+ 1) + 2*B*a*b*c^2*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + A*b^2*c^
2*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 2*B*a^2*c*d*e^m*x*e^(m*lo
g(x) + 2*n*log(x))/(m + 2*n + 1) + 4*A*a*b*c*d*e^m*x*e^(m*log(x) + 2*n*log(
x))/(m + 2*n + 1) + A*a^2*d^2*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1)
 + B*a^2*c^2*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1) + 2*A*a*b*c^2*e^m*x*
e^(m*log(x) + n*log(x))/(m + n + 1) + 2*A*a^2*c*d*e^m*x*e^(m*log(x) + n*log
(x))/(m + n + 1) + (e*x)^(m + 1)*A*a^2*c^2/(e*(m + 1))
```

mupad [B] time = 5.59, size = 1119, normalized size = 4.72

$$\frac{x\,x^{2n}(ex)^m\left(2Ba^2cd+Aa^2d^2+2Babc^2+4Aabcd+Ab^2c^2\right)\left(m^4+13m^3n+4m^3+59m^2n^2+39m^2n+6m^2+107mn^3+118mn^2+39mn+4m+60n^4+107n^3+59n^2+13n+1\right)}{m^5+15m^4n+5m^4+85m^3n^2+60m^3n+10m^3+225m^2n^3+255m^2n^2+90m^2n+10m^2+274mn^4+450mn^3+255mn^2+60mn+5m+120n^5+274n^4+225n^3+85n^2+15n+1}+\frac{x\,x^{3n}(ex)^m\left(Ba^2d^2+4Babcd+2Aabd^2+Bb^2c^2+2Ab^2cd\right)\left(m^4+12m^3n+4m^3+49m^2n^2+36m^2n+6m^2+78mn^3+98mn^2+36mn+4m+40n^4+78n^3+49n^2+12n+1\right)}{m^5+15m^4n+5m^4+85m^3n^2+60m^3n+10m^3+225m^2n^3+255m^2n^2+90m^2n+10m^2+274mn^4+450mn^3+255mn^2+60mn+5m+120n^5+274n^4+225n^3+85n^2+15n+1}+\frac{Aa^2c^2x(ex)^m}{m+1}+\frac{bd\,x\,x^{4n}(ex)^m\left(Abd+2Bad+2Bbc\right)\left(m^4+11m^3n+4m^3+41m^2n^2+33m^2n+6m^2+61mn^3+82mn^2+33mn+4m+30n^4+61n^3+41n^2+11n+1\right)}{m^5+15m^4n+5m^4+85m^3n^2+60m^3n+10m^3+225m^2n^3+255m^2n^2+90m^2n+10m^2+274mn^4+450mn^3+255mn^2+60mn+5m+120n^5+274n^4+225n^3+85n^2+15n+1}+\frac{Bb^2d^2x\,x^{5n}(ex)^m\left(m^4+10m^3n+4m^3+35m^2n^2+30m^2n+6m^2+50mn^3+70mn^2+30mn+4m+24n^4+50n^3+35n^2+10n+1\right)}{m^5+15m^4n+5m^4+85m^3n^2+60m^3n+10m^3+225m^2n^3+255m^2n^2+90m^2n+10m^2+274mn^4+450mn^3+255mn^2+60mn+5m+120n^5+274n^4+225n^3+85n^2+15n+1}+\frac{ac\,x\,x^n(ex)^m\left(2Aad+2Abc+Bac\right)\left(m^4+14m^3n+4m^3+71m^2n^2+42m^2n+6m^2+154mn^3+142mn^2+42mn+4m+120n^4+154n^3+71n^2+14n+1\right)}{m^5+15m^4n+5m^4+85m^3n^2+60m^3n+10m^3+225m^2n^3+255m^2n^2+90m^2n+10m^2+274mn^4+450mn^3+255mn^2+60mn+5m+120n^5+274n^4+225n^3+85n^2+15n+1}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(A + B*x^n)*(a + b*x^n)^2*(c + d*x^n)^2,x)
```

```
[Out] (x*x^(2*n)*(e*x)^m*(A*a^2*d^2 + A*b^2*c^2 + 2*B*a*b*c^2 + 2*B*a^2*c*d + 4*A
*a*b*c*d)*(4*m + 13*n + 39*m*n + 118*m*n^2 + 39*m^2*n + 107*m*n^3 + 13*m^3*
n + 6*m^2 + 4*m^3 + m^4 + 59*n^2 + 107*n^3 + 60*n^4 + 59*m^2*n^2 + 1))/(5*m
 + 15*n + 60*m*n + 255*m*n^2 + 90*m^2*n + 450*m*n^3 + 60*m^3*n + 274*m*n^4
+ 15*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 85*n^2 + 225*n^3 + 274*n^4 + 1
20*n^5 + 255*m^2*n^2 + 225*m^2*n^3 + 85*m^3*n^2 + 1) + (x*x^(3*n)*(e*x)^m*(
B*a^2*d^2 + B*b^2*c^2 + 2*A*a*b*d^2 + 2*A*b^2*c*d + 4*B*a*b*c*d)*(4*m + 12*
n + 36*m*n + 98*m*n^2 + 36*m^2*n + 78*m*n^3 + 12*m^3*n + 6*m^2 + 4*m^3 + m^
4 + 49*n^2 + 78*n^3 + 40*n^4 + 49*m^2*n^2 + 1))/(5*m + 15*n + 60*m*n + 255*
m*n^2 + 90*m^2*n + 450*m*n^3 + 60*m^3*n + 274*m*n^4 + 15*m^4*n + 10*m^2 + 1
0*m^3 + 5*m^4 + m^5 + 85*n^2 + 225*n^3 + 274*n^4 + 120*n^5 + 255*m^2*n^2 +
225*m^2*n^3 + 85*m^3*n^2 + 1) + (A*a^2*c^2*x*(e*x)^m)/(m + 1) + (b*d*x*x^(4
*n)*(e*x)^m*(A*b*d + 2*B*a*d + 2*B*b*c)*(4*m + 11*n + 33*m*n + 82*m*n^2 + 3
3*m^2*n + 61*m*n^3 + 11*m^3*n + 6*m^2 + 4*m^3 + m^4 + 41*n^2 + 61*n^3 + 30*
n^4 + 41*m^2*n^2 + 1))/(5*m + 15*n + 60*m*n + 255*m*n^2 + 90*m^2*n + 450*m*
n^3 + 60*m^3*n + 274*m*n^4 + 15*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 85*
n^2 + 225*n^3 + 274*n^4 + 120*n^5 + 255*m^2*n^2 + 225*m^2*n^3 + 85*m^3*n^2
+ 1) + (B*b^2*d^2*x*x^(5*n)*(e*x)^m*(4*m + 10*n + 30*m*n + 70*m*n^2 + 30*m^
2*n + 50*m*n^3 + 10*m^3*n + 6*m^2 + 4*m^3 + m^4 + 35*n^2 + 50*n^3 + 24*n^4
+ 35*m^2*n^2 + 1))/(5*m + 15*n + 60*m*n + 255*m*n^2 + 90*m^2*n + 450*m*n^3
+ 60*m^3*n + 274*m*n^4 + 15*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 85*n^2
+ 225*n^3 + 274*n^4 + 120*n^5 + 255*m^2*n^2 + 225*m^2*n^3 + 85*m^3*n^2 + 1)
 + (a*c*x*x^n*(e*x)^m*(2*A*a*d + 2*A*b*c + B*a*c)*(4*m + 14*n + 42*m*n + 14
2*m*n^2 + 42*m^2*n + 154*m*n^3 + 14*m^3*n + 6*m^2 + 4*m^3 + m^4 + 71*n^2 +
154*n^3 + 120*n^4 + 71*m^2*n^2 + 1))/(5*m + 15*n + 60*m*n + 255*m*n^2 + 90*
m^2*n + 450*m*n^3 + 60*m^3*n + 274*m*n^4 + 15*m^4*n + 10*m^2 + 10*m^3 + 5*m
```

```
^4 + m^5 + 85*n^2 + 225*n^3 + 274*n^4 + 120*n^5 + 255*m^2*n^2 + 225*m^2*n^3
 + 85*m^3*n^2 + 1)
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)**m*(a+b*x**n)**2*(A+B*x**n)*(c+d*x**n)**2,x)
```

```
[Out] Timed out
```

3.7 $\int (ex)^m (a+bx^n) (A+Bx^n) (c+dx^n)^2 \, dx$

Optimal. Leaf size=160

$$\frac{cx^{n+1}(ex)^m(2aAd+aBc+Abc)}{m+n+1}+\frac{x^{2n+1}(ex)^m(ad(Ad+2Bc)+bc(2Ad+Bc))}{m+2n+1}+\frac{dx^{3n+1}(ex)^m(aBd+Abd+2bBc}{m+3n+1}$$

Rubi [A] time = 0.17, antiderivative size = 160, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 3, integrand size = 29, $\frac{\text{number of rules}}{\text{integrand size}} = 0.103$, Rules used = {570, 20, 30}

$$\frac{cx^{n+1}(ex)^m(2aAd+aBc+Abc)}{m+n+1}+\frac{x^{2n+1}(ex)^m(ad(Ad+2Bc)+bc(2Ad+Bc))}{m+2n+1}+\frac{dx^{3n+1}(ex)^m(aBd+Abd+2bBc)}{m+3n+1}+\frac{aAc^2(ex)^{m+1}}{e(m+1)}+\frac{bBd^2x^{4n+1}(ex)^m}{m+4n+1}$$

Antiderivative was successfully verified.

```
[In] Int[(e*x)^m*(a + b*x^n)*(A + B*x^n)*(c + d*x^n)^2,x]
```

```
[Out] (c*(A*b*c + a*B*c + 2*a*A*d)*x^(1 + n)*(e*x)^m)/(1 + m + n) + ((a*d*(2*B*c
+ A*d) + b*c*(B*c + 2*A*d))*x^(1 + 2*n)*(e*x)^m)/(1 + m + 2*n) + (d*(2*b*B*
c + A*b*d + a*B*d)*x^(1 + 3*n)*(e*x)^m)/(1 + m + 3*n) + (b*B*d^2*x^(1 + 4*n
)*(e*x)^m)/(1 + m + 4*n) + (a*A*c^2*(e*x)^(1 + m))/(e*(1 + m))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 30

```
Int[(x_)^(m_.), x_Symbol] :> Simp[x^(m + 1)/(m + 1), x] /; FreeQ[m, x] && N
eQ[m, -1]
```

Rule 570

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

Rubi steps

$$\begin{aligned}\int (ex)^m (a+bx^n)(A+Bx^n)(c+dx^n)^2 \, dx &= \int \left(aAc^2(ex)^m + c(Abc+aBc+2aAd)x^n(ex)^m + (ad(2Bc+Ad)\right.\\ &= \frac{aAc^2(ex)^{1+m}}{e(1+m)} + \left(bBd^2\right)\int x^{4n}(ex)^m\,dx + (c(Abc+aBc+2aAd))\\ &= \frac{aAc^2(ex)^{1+m}}{e(1+m)} + \left(bBd^2x^{-m}(ex)^m\right)\int x^{m+4n}\,dx + (c(Abc+aBc+2a\\ &= \frac{c(Abc+aBc+2aAd)x^{1+n}(ex)^m}{1+m+n} + \frac{(ad(2Bc+Ad)+bc(Bc+2Ad}{1+m+2n}\end{aligned}$$

Mathematica [A] time = 0.31, size = 129, normalized size = 0.81

$$x(ex)^m\left(\frac{x^{2n}(ad(Ad+2Bc)+bc(2Ad+Bc))}{m+2n+1}+\frac{dx^{3n}(aBd+Abd+2bBc)}{m+3n+1}+\frac{cx^n(2aAd+aBc+Abc)}{m+n+1}+\frac{aAc^2}{m+1}+\frac{bBd^2x^{4n}}{m+4n+1}\right)$$

Antiderivative was successfully verified.

```
[In] Integrate[(e*x)^m*(a + b*x^n)*(A + B*x^n)*(c + d*x^n)^2,x]
```

```
[Out] x*(e*x)^m*((a*A*c^2)/(1 + m) + (c*(A*b*c + a*B*c + 2*a*A*d)*x^n)/(1 + m + n
) + ((a*d*(2*B*c + A*d) + b*c*(B*c + 2*A*d))*x^(2*n))/(1 + m + 2*n) + (d*(2
*b*B*c + A*b*d + a*B*d)*x^(3*n))/(1 + m + 3*n) + (b*B*d^2*x^(4*n))/(1 + m +
 4*n))
```

IntegrateAlgebraic [F] time = 0.16, size = 0, normalized size = 0.00

$$\int (ex)^m (a + bx^n) (A + Bx^n) (c + dx^n)^2 \, dx$$

Verification is not applicable to the result.

```
[In] IntegrateAlgebraic[(e*x)^m*(a + b*x^n)*(A + B*x^n)*(c + d*x^n)^2,x]
```

```
[Out] Defer[IntegrateAlgebraic][(e*x)^m*(a + b*x^n)*(A + B*x^n)*(c + d*x^n)^2, x]
```

fricas [B] time = 0.48, size = 1426, normalized size = 8.91

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)*(A+B*x^n)*(c+d*x^n)^2,x, algorithm="fricas")
```

```
[Out] ((B*b*d^2*m^4 + 4*B*b*d^2*m^3 + 6*B*b*d^2*m^2 + 4*B*b*d^2*m + B*b*d^2 + 6*(
B*b*d^2*m + B*b*d^2)*n^3 + 11*(B*b*d^2*m^2 + 2*B*b*d^2*m + B*b*d^2)*n^2 + 6
*(B*b*d^2*m^3 + 3*B*b*d^2*m^2 + 3*B*b*d^2*m + B*b*d^2)*n)*x*x^(4*n)*e^(m*lo
g(e) + m*log(x)) + ((2*B*b*c*d + (B*a + A*b)*d^2)*m^4 + 2*B*b*c*d + 4*(2*B*
b*c*d + (B*a + A*b)*d^2)*m^3 + 8*(2*B*b*c*d + (B*a + A*b)*d^2 + (2*B*b*c*d
+ (B*a + A*b)*d^2)*m)*n^3 + (B*a + A*b)*d^2 + 6*(2*B*b*c*d + (B*a + A*b)*d^
2)*m^2 + 14*(2*B*b*c*d + (B*a + A*b)*d^2 + (2*B*b*c*d + (B*a + A*b)*d^2)*m^
2 + 2*(2*B*b*c*d + (B*a + A*b)*d^2)*m)*n^2 + 4*(2*B*b*c*d + (B*a + A*b)*d^2
)*m + 7*(2*B*b*c*d + (2*B*b*c*d + (B*a + A*b)*d^2)*m^3 + (B*a + A*b)*d^2 +
3*(2*B*b*c*d + (B*a + A*b)*d^2)*m^2 + 3*(2*B*b*c*d + (B*a + A*b)*d^2)*m)*n)
*x*x^(3*n)*e^(m*log(e) + m*log(x)) + ((B*b*c^2 + A*a*d^2 + 2*(B*a + A*b)*c*
d)*m^4 + B*b*c^2 + A*a*d^2 + 4*(B*b*c^2 + A*a*d^2 + 2*(B*a + A*b)*c*d)*m^3
+ 12*(B*b*c^2 + A*a*d^2 + 2*(B*a + A*b)*c*d + (B*b*c^2 + A*a*d^2 + 2*(B*a +
 A*b)*c*d)*m)*n^3 + 2*(B*a + A*b)*c*d + 6*(B*b*c^2 + A*a*d^2 + 2*(B*a + A*b
)*c*d)*m^2 + 19*(B*b*c^2 + A*a*d^2 + 2*(B*a + A*b)*c*d + (B*b*c^2 + A*a*d^2
 + 2*(B*a + A*b)*c*d)*m^2 + 2*(B*b*c^2 + A*a*d^2 + 2*(B*a + A*b)*c*d)*m)*n^
2 + 4*(B*b*c^2 + A*a*d^2 + 2*(B*a + A*b)*c*d)*m + 8*(B*b*c^2 + A*a*d^2 + (B
*b*c^2 + A*a*d^2 + 2*(B*a + A*b)*c*d)*m^3 + 2*(B*a + A*b)*c*d + 3*(B*b*c^2
+ A*a*d^2 + 2*(B*a + A*b)*c*d)*m^2 + 3*(B*b*c^2 + A*a*d^2 + 2*(B*a + A*b)*c
*d)*m)*n)*x*x^(2*n)*e^(m*log(e) + m*log(x)) + ((2*A*a*c*d + (B*a + A*b)*c^2
)*m^4 + 2*A*a*c*d + 4*(2*A*a*c*d + (B*a + A*b)*c^2)*m^3 + 24*(2*A*a*c*d + (
B*a + A*b)*c^2 + (2*A*a*c*d + (B*a + A*b)*c^2)*m)*n^3 + (B*a + A*b)*c^2 + 6
*(2*A*a*c*d + (B*a + A*b)*c^2)*m^2 + 26*(2*A*a*c*d + (B*a + A*b)*c^2 + (2*A
*a*c*d + (B*a + A*b)*c^2)*m^2 + 2*(2*A*a*c*d + (B*a + A*b)*c^2)*m)*n^2 + 4*
(2*A*a*c*d + (B*a + A*b)*c^2)*m + 9*(2*A*a*c*d + (2*A*a*c*d + (B*a + A*b)*c
^2)*m^3 + (B*a + A*b)*c^2 + 3*(2*A*a*c*d + (B*a + A*b)*c^2)*m^2 + 3*(2*A*a*
c*d + (B*a + A*b)*c^2)*m)*n)*x*x^n*e^(m*log(e) + m*log(x)) + (A*a*c^2*m^4 +
 24*A*a*c^2*n^4 + 4*A*a*c^2*m^3 + 6*A*a*c^2*m^2 + 4*A*a*c^2*m + A*a*c^2 + 5
0*(A*a*c^2*m + A*a*c^2)*n^3 + 35*(A*a*c^2*m^2 + 2*A*a*c^2*m + A*a*c^2)*n^2
+ 10*(A*a*c^2*m^3 + 3*A*a*c^2*m^2 + 3*A*a*c^2*m + A*a*c^2)*n)*x*e^(m*log(e)
 + m*log(x)))/(m^5 + 24*(m + 1)*n^4 + 5*m^4 + 50*(m^2 + 2*m + 1)*n^3 + 10*m
^3 + 35*(m^3 + 3*m^2 + 3*m + 1)*n^2 + 10*m^2 + 10*(m^4 + 4*m^3 + 6*m^2 + 4*
m + 1)*n + 5*m + 1)
```

giac [B] time = 0.82, size = 3415, normalized size = 21.34

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)*(A+B*x^n)*(c+d*x^n)^2,x, algorithm="giac")
```

```
[Out] (B*b*d^2*m^4*x*x^m*x^(4*n)*e^m + 6*B*b*d^2*m^3*n*x*x^m*x^(4*n)*e^m + 11*B*b
*d^2*m^2*n^2*x*x^m*x^(4*n)*e^m + 6*B*b*d^2*m*n^3*x*x^m*x^(4*n)*e^m + 2*B*b*
c*d*m^4*x*x^m*x^(3*n)*e^m + B*a*d^2*m^4*x*x^m*x^(3*n)*e^m + A*b*d^2*m^4*x*x
^m*x^(3*n)*e^m + 14*B*b*c*d*m^3*n*x*x^m*x^(3*n)*e^m + 7*B*a*d^2*m^3*n*x*x^m
*x^(3*n)*e^m + 7*A*b*d^2*m^3*n*x*x^m*x^(3*n)*e^m + 28*B*b*c*d*m^2*n^2*x*x^m
*x^(3*n)*e^m + 14*B*a*d^2*m^2*n^2*x*x^m*x^(3*n)*e^m + 14*A*b*d^2*m^2*n^2*x*
x^m*x^(3*n)*e^m + 16*B*b*c*d*m*n^3*x*x^m*x^(3*n)*e^m + 8*B*a*d^2*m*n^3*x*x^
m*x^(3*n)*e^m + 8*A*b*d^2*m*n^3*x*x^m*x^(3*n)*e^m + B*b*c^2*m^4*x*x^m*x^(2*
n)*e^m + 2*B*a*c*d*m^4*x*x^m*x^(2*n)*e^m + 2*A*b*c*d*m^4*x*x^m*x^(2*n)*e^m
+ A*a*d^2*m^4*x*x^m*x^(2*n)*e^m + 8*B*b*c^2*m^3*n*x*x^m*x^(2*n)*e^m + 16*B*
a*c*d*m^3*n*x*x^m*x^(2*n)*e^m + 16*A*b*c*d*m^3*n*x*x^m*x^(2*n)*e^m + 8*A*a*
d^2*m^3*n*x*x^m*x^(2*n)*e^m + 19*B*b*c^2*m^2*n^2*x*x^m*x^(2*n)*e^m + 38*B*a
*c*d*m^2*n^2*x*x^m*x^(2*n)*e^m + 38*A*b*c*d*m^2*n^2*x*x^m*x^(2*n)*e^m + 19*
A*a*d^2*m^2*n^2*x*x^m*x^(2*n)*e^m + 12*B*b*c^2*m*n^3*x*x^m*x^(2*n)*e^m + 24
*B*a*c*d*m*n^3*x*x^m*x^(2*n)*e^m + 24*A*b*c*d*m*n^3*x*x^m*x^(2*n)*e^m + 12*
A*a*d^2*m*n^3*x*x^m*x^(2*n)*e^m + B*a*c^2*m^4*x*x^m*x^n*e^m + A*b*c^2*m^4*x
*x^m*x^n*e^m + 2*A*a*c*d*m^4*x*x^m*x^n*e^m + 9*B*a*c^2*m^3*n*x*x^m*x^n*e^m
+ 9*A*b*c^2*m^3*n*x*x^m*x^n*e^m + 18*A*a*c*d*m^3*n*x*x^m*x^n*e^m + 26*B*a*c
^2*m^2*n^2*x*x^m*x^n*e^m + 26*A*b*c^2*m^2*n^2*x*x^m*x^n*e^m + 52*A*a*c*d*m^
2*n^2*x*x^m*x^n*e^m + 24*B*a*c^2*m*n^3*x*x^m*x^n*e^m + 24*A*b*c^2*m*n^3*x*x
^m*x^n*e^m + 48*A*a*c*d*m*n^3*x*x^m*x^n*e^m + A*a*c^2*m^4*x*x^m*e^m + 10*A*
a*c^2*m^3*n*x*x^m*e^m + 35*A*a*c^2*m^2*n^2*x*x^m*e^m + 50*A*a*c^2*m*n^3*x*x
^m*e^m + 24*A*a*c^2*n^4*x*x^m*e^m + 4*B*b*d^2*m^3*x*x^m*x^(4*n)*e^m + 18*B*
b*d^2*m^2*n*x*x^m*x^(4*n)*e^m + 22*B*b*d^2*m*n^2*x*x^m*x^(4*n)*e^m + 6*B*b*
d^2*n^3*x*x^m*x^(4*n)*e^m + 8*B*b*c*d*m^3*x*x^m*x^(3*n)*e^m + 4*B*a*d^2*m^3
*x*x^m*x^(3*n)*e^m + 4*A*b*d^2*m^3*x*x^m*x^(3*n)*e^m + 42*B*b*c*d*m^2*n*x*x
^m*x^(3*n)*e^m + 21*B*a*d^2*m^2*n*x*x^m*x^(3*n)*e^m + 21*A*b*d^2*m^2*n*x*x^
m*x^(3*n)*e^m + 56*B*b*c*d*m*n^2*x*x^m*x^(3*n)*e^m + 28*B*a*d^2*m*n^2*x*x^m
*x^(3*n)*e^m + 28*A*b*d^2*m*n^2*x*x^m*x^(3*n)*e^m + 16*B*b*c*d*n^3*x*x^m*x^
(3*n)*e^m + 8*B*a*d^2*n^3*x*x^m*x^(3*n)*e^m + 8*A*b*d^2*n^3*x*x^m*x^(3*n)*e
^m + 4*B*b*c^2*m^3*x*x^m*x^(2*n)*e^m + 8*B*a*c*d*m^3*x*x^m*x^(2*n)*e^m + 8*
A*b*c*d*m^3*x*x^m*x^(2*n)*e^m + 4*A*a*d^2*m^3*x*x^m*x^(2*n)*e^m + 24*B*b*c^
2*m^2*n*x*x^m*x^(2*n)*e^m + 48*B*a*c*d*m^2*n*x*x^m*x^(2*n)*e^m + 48*A*b*c*d
*m^2*n*x*x^m*x^(2*n)*e^m + 24*A*a*d^2*m^2*n*x*x^m*x^(2*n)*e^m + 38*B*b*c^2*
m*n^2*x*x^m*x^(2*n)*e^m + 76*B*a*c*d*m*n^2*x*x^m*x^(2*n)*e^m + 76*A*b*c*d*m
*n^2*x*x^m*x^(2*n)*e^m + 38*A*a*d^2*m*n^2*x*x^m*x^(2*n)*e^m + 12*B*b*c^2*n^
3*x*x^m*x^(2*n)*e^m + 24*B*a*c*d*n^3*x*x^m*x^(2*n)*e^m + 24*A*b*c*d*n^3*x*x
^m*x^(2*n)*e^m + 12*A*a*d^2*n^3*x*x^m*x^(2*n)*e^m + 4*B*a*c^2*m^3*x*x^m*x^n
*e^m + 4*A*b*c^2*m^3*x*x^m*x^n*e^m + 8*A*a*c*d*m^3*x*x^m*x^n*e^m + 27*B*a*c
^2*m^2*n*x*x^m*x^n*e^m + 27*A*b*c^2*m^2*n*x*x^m*x^n*e^m + 54*A*a*c*d*m^2*n*
x*x^m*x^n*e^m + 52*B*a*c^2*m*n^2*x*x^m*x^n*e^m + 52*A*b*c^2*m*n^2*x*x^m*x^n
*e^m + 104*A*a*c*d*m*n^2*x*x^m*x^n*e^m + 24*B*a*c^2*n^3*x*x^m*x^n*e^m + 24*
A*b*c^2*n^3*x*x^m*x^n*e^m + 48*A*a*c*d*n^3*x*x^m*x^n*e^m + 4*A*a*c^2*m^3*x*
x^m*e^m + 30*A*a*c^2*m^2*n*x*x^m*e^m + 70*A*a*c^2*m*n^2*x*x^m*e^m + 50*A*a*
c^2*n^3*x*x^m*e^m + 6*B*b*d^2*m^2*x*x^m*x^(4*n)*e^m + 18*B*b*d^2*m*n*x*x^m*
x^(4*n)*e^m + 11*B*b*d^2*n^2*x*x^m*x^(4*n)*e^m + 12*B*b*c*d*m^2*x*x^m*x^(3*
n)*e^m + 6*B*a*d^2*m^2*x*x^m*x^(3*n)*e^m + 6*A*b*d^2*m^2*x*x^m*x^(3*n)*e^m
+ 42*B*b*c*d*m*n*x*x^m*x^(3*n)*e^m + 21*B*a*d^2*m*n*x*x^m*x^(3*n)*e^m + 21*
A*b*d^2*m*n*x*x^m*x^(3*n)*e^m + 28*B*b*c*d*n^2*x*x^m*x^(3*n)*e^m + 14*B*a*d
^2*n^2*x*x^m*x^(3*n)*e^m + 14*A*b*d^2*n^2*x*x^m*x^(3*n)*e^m + 6*B*b*c^2*m^2
*x*x^m*x^(2*n)*e^m + 12*B*a*c*d*m^2*x*x^m*x^(2*n)*e^m + 12*A*b*c*d*m^2*x*x^
m*x^(2*n)*e^m + 6*A*a*d^2*m^2*x*x^m*x^(2*n)*e^m + 24*B*b*c^2*m*n*x*x^m*x^(2
*n)*e^m + 48*B*a*c*d*m*n*x*x^m*x^(2*n)*e^m + 48*A*b*c*d*m*n*x*x^m*x^(2*n)*e
^m + 24*A*a*d^2*m*n*x*x^m*x^(2*n)*e^m + 19*B*b*c^2*n^2*x*x^m*x^(2*n)*e^m +
```

```
38*B*a*c*d*n^2*x*x^m*x^(2*n)*e^m + 38*A*b*c*d*n^2*x*x^m*x^(2*n)*e^m + 19*A*
a*d^2*n^2*x*x^m*x^(2*n)*e^m + 6*B*a*c^2*m^2*x*x^m*x^n*e^m + 6*A*b*c^2*m^2*x
*x^m*x^n*e^m + 12*A*a*c*d*m^2*x*x^m*x^n*e^m + 27*B*a*c^2*m*n*x*x^m*x^n*e^m
+ 27*A*b*c^2*m*n*x*x^m*x^n*e^m + 54*A*a*c*d*m*n*x*x^m*x^n*e^m + 26*B*a*c^2*
n^2*x*x^m*x^n*e^m + 26*A*b*c^2*n^2*x*x^m*x^n*e^m + 52*A*a*c*d*n^2*x*x^m*x^n
*e^m + 6*A*a*c^2*m^2*x*x^m*e^m + 30*A*a*c^2*m*n*x*x^m*e^m + 35*A*a*c^2*n^2*
x*x^m*e^m + 4*B*b*d^2*m*x*x^m*x^(4*n)*e^m + 6*B*b*d^2*n*x*x^m*x^(4*n)*e^m +
 8*B*b*c*d*m*x*x^m*x^(3*n)*e^m + 4*B*a*d^2*m*x*x^m*x^(3*n)*e^m + 4*A*b*d^2*
m*x*x^m*x^(3*n)*e^m + 14*B*b*c*d*n*x*x^m*x^(3*n)*e^m + 7*B*a*d^2*n*x*x^m*x^
(3*n)*e^m + 7*A*b*d^2*n*x*x^m*x^(3*n)*e^m + 4*B*b*c^2*m*x*x^m*x^(2*n)*e^m +
 8*B*a*c*d*m*x*x^m*x^(2*n)*e^m + 8*A*b*c*d*m*x*x^m*x^(2*n)*e^m + 4*A*a*d^2*
m*x*x^m*x^(2*n)*e^m + 8*B*b*c^2*n*x*x^m*x^(2*n)*e^m + 16*B*a*c*d*n*x*x^m*x^
(2*n)*e^m + 16*A*b*c*d*n*x*x^m*x^(2*n)*e^m + 8*A*a*d^2*n*x*x^m*x^(2*n)*e^m
+ 4*B*a*c^2*m*x*x^m*x^n*e^m + 4*A*b*c^2*m*x*x^m*x^n*e^m + 8*A*a*c*d*m*x*x^m
*x^n*e^m + 9*B*a*c^2*n*x*x^m*x^n*e^m + 9*A*b*c^2*n*x*x^m*x^n*e^m + 18*A*a*c
*d*n*x*x^m*x^n*e^m + 4*A*a*c^2*m*x*x^m*e^m + 10*A*a*c^2*n*x*x^m*e^m + B*b*d
^2*x*x^m*x^(4*n)*e^m + 2*B*b*c*d*x*x^m*x^(3*n)*e^m + B*a*d^2*x*x^m*x^(3*n)*
e^m + A*b*d^2*x*x^m*x^(3*n)*e^m + B*b*c^2*x*x^m*x^(2*n)*e^m + 2*B*a*c*d*x*x
^m*x^(2*n)*e^m + 2*A*b*c*d*x*x^m*x^(2*n)*e^m + A*a*d^2*x*x^m*x^(2*n)*e^m +
B*a*c^2*x*x^m*x^n*e^m + A*b*c^2*x*x^m*x^n*e^m + 2*A*a*c*d*x*x^m*x^n*e^m + A
*a*c^2*x*x^m*e^m)/(m^5 + 10*m^4*n + 35*m^3*n^2 + 50*m^2*n^3 + 24*m*n^4 + 5*
m^4 + 40*m^3*n + 105*m^2*n^2 + 100*m*n^3 + 24*n^4 + 10*m^3 + 60*m^2*n + 105
*m*n^2 + 50*n^3 + 10*m^2 + 40*m*n + 35*n^2 + 5*m + 10*n + 1)
```

maple [C] time = 0.14, size = 2410, normalized size = 15.06

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(b*x^n+a)*(B*x^n+A)*(d*x^n+c)^2,x)
```

```
[Out] x*(28*A*b*d^2*m*n^2*(x^n)^3+18*B*b*d^2*m^2*n*(x^n)^4+22*B*b*d^2*m*n^2*(x^n)
^4+8*A*a*d^2*m^3*n*(x^n)^2+2*B*a*c*d*m^4*(x^n)^2+21*B*a*d^2*m^2*n*(x^n)^3+2
8*B*a*d^2*m*n^2*(x^n)^3+8*B*b*c^2*m^3*n*(x^n)^2+19*B*b*c^2*m^2*n^2*(x^n)^2+
12*B*b*c^2*m*n^3*(x^n)^2+8*B*b*c*d*m^3*(x^n)^3+16*B*b*c*d*n^3*(x^n)^3+18*B*
b*d^2*m*n*(x^n)^4+21*A*b*d^2*m^2*n*(x^n)^3+24*A*a*d^2*m^2*n*(x^n)^2+38*A*a*
d^2*m*n^2*(x^n)^2+9*A*b*c^2*m^3*n*x^n+26*A*b*c^2*m^2*n^2*x^n+24*A*b*c^2*m*n
^3*x^n+8*A*b*c*d*m^3*(x^n)^2+24*A*b*c*d*n^3*(x^n)^2+21*A*b*d^2*m*n*(x^n)^3+
9*B*a*c^2*m^3*n*x^n+26*B*a*c^2*m^2*n^2*x^n+24*B*a*c^2*m*n^3*x^n+8*A*a*c*d*m
^3*x^n+48*A*a*c*d*n^3*x^n+24*A*a*d^2*m*n*(x^n)^2+27*A*b*c^2*m^2*n*x^n+52*A*
b*c^2*m*n^2*x^n+8*B*a*c*d*m^3*(x^n)^2+24*B*a*c*d*n^3*(x^n)^2+21*B*a*d^2*m*n
*(x^n)^3+24*B*b*c^2*m^2*n*(x^n)^2+38*B*b*c^2*m*n^2*(x^n)^2+12*B*b*c*d*m^2*(
x^n)^3+2*A*a*c*d*m^4*x^n+8*B*b*c*d*(x^n)^3*m+14*B*b*c*d*(x^n)^3*n+12*A*b*c*
d*m^2*(x^n)^2+38*A*b*c*d*n^2*(x^n)^2+27*B*a*c^2*m^2*n*x^n+52*B*a*c^2*m*n^2*
x^n+12*B*a*c*d*m^2*(x^n)^2+38*B*a*c*d*n^2*(x^n)^2+24*B*b*c^2*m*n*(x^n)^2+28
*B*b*c*d*n^2*(x^n)^3+16*B*a*c*d*(x^n)^2*n+8*A*a*c*d*x^n*m+18*A*a*c*d*x^n*n+
12*A*a*c*d*m^2*x^n+52*A*a*c*d*n^2*x^n+27*A*b*c^2*m*n*x^n+8*A*b*c*d*(x^n)^2*
m+16*A*b*c*d*(x^n)^2*n+27*B*a*c^2*m*n*x^n+8*B*a*c*d*(x^n)^2*m+B*b*c^2*(x^n)
^2+A*b*c^2*x^n+B*a*c^2*x^n+b*B*d^2*(x^n)^4+A*b*d^2*(x^n)^3+B*a*d^2*(x^n)^3+
A*a*d^2*(x^n)^2+a*A*c^2+6*A*a*c^2*m^2+35*A*a*c^2*n^2+A*a*c^2*m^4+4*A*a*c^2*
m^3+50*A*a*c^2*n^3+24*A*a*c^2*n^4+4*a*A*c^2*m+10*a*A*c^2*n+38*A*b*c*d*m^2*n
^2*(x^n)^2+24*A*b*c*d*m*n^3*(x^n)^2+16*B*a*c*d*m^3*n*(x^n)^2+24*B*a*c*d*m*n
^3*(x^n)^2+42*B*b*c*d*m^2*n*(x^n)^3+56*B*b*c*d*m*n^2*(x^n)^3+18*A*a*c*d*m^3
*n*x^n+52*A*a*c*d*m^2*n^2*x^n+48*A*a*c*d*m*n^3*x^n+4*B*a*d^2*m^3*(x^n)^3+38
*B*a*c*d*m^2*n^2*(x^n)^2+48*B*a*c*d*m*n*(x^n)^2+54*A*a*c*d*m*n*x^n+16*B*b*c
*d*m*n^3*(x^n)^3+16*A*b*c*d*m^3*n*(x^n)^2+42*B*b*c*d*m*n*(x^n)^3+54*A*a*c*d
*m^2*n*x^n+104*A*a*c*d*m*n^2*x^n+14*B*b*c*d*m^3*n*(x^n)^3+28*B*b*c*d*m^2*n^
2*(x^n)^3+48*A*b*c*d*m^2*n*(x^n)^2+76*A*b*c*d*m*n^2*(x^n)^2+48*B*a*c*d*m^2*
n*(x^n)^2+76*B*a*c*d*m*n^2*(x^n)^2+48*A*b*c*d*m*n*(x^n)^2+14*A*b*d^2*n^2*(x
^n)^3+B*a*c^2*m^4*x^n+6*B*a*d^2*m^2*(x^n)^3+14*B*a*d^2*n^2*(x^n)^3+4*B*b*c^
```

```
2*m^3*(x^n)^2+12*B*b*c^2*n^3*(x^n)^2+4*m*b*B*d^2*(x^n)^4+6*b*B*d^2*(x^n)^4*
n+8*B*a*d^2*m*n^3*(x^n)^3+2*B*b*c*d*m^4*(x^n)^3+14*A*b*d^2*m^2*n^2*(x^n)^3+
7*A*b*d^2*m^3*n*(x^n)^3+6*B*b*d^2*m*n^3*(x^n)^4+6*B*b*d^2*m^3*n*(x^n)^4+11*
B*b*d^2*m^2*n^2*(x^n)^4+19*A*a*d^2*m^2*n^2*(x^n)^2+12*A*a*d^2*m*n^3*(x^n)^2
+2*A*b*c*d*m^4*(x^n)^2+8*A*b*d^2*m*n^3*(x^n)^3+7*B*a*d^2*m^3*n*(x^n)^3+14*B
*a*d^2*m^2*n^2*(x^n)^3+8*B*a*d^2*n^3*(x^n)^3+6*A*b*c^2*m^2*x^n+26*A*b*c^2*n
^2*x^n+6*B*b*c^2*m^2*(x^n)^2+19*B*b*c^2*n^2*(x^n)^2+4*A*a*d^2*(x^n)^2*m+8*A
*a*d^2*(x^n)^2*n+70*A*a*c^2*m*n^2+30*A*a*c^2*m*n+2*(x^n)^3*b*B*c*d+2*(x^n)^
2*B*a*c*d+2*x^n*a*A*c*d+2*(x^n)^2*A*b*c*d+10*A*a*c^2*m^3*n+35*A*a*c^2*m^2*n
^2+50*A*a*c^2*m*n^3+30*A*a*c^2*m^2*n+11*B*b*d^2*n^2*(x^n)^4+4*A*a*d^2*m^3*(
x^n)^2+12*A*a*d^2*n^3*(x^n)^2+A*b*c^2*m^4*x^n+6*A*b*d^2*m^2*(x^n)^3+B*b*c^2
*m^4*(x^n)^2+6*B*b*d^2*m^2*(x^n)^4+24*A*b*c^2*n^3*x^n+4*A*b*d^2*(x^n)^3*m+7
*A*b*d^2*(x^n)^3*n+4*B*a*c^2*m^3*x^n+24*B*a*c^2*n^3*x^n+4*B*a*d^2*(x^n)^3*m
+7*B*a*d^2*(x^n)^3*n+8*A*b*d^2*n^3*(x^n)^3+B*b*d^2*m^4*(x^n)^4+A*b*d^2*m^4*
(x^n)^3+B*a*d^2*m^4*(x^n)^3+4*B*b*d^2*m^3*(x^n)^4+6*B*b*d^2*n^3*(x^n)^4+A*a
*d^2*m^4*(x^n)^2+4*A*b*d^2*m^3*(x^n)^3+4*A*b*c^2*x^n*m+9*A*b*c^2*x^n*n+4*B*
a*c^2*x^n*m+9*B*a*c^2*x^n*n+6*A*a*d^2*m^2*(x^n)^2+19*A*a*d^2*n^2*(x^n)^2+4*
A*b*c^2*m^3*x^n+4*B*b*c^2*(x^n)^2*m+8*B*b*c^2*(x^n)^2*n+6*B*a*c^2*m^2*x^n+2
6*B*a*c^2*n^2*x^n)/(m+1)/(m+n+1)/(m+2*n+1)/(m+3*n+1)/(m+4*n+1)*exp(1/2*(-I*
Pi*csgn(I*e)*csgn(I*x)*csgn(I*e*x)+I*Pi*csgn(I*e)*csgn(I*e*x)^2+I*Pi*csgn(I
*x)*csgn(I*e*x)^2-I*Pi*csgn(I*e*x)^3+2*ln(e)+2*ln(x))*m)
```

maxima [B]   time = 0.78, size = 332, normalized size = 2.08

$$\frac{Bbd^2e^mxe^{\left(m\log(x)+4\,n\log(x)\right)}}{m+4\,n+1}+\frac{2\,Bbcde^mxe^{\left(m\log(x)+3\,n\log(x)\right)}}{m+3\,n+1}+\frac{Bad^2e^mxe^{\left(m\log(x)+3\,n\log(x)\right)}}{m+3\,n+1}+\frac{Abd^2e^mxe^{\left(m\log(x)+3\,n\log(x)\right)}}{m+3\,n+1}+\frac{Bbc^2e^mxe^{\left(m\log(x)+2\,n\log(x)\right)}}{m+2\,n+1}+\frac{2\,Bacde^mxe^{\left(m\log(x)+2\,n\log(x)\right)}}{m+2\,n+1}+\frac{2\,Abcde^mxe^{\left(m\log(x)+2\,n\log(x)\right)}}{m+2\,n+1}+\frac{Aad^2e^mxe^{\left(m\log(x)+2\,n\log(x)\right)}}{m+2\,n+1}+\frac{Bac^2e^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{Abc^2e^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{2\,Aacde^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{(ex)^{m+1}Aac^2}{e(m+1)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)*(A+B*x^n)*(c+d*x^n)^2,x, algorithm="maxima")
```

```
[Out] B*b*d^2*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) + 2*B*b*c*d*e^m*x*e^(
m*log(x) + 3*n*log(x))/(m + 3*n + 1) + B*a*d^2*e^m*x*e^(m*log(x) + 3*n*log(
x))/(m + 3*n + 1) + A*b*d^2*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) +
 B*b*c^2*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 2*B*a*c*d*e^m*x*e^
(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 2*A*b*c*d*e^m*x*e^(m*log(x) + 2*n*l
og(x))/(m + 2*n + 1) + A*a*d^2*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1
) + B*a*c^2*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1) + A*b*c^2*e^m*x*e^(m*
log(x) + n*log(x))/(m + n + 1) + 2*A*a*c*d*e^m*x*e^(m*log(x) + n*log(x))/(m
 + n + 1) + (e*x)^(m + 1)*A*a*c^2/(e*(m + 1))
```

mupad [B]   time = 5.20, size = 588, normalized size = 3.68

$$\frac{x\,x^{2\,n}\,{\left(e\,x\right)}^m\,\left(A\,a\,d^2+B\,b\,c^2+2\,A\,b\,c\,d+2\,B\,a\,c\,d\right)\,\left(m^3+8\,m^2\,n+3\,m^2+19\,m\,n^2+16\,m\,n+3\,m+12\,n^3+19\,n^2+8\,n+1\right)}{m^4+10\,m^3\,n+4\,m^3+35\,m^2\,n^2+30\,m^2\,n+6\,m^2+50\,m\,n^3+70\,m\,n^2+30\,m\,n+4\,m+24\,n^4+50\,n^3+35\,n^2+10\,n+1}+\frac{A\,a\,c^2\,x\,{\left(e\,x\right)}^m}{m+1}+\frac{c\,x\,x^n\,{\left(e\,x\right)}^m\,\left(2\,A\,a\,d+A\,b\,c+B\,a\,c\right)\,\left(m^3+9\,m^2\,n+3\,m^2+26\,m\,n^2+18\,m\,n+3\,m+24\,n^3+26\,n^2+9\,n+1\right)}{m^4+10\,m^3\,n+4\,m^3+35\,m^2\,n^2+30\,m^2\,n+6\,m^2+50\,m\,n^3+70\,m\,n^2+30\,m\,n+4\,m+24\,n^4+50\,n^3+35\,n^2+10\,n+1}+\frac{d\,x\,x^{3\,n}\,{\left(e\,x\right)}^m\,\left(A\,b\,d+B\,a\,d+2\,B\,b\,c\right)\,\left(m^3+7\,m^2\,n+3\,m^2+14\,m\,n^2+14\,m\,n+3\,m+8\,n^3+14\,n^2+7\,n+1\right)}{m^4+10\,m^3\,n+4\,m^3+35\,m^2\,n^2+30\,m^2\,n+6\,m^2+50\,m\,n^3+70\,m\,n^2+30\,m\,n+4\,m+24\,n^4+50\,n^3+35\,n^2+10\,n+1}+\frac{B\,b\,d^2\,x\,x^{4\,n}\,{\left(e\,x\right)}^m\,\left(m^3+6\,m^2\,n+3\,m^2+11\,m\,n^2+12\,m\,n+3\,m+6\,n^3+11\,n^2+6\,n+1\right)}{m^4+10\,m^3\,n+4\,m^3+35\,m^2\,n^2+30\,m^2\,n+6\,m^2+50\,m\,n^3+70\,m\,n^2+30\,m\,n+4\,m+24\,n^4+50\,n^3+35\,n^2+10\,n+1}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(A + B*x^n)*(a + b*x^n)*(c + d*x^n)^2,x)
```

```
[Out] (x*x^(2*n)*(e*x)^m*(A*a*d^2 + B*b*c^2 + 2*A*b*c*d + 2*B*a*c*d)*(3*m + 8*n +
 16*m*n + 19*m*n^2 + 8*m^2*n + 3*m^2 + m^3 + 19*n^2 + 12*n^3 + 1))/(4*m + 1
0*n + 30*m*n + 70*m*n^2 + 30*m^2*n + 50*m*n^3 + 10*m^3*n + 6*m^2 + 4*m^3 +
m^4 + 35*n^2 + 50*n^3 + 24*n^4 + 35*m^2*n^2 + 1) + (A*a*c^2*x*(e*x)^m)/(m +
 1) + (c*x*x^n*(e*x)^m*(2*A*a*d + A*b*c + B*a*c)*(3*m + 9*n + 18*m*n + 26*m
*n^2 + 9*m^2*n + 3*m^2 + m^3 + 26*n^2 + 24*n^3 + 1))/(4*m + 10*n + 30*m*n +
 70*m*n^2 + 30*m^2*n + 50*m*n^3 + 10*m^3*n + 6*m^2 + 4*m^3 + m^4 + 35*n^2 +
 50*n^3 + 24*n^4 + 35*m^2*n^2 + 1) + (d*x*x^(3*n)*(e*x)^m*(A*b*d + B*a*d +
2*B*b*c)*(3*m + 7*n + 14*m*n + 14*m*n^2 + 7*m^2*n + 3*m^2 + m^3 + 14*n^2 +
8*n^3 + 1))/(4*m + 10*n + 30*m*n + 70*m*n^2 + 30*m^2*n + 50*m*n^3 + 10*m^3*
n + 6*m^2 + 4*m^3 + m^4 + 35*n^2 + 50*n^3 + 24*n^4 + 35*m^2*n^2 + 1) + (B*b
*d^2*x*x^(4*n)*(e*x)^m*(3*m + 6*n + 12*m*n + 11*m*n^2 + 6*m^2*n + 3*m^2 + m
^3 + 11*n^2 + 6*n^3 + 1))/(4*m + 10*n + 30*m*n + 70*m*n^2 + 30*m^2*n + 50*m
```

```
*n^3 + 10*m^3*n + 6*m^2 + 4*m^3 + m^4 + 35*n^2 + 50*n^3 + 24*n^4 + 35*m^2*n
^2 + 1)
```

sympy [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)**m*(a+b*x**n)*(A+B*x**n)*(c+d*x**n)**2,x)
[Out] Timed out
```

3.8 $\int (ex)^m (A+Bx^n)(c+dx^n)^2 \, dx$

Optimal. Leaf size=102

$$\frac{cx^{n+1}(ex)^m(2Ad+Bc)}{m+n+1}+\frac{dx^{2n+1}(ex)^m(Ad+2Bc)}{m+2n+1}+\frac{Ac^2(ex)^{m+1}}{e(m+1)}+\frac{Bd^2x^{3n+1}(ex)^m}{m+3n+1}$$

Rubi [A] time = 0.08, antiderivative size = 102, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 3, integrand size = 22, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.136, Rules used = {448, 20, 30}

$$\frac{cx^{n+1}(ex)^m(2Ad+Bc)}{m+n+1}+\frac{dx^{2n+1}(ex)^m(Ad+2Bc)}{m+2n+1}+\frac{Ac^2(ex)^{m+1}}{e(m+1)}+\frac{Bd^2x^{3n+1}(ex)^m}{m+3n+1}$$

Antiderivative was successfully verified.

```
[In] Int[(e*x)^m*(A + B*x^n)*(c + d*x^n)^2,x]
```

```
[Out] (c*(B*c + 2*A*d)*x^(1 + n)*(e*x)^m)/(1 + m + n) + (d*(2*B*c + A*d)*x^(1 + 2
*n)*(e*x)^m)/(1 + m + 2*n) + (B*d^2*x^(1 + 3*n)*(e*x)^m)/(1 + m + 3*n) + (A
*c^2*(e*x)^(1 + m))/(e*(1 + m))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 30

```
Int[(x_)^(m_.), x_Symbol] :> Simp[x^(m + 1)/(m + 1), x] /; FreeQ[m, x] && N
eQ[m, -1]
```

Rule 448

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

Rubi steps

$$\begin{aligned}\int (ex)^m (A+Bx^n)(c+dx^n)^2 \, dx &= \int \left(Ac^2(ex)^m+c(Bc+2Ad)x^n(ex)^m+d(2Bc+Ad)x^{2n}(ex)^m+Bd^2x^{3n}(ex)^{\text{}}\right.\\ &=\frac{Ac^2(ex)^{1+m}}{e(1+m)}+\left(Bd^2\right)\int x^{3n}(ex)^m\,dx+(d(2Bc+Ad))\int x^{2n}(ex)^m\,dx+(c(B\\ &=\frac{Ac^2(ex)^{1+m}}{e(1+m)}+\left(Bd^2x^{-m}(ex)^m\right)\int x^{m+3n}\,dx+\left(d(2Bc+Ad)x^{-m}(ex)^m\right)\int x^{m\text{}}\\ &=\frac{c(Bc+2Ad)x^{1+n}(ex)^m}{1+m+n}+\frac{d(2Bc+Ad)x^{1+2n}(ex)^m}{1+m+2n}+\frac{Bd^2x^{1+3n}(ex)^m}{1+m+3n}+\frac{Ac^2}{e(1}\end{aligned}$$

Mathematica [A] time = 0.18, size = 78, normalized size = 0.76

$$x(ex)^m\left(\frac{dx^{2n}(Ad+2Bc)}{m+2n+1}+\frac{cx^n(2Ad+Bc)}{m+n+1}+\frac{Ac^2}{m+1}+\frac{Bd^2x^{3n}}{m+3n+1}\right)$$

Antiderivative was successfully verified.

```
[In] Integrate[(e*x)^m*(A + B*x^n)*(c + d*x^n)^2,x]
```

```
[Out] x*(e*x)^m*((A*c^2)/(1 + m) + (c*(B*c + 2*A*d)*x^n)/(1 + m + n) + (d*(2*B*c
+ A*d)*x^(2*n))/(1 + m + 2*n) + (B*d^2*x^(3*n))/(1 + m + 3*n))
```

IntegrateAlgebraic [F] time = 0.10, size = 0, normalized size = 0.00

$$\int (ex)^m (A + Bx^n) (c + dx^n)^2 \, dx$$

Verification is not applicable to the result.

```
[In] IntegrateAlgebraic[(e*x)^m*(A + B*x^n)*(c + d*x^n)^2,x]
```

```
[Out] Defer[IntegrateAlgebraic][(e*x)^m*(A + B*x^n)*(c + d*x^n)^2, x]
```

fricas [B] time = 0.45, size = 527, normalized size = 5.17

$$\frac{\left(Bd^2m^3 + 3\,Bd^2m^2 + 3\,Bd^2m + Bd^2 + 2\left(Bd^2m + Bd^2\right)n^2 + 3\left(Bd^2m^2 + 2\,Bd^2m + Bd^2\right)n\right)xx^{3\,n}e^{\left(m\log(e) + m\log(x)\right)} + \left(\left(2\,Bcd + Ad^2\right)m^3 + 2\,Bcd + Ad^2 + 3\left(2\,Bcd + Ad^2\right)m^2 + 3\left(2\,Bcd + Ad^2 + \left(2\,Bcd + Ad^2\right)m\right)n^2 + 3\left(2\,Bcd + Ad^2\right)m + 4\left(2\,Bcd + Ad^2 + \left(2\,Bcd + Ad^2\right)m^2 + 2\left(2\,Bcd + Ad^2\right)m\right)n\right)xx^{2\,n}e^{\left(m\log(e) + m\log(x)\right)} + \left(\left(Bc^2 + 2\,Acd\right)m^3 + Bc^2 + 2\,Acd + 3\left(Bc^2 + 2\,Acd\right)m^2 + 6\left(Bc^2 + 2\,Acd + \left(Bc^2 + 2\,Acd\right)m\right)n^2 + 3\left(Bc^2 + 2\,Acd\right)m + 5\left(Bc^2 + 2\,Acd + \left(Bc^2 + 2\,Acd\right)m^2 + 2\left(Bc^2 + 2\,Acd\right)m\right)n\right)xx^{n}e^{\left(m\log(e) + m\log(x)\right)} + \left(Ac^2m^3 + 6\,Ac^2n^3 + 3\,Ac^2m^2 + 3\,Ac^2m + Ac^2 + 11\left(Ac^2m + Ac^2\right)n^2 + 6\left(Ac^2m^2 + 2\,Ac^2m + Ac^2\right)n\right)xe^{\left(m\log(e) + m\log(x)\right)}}{m^4 + 6\,(m+1)n^3 + 4\,m^3 + 11\left(m^2 + 2\,m + 1\right)n^2 + 6\,m^2 + 6\left(m^3 + 3\,m^2 + 3\,m + 1\right)n + 4\,m + 1}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(A+B*x^n)*(c+d*x^n)^2,x, algorithm="fricas")
```

```
[Out] ((B*d^2*m^3 + 3*B*d^2*m^2 + 3*B*d^2*m + B*d^2 + 2*(B*d^2*m + B*d^2)*n^2 + 3
*(B*d^2*m^2 + 2*B*d^2*m + B*d^2)*n)*x*x^(3*n)*e^(m*log(e) + m*log(x)) + ((2
*B*c*d + A*d^2)*m^3 + 2*B*c*d + A*d^2 + 3*(2*B*c*d + A*d^2)*m^2 + 3*(2*B*c*
d + A*d^2 + (2*B*c*d + A*d^2)*m)*n^2 + 3*(2*B*c*d + A*d^2)*m + 4*(2*B*c*d +
 A*d^2 + (2*B*c*d + A*d^2)*m^2 + 2*(2*B*c*d + A*d^2)*m)*n)*x*x^(2*n)*e^(m*l
og(e) + m*log(x)) + ((B*c^2 + 2*A*c*d)*m^3 + B*c^2 + 2*A*c*d + 3*(B*c^2 + 2
*A*c*d)*m^2 + 6*(B*c^2 + 2*A*c*d + (B*c^2 + 2*A*c*d)*m)*n^2 + 3*(B*c^2 + 2*
A*c*d)*m + 5*(B*c^2 + 2*A*c*d + (B*c^2 + 2*A*c*d)*m^2 + 2*(B*c^2 + 2*A*c*d)
*m)*n)*x*x^n*e^(m*log(e) + m*log(x)) + (A*c^2*m^3 + 6*A*c^2*n^3 + 3*A*c^2*m
^2 + 3*A*c^2*m + A*c^2 + 11*(A*c^2*m + A*c^2)*n^2 + 6*(A*c^2*m^2 + 2*A*c^2*
m + A*c^2)*n)*x*e^(m*log(e) + m*log(x)))/(m^4 + 6*(m + 1)*n^3 + 4*m^3 + 11*
(m^2 + 2*m + 1)*n^2 + 6*m^2 + 6*(m^3 + 3*m^2 + 3*m + 1)*n + 4*m + 1)
```

giac [B] time = 0.58, size = 1023, normalized size = 10.03

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(A+B*x^n)*(c+d*x^n)^2,x, algorithm="giac")
```

```
[Out] (B*d^2*m^3*x*x^m*x^(3*n)*e^m + 3*B*d^2*m^2*n*x*x^m*x^(3*n)*e^m + 2*B*d^2*m*
n^2*x*x^m*x^(3*n)*e^m + 2*B*c*d*m^3*x*x^m*x^(2*n)*e^m + A*d^2*m^3*x*x^m*x^(
2*n)*e^m + 8*B*c*d*m^2*n*x*x^m*x^(2*n)*e^m + 4*A*d^2*m^2*n*x*x^m*x^(2*n)*e^
m + 6*B*c*d*m*n^2*x*x^m*x^(2*n)*e^m + 3*A*d^2*m*n^2*x*x^m*x^(2*n)*e^m + B*c
^2*m^3*x*x^m*x^n*e^m + 2*A*c*d*m^3*x*x^m*x^n*e^m + 5*B*c^2*m^2*n*x*x^m*x^n*
e^m + 10*A*c*d*m^2*n*x*x^m*x^n*e^m + 6*B*c^2*m*n^2*x*x^m*x^n*e^m + 12*A*c*d
*m*n^2*x*x^m*x^n*e^m + A*c^2*m^3*x*x^m*e^m + 6*A*c^2*m^2*n*x*x^m*e^m + 11*A
*c^2*m*n^2*x*x^m*e^m + 6*A*c^2*n^3*x*x^m*e^m + 3*B*d^2*m^2*x*x^m*x^(3*n)*e^
m + 6*B*d^2*m*n*x*x^m*x^(3*n)*e^m + 2*B*d^2*n^2*x*x^m*x^(3*n)*e^m + 6*B*c*d
*m^2*x*x^m*x^(2*n)*e^m + 3*A*d^2*m^2*x*x^m*x^(2*n)*e^m + 16*B*c*d*m*n*x*x^m
*x^(2*n)*e^m + 8*A*d^2*m*n*x*x^m*x^(2*n)*e^m + 6*B*c*d*n^2*x*x^m*x^(2*n)*e^
m + 3*A*d^2*n^2*x*x^m*x^(2*n)*e^m + 3*B*c^2*m^2*x*x^m*x^n*e^m + 6*A*c*d*m^2
*x*x^m*x^n*e^m + 10*B*c^2*m*n*x*x^m*x^n*e^m + 20*A*c*d*m*n*x*x^m*x^n*e^m +
6*B*c^2*n^2*x*x^m*x^n*e^m + 12*A*c*d*n^2*x*x^m*x^n*e^m + 3*A*c^2*m^2*x*x^m*
e^m + 12*A*c^2*m*n*x*x^m*e^m + 11*A*c^2*n^2*x*x^m*e^m + 3*B*d^2*m*x*x^m*x^(
3*n)*e^m + 3*B*d^2*n*x*x^m*x^(3*n)*e^m + 6*B*c*d*m*x*x^m*x^(2*n)*e^m + 3*A*
```

```
d^2*m*x*x^m*x^(2*n)*e^m + 8*B*c*d*n*x*x^m*x^(2*n)*e^m + 4*A*d^2*n*x*x^m*x^(
2*n)*e^m + 3*B*c^2*m*x*x^m*x^n*e^m + 6*A*c*d*m*x*x^m*x^n*e^m + 5*B*c^2*n*x*
x^m*x^n*e^m + 10*A*c*d*n*x*x^m*x^n*e^m + 3*A*c^2*m*x*x^m*e^m + 6*A*c^2*n*x*
x^m*e^m + B*d^2*x*x^m*x^(3*n)*e^m + 2*B*c*d*x*x^m*x^(2*n)*e^m + A*d^2*x*x^m
*x^(2*n)*e^m + B*c^2*x*x^m*x^n*e^m + 2*A*c*d*x*x^m*x^n*e^m + A*c^2*x*x^m*e^
m)/(m^4 + 6*m^3*n + 11*m^2*n^2 + 6*m*n^3 + 4*m^3 + 18*m^2*n + 22*m*n^2 + 6*
n^3 + 6*m^2 + 18*m*n + 11*n^2 + 4*m + 6*n + 1)
```

maple [C] time = 0.11, size = 732, normalized size = 7.18

[illegible]

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(B*x^n+A)*(d*x^n+c)^2,x)
```

```
[Out] x*(2*B*c*d*m^3*(x^n)^2+6*B*d^2*m*n*(x^n)^3+2*B*d^2*n^2*(x^n)^3+A*d^2*m^3*(x
^n)^2+3*B*d^2*m^2*(x^n)^3+B*d^2*m^3*(x^n)^3+B*c^2*m^3*x^n+3*A*d^2*(x^n)^2*m
+3*A*d^2*m^2*(x^n)^2+3*A*d^2*n^2*(x^n)^2+3*m*B*d^2*(x^n)^3+3*B*d^2*(x^n)^3*
n+2*B*c*d*(x^n)^2+2*A*c*d*x^n+3*B*c^2*m^2*x^n+6*B*c^2*n^2*x^n+3*B*c^2*x^n*m
+5*B*c^2*x^n*n+4*A*d^2*(x^n)^2*n+6*A*c^2*m^2*n+11*A*c^2*m*n^2+12*A*c^2*m*n+
A*c^2+6*B*c*d*m*n^2*(x^n)^2+10*A*c*d*m^2*n*x^n+12*A*c*d*m*n^2*x^n+16*B*c*d*
m*n*(x^n)^2+20*A*c*d*m*n*x^n+8*B*c*d*m^2*n*(x^n)^2+6*A*c*d*m^2*x^n+12*A*c*d
*n^2*x^n+10*B*c^2*m*n*x^n+6*B*c*d*(x^n)^2*m+8*B*c*d*(x^n)^2*n+6*A*c*d*x^n*m
+10*A*c*d*x^n*n+3*B*d^2*m^2*n*(x^n)^3+2*B*d^2*m*n^2*(x^n)^3+3*A*c^2*m+6*A*c
^2*n+6*A*c^2*n^3+3*A*c^2*m^2+11*A*c^2*n^2+x^n*B*c^2+(x^n)^3*B*d^2+A*c^2*m^3
+(x^n)^2*A*d^2+4*A*d^2*m^2*n*(x^n)^2+3*A*d^2*m*n^2*(x^n)^2+2*A*c*d*m^3*x^n+
8*A*d^2*m*n*(x^n)^2+5*B*c^2*m^2*n*x^n+6*B*c^2*m*n^2*x^n+6*B*c*d*m^2*(x^n)^2
+6*B*c*d*n^2*(x^n)^2)/(m+1)/(m+n+1)/(m+2*n+1)/(m+3*n+1)*exp(1/2*(-I*Pi*csgn
(I*e)*csgn(I*x)*csgn(I*e*x)+I*Pi*csgn(I*e)*csgn(I*e*x)^2+I*Pi*csgn(I*x)*csg
n(I*e*x)^2-I*Pi*csgn(I*e*x)^3+2*ln(e)+2*ln(x))*m)
```

maxima [A] time = 0.63, size = 155, normalized size = 1.52

$$\frac{Bd^2e^m xe^{\left(m\log(x)+3\,n\log(x)\right)}}{m+3\,n+1}+\frac{2\,Bcde^m xe^{\left(m\log(x)+2\,n\log(x)\right)}}{m+2\,n+1}+\frac{Ad^2e^m xe^{\left(m\log(x)+2\,n\log(x)\right)}}{m+2\,n+1}+\frac{Bc^2e^m xe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{2\,Acde^m xe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{(ex)^{m+1}\,Ac^2}{e(m+1)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(A+B*x^n)*(c+d*x^n)^2,x, algorithm="maxima")
```

```
[Out] B*d^2*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + 2*B*c*d*e^m*x*e^(m*lo
g(x) + 2*n*log(x))/(m + 2*n + 1) + A*d^2*e^m*x*e^(m*log(x) + 2*n*log(x))/(m
 + 2*n + 1) + B*c^2*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1) + 2*A*c*d*e^m
*x*e^(m*log(x) + n*log(x))/(m + n + 1) + (e*x)^(m + 1)*A*c^2/(e*(m + 1))
```

mupad [B] time = 5.11, size = 265, normalized size = 2.60

$$\frac{A\,c^2\,x\,(e\,x)^m}{m+1}+\frac{c\,x\,x^n\,(e\,x)^m\,(2\,A\,d+B\,c)\,\left(m^2+5\,m\,n+2\,m+6\,n^2+5\,n+1\right)}{m^3+6\,m^2\,n+3\,m^2+11\,m\,n^2+12\,m\,n+3\,m+6\,n^3+11\,n^2+6\,n+1}+\frac{d\,x\,x^{2\,n}\,(e\,x)^m\,(A\,d+2\,B\,c)\,\left(m^2+4\,m\,n+2\,m+3\,n^2+4\,n+1\right)}{m^3+6\,m^2\,n+3\,m^2+11\,m\,n^2+12\,m\,n+3\,m+6\,n^3+11\,n^2+6\,n+1}+\frac{B\,d^2\,x\,x^{3\,n}\,(e\,x)^m\,\left(m^2+3\,m\,n+2\,m+2\,n^2+3\,n+1\right)}{m^3+6\,m^2\,n+3\,m^2+11\,m\,n^2+12\,m\,n+3\,m+6\,n^3+11\,n^2+6\,n+1}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(A + B*x^n)*(c + d*x^n)^2,x)
```

```
[Out] (A*c^2*x*(e*x)^m)/(m + 1) + (c*x*x^n*(e*x)^m*(2*A*d + B*c)*(2*m + 5*n + 5*m
*n + m^2 + 6*n^2 + 1))/(3*m + 6*n + 12*m*n + 11*m*n^2 + 6*m^2*n + 3*m^2 + m
^3 + 11*n^2 + 6*n^3 + 1) + (d*x*x^(2*n)*(e*x)^m*(A*d + 2*B*c)*(2*m + 4*n +
4*m*n + m^2 + 3*n^2 + 1))/(3*m + 6*n + 12*m*n + 11*m*n^2 + 6*m^2*n + 3*m^2
+ m^3 + 11*n^2 + 6*n^3 + 1) + (B*d^2*x*x^(3*n)*(e*x)^m*(2*m + 3*n + 3*m*n +
 m^2 + 2*n^2 + 1))/(3*m + 6*n + 12*m*n + 11*m*n^2 + 6*m^2*n + 3*m^2 + m^3 +
 11*n^2 + 6*n^3 + 1)
```

sympy [A] time = 74.16, size = 6399, normalized size = 62.74

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)**m*(A+B*x**n)*(c+d*x**n)**2,x)
```

```
[Out] Piecewise(((A + B)*(c + d)**2*log(x)/e, Eq(m, -1) & Eq(n, 0)), ((A*c**2*log
(x) + 2*A*c*d*x**n/n + A*d**2*x**(2*n)/(2*n) + B*c**2*x**n/n + B*c*d*x**(2*
n)/n + B*d**2*x**(3*n)/(3*n))/e, Eq(m, -1)), (A*c**2*Piecewise((log(x), Eq(
n, 0)), (-x**(-3*n)*(0**(1/n))**(-3*n)/(3*n), Eq(e, 0**(1/n))), (-e**(-3*n)
*x**(-3*n)/(3*n), True))/e + 2*A*c*d*Piecewise((log(x), Eq(n, 0)), (-x**n/(
3*0**(1/n)*zoo**(1/n)*n*x**(3*n)*(0**(1/n))**(3*n) - n*x**(3*n)*(0**(1/n))*
*(3*n)), Eq(e, 0**(1/n))), (-e**(-3*n)*x**(-2*n)/(2*n), True))/e + A*d**2*P
iecewise((log(x), Eq(n, 0)), (-x**(2*n)/(3*0**(1/n)*zoo**(1/n)*n*x**(3*n)*(
0**(1/n))**(3*n) - 2*n*x**(3*n)*(0**(1/n))**(3*n)), Eq(e, 0**(1/n))), (-e**
(-3*n)*x**(-n)/n, True))/e + B*c**2*Piecewise((log(x), Eq(n, 0)), (-x**n/(3
*0**(1/n)*zoo**(1/n)*n*x**(3*n)*(0**(1/n))**(3*n) - n*x**(3*n)*(0**(1/n))**
(3*n)), Eq(e, 0**(1/n))), (-e**(-3*n)*x**(-2*n)/(2*n), True))/e + 2*B*c*d*P
iecewise((log(x), Eq(n, 0)), (-x**(2*n)/(3*0**(1/n)*zoo**(1/n)*n*x**(3*n)*(
0**(1/n))**(3*n) - 2*n*x**(3*n)*(0**(1/n))**(3*n)), Eq(e, 0**(1/n))), (-e**
(-3*n)*x**(-n)/n, True))/e + B*d**2*Piecewise((e**(-3*n)*log(x), Abs(x) < 1
), (-e**(-3*n)*log(1/x), 1/Abs(x) < 1), (-e**(-3*n)*meijerg(((), (1, 1)), (
(0, 0), ()), x) + e**(-3*n)*meijerg(((1, 1), ()), ((), (0, 0)), x), True))/
e, Eq(m, -3*n - 1)), (A*c**2*Piecewise((log(x), Eq(n, 0)), (-x**(-2*n)*(0**
(1/n))**(-2*n)/(2*n), Eq(e, 0**(1/n))), (-e**(-2*n)*x**(-2*n)/(2*n), True))
/e + 2*A*c*d*Piecewise((log(x), Eq(n, 0)), (-x**n/(2*0**(1/n)*zoo**(1/n)*n*
x**(2*n)*(0**(1/n))**(2*n) - n*x**(2*n)*(0**(1/n))**(2*n)), Eq(e, 0**(1/n))
), (-e**(-2*n)*x**(-n)/n, True))/e + A*d**2*Piecewise((e**(-2*n)*log(x), Ab
s(x) < 1), (-e**(-2*n)*log(1/x), 1/Abs(x) < 1), (-e**(-2*n)*meijerg(((), (1
, 1)), ((0, 0), ()), x) + e**(-2*n)*meijerg(((1, 1), ()), ((), (0, 0)), x),
 True))/e + B*c**2*Piecewise((log(x), Eq(n, 0)), (-x**n/(2*0**(1/n)*zoo**(1
/n)*n*x**(2*n)*(0**(1/n))**(2*n) - n*x**(2*n)*(0**(1/n))**(2*n)), Eq(e, 0**
(1/n))), (-e**(-2*n)*x**(-n)/n, True))/e + 2*B*c*d*Piecewise((e**(-2*n)*log
(x), Abs(x) < 1), (-e**(-2*n)*log(1/x), 1/Abs(x) < 1), (-e**(-2*n)*meijerg(
((), (1, 1)), ((0, 0), ()), x) + e**(-2*n)*meijerg(((1, 1), ()), ((), (0, 0
)), x), True))/e + B*d**2*Piecewise((log(x), Eq(n, 0)), (-x**(3*n)/(2*0**(1
/n)*zoo**(1/n)*n*x**(2*n)*(0**(1/n))**(2*n) - 3*n*x**(2*n)*(0**(1/n))**(2*n
)), Eq(e, 0**(1/n))), (e**(-2*n)*x**n/n, True))/e, Eq(m, -2*n - 1)), (A*c**
2*Piecewise((log(x), Eq(n, 0)), (-x**(-n)*(0**(1/n))**(-n)/n, Eq(e, 0**(1/n
))), (-e**(-n)*x**(-n)/n, True))/e + 2*A*c*d*Piecewise((e**(-n)*log(x), Abs
(x) < 1), (-e**(-n)*log(1/x), 1/Abs(x) < 1), (-e**(-n)*meijerg(((), (1, 1))
, ((0, 0), ()), x) + e**(-n)*meijerg(((1, 1), ()), ((), (0, 0)), x), True))
/e + A*d**2*Piecewise((log(x), Eq(n, 0)), (-x**(2*n)/(0**(1/n)*zoo**(1/n)*n
*x**n*(0**(1/n))**n - 2*n*x**n*(0**(1/n))**n), Eq(e, 0**(1/n))), (e**(-n)*x
**n/n, True))/e + B*c**2*Piecewise((e**(-n)*log(x), Abs(x) < 1), (-e**(-n)*
log(1/x), 1/Abs(x) < 1), (-e**(-n)*meijerg(((), (1, 1)), ((0, 0), ()), x) +
 e**(-n)*meijerg(((1, 1), ()), ((), (0, 0)), x), True))/e + 2*B*c*d*Piecewi
se((log(x), Eq(n, 0)), (-x**(2*n)/(0**(1/n)*zoo**(1/n)*n*x**n*(0**(1/n))**n
 - 2*n*x**n*(0**(1/n))**n), Eq(e, 0**(1/n))), (e**(-n)*x**n/n, True))/e + B
*d**2*Piecewise((log(x), Eq(n, 0)), (-x**(3*n)/(0**(1/n)*zoo**(1/n)*n*x**n*
(0**(1/n))**n - 3*n*x**n*(0**(1/n))**n), Eq(e, 0**(1/n))), (e**(-n)*x**(2*n
)/(2*n), True))/e, Eq(m, -n - 1)), (A*c**2*e**m*m**3*x*x**m/(m**4 + 6*m**3*
n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*
m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 6*A*c**2*e**m*m**2*n*x*x**m/(m**4
 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*
n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*A*c**2*e**m*m**2*x*x*
*m/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3
 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 11*A*c**2*e**m*
m*n**2*x*x**m/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2
 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 12*A
```

```
*c**2*e**m*m*n*x*x**m/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n
+ 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1
) + 3*A*c**2*e**m*m*x*x**m/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m*
*2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*
n + 1) + 6*A*c**2*e**m*n**3*x*x**m/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2
 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n
**2 + 6*n + 1) + 11*A*c**2*e**m*n**2*x*x**m/(m**4 + 6*m**3*n + 4*m**3 + 11*
m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n*
*3 + 11*n**2 + 6*n + 1) + 6*A*c**2*e**m*n*x*x**m/(m**4 + 6*m**3*n + 4*m**3
+ 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m +
 6*n**3 + 11*n**2 + 6*n + 1) + A*c**2*e**m*x*x**m/(m**4 + 6*m**3*n + 4*m**3
 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m
+ 6*n**3 + 11*n**2 + 6*n + 1) + 2*A*c*d*e**m*m**3*x*x**m*x**n/(m**4 + 6*m**
3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 1
8*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 10*A*c*d*e**m*m**2*n*x*x**m*x**
n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3
+ 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 6*A*c*d*e**m*m**
2*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2
 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 12*A
*c*d*e**m*m*n**2*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*
m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 +
6*n + 1) + 20*A*c*d*e**m*m*n*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**
2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3
+ 11*n**2 + 6*n + 1) + 6*A*c*d*e**m*m*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3
 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m
+ 6*n**3 + 11*n**2 + 6*n + 1) + 12*A*c*d*e**m*n**2*x*x**m*x**n/(m**4 + 6*m*
*3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 +
18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 10*A*c*d*e**m*n*x*x**m*x**n/(m
**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22
*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 2*A*c*d*e**m*x*x**m*
x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n*
*3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + A*d**2*e**m*m
**3*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n +
6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1)
+ 4*A*d**2*e**m*m**2*n*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*
n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 +
11*n**2 + 6*n + 1) + 3*A*d**2*e**m*m**2*x*x**m*x**(2*n)/(m**4 + 6*m**3*n +
4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n
+ 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*A*d**2*e**m*m*n**2*x*x**m*x**(2*n)/
(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 +
22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 8*A*d**2*e**m*m*n*
x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m*
*2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*
A*d**2*e**m*m*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18
*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 +
 6*n + 1) + 3*A*d**2*e**m*n**2*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 +
11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6
*n**3 + 11*n**2 + 6*n + 1) + 4*A*d**2*e**m*n*x*x**m*x**(2*n)/(m**4 + 6*m**3
*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18
*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + A*d**2*e**m*x*x**m*x**(2*n)/(m**
4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m
*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + B*c**2*e**m*m**3*x*x**
m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*
n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 5*B*c**2*e*
*m*m**2*n*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n
+ 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1
) + 3*B*c**2*e**m*m**2*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2
 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n
**2 + 6*n + 1) + 6*B*c**2*e**m*m*n**2*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3
```

```
 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m
+ 6*n**3 + 11*n**2 + 6*n + 1) + 10*B*c**2*e**m*m*n*x*x**m*x**n/(m**4 + 6*m*
*3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 +
18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*B*c**2*e**m*m*x*x**m*x**n/(m
**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22
*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 6*B*c**2*e**m*n**2*x
*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 +
6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 5*B*c**
2*e**m*n*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n +
 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1)
 + B*c**2*e**m*x*x**m*x**n/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m*
*2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*
n + 1) + 2*B*c*d*e**m*m**3*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m
**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**
3 + 11*n**2 + 6*n + 1) + 8*B*c*d*e**m*m**2*n*x*x**m*x**(2*n)/(m**4 + 6*m**3
*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18
*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 6*B*c*d*e**m*m**2*x*x**m*x**(2*n
)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3
+ 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 6*B*c*d*e**m*m*n
**2*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n +
6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1)
+ 16*B*c*d*e**m*m*n*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**
2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*
n**2 + 6*n + 1) + 6*B*c*d*e**m*m*x*x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3
+ 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m +
 6*n**3 + 11*n**2 + 6*n + 1) + 6*B*c*d*e**m*n**2*x*x**m*x**(2*n)/(m**4 + 6*
m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2
+ 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 8*B*c*d*e**m*n*x*x**m*x**(2*
n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3
 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 2*B*c*d*e**m*x*
x**m*x**(2*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2
 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + B*d*
*2*e**m*m**3*x*x**m*x**(3*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*
m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 +
6*n + 1) + 3*B*d**2*e**m*m**2*n*x*x**m*x**(3*n)/(m**4 + 6*m**3*n + 4*m**3 +
 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m +
6*n**3 + 11*n**2 + 6*n + 1) + 3*B*d**2*e**m*m**2*x*x**m*x**(3*n)/(m**4 + 6*
m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2
+ 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 2*B*d**2*e**m*m*n**2*x*x**m*
x**(3*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*
m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 6*B*d**2*
e**m*m*n*x*x**m*x**(3*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2
*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n
+ 1) + 3*B*d**2*e**m*m*x*x**m*x**(3*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*
n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 +
11*n**2 + 6*n + 1) + 2*B*d**2*e**m*n**2*x*x**m*x**(3*n)/(m**4 + 6*m**3*n +
4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*n**2 + 18*m*n
+ 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + 3*B*d**2*e**m*n*x*x**m*x**(3*n)/(m**4
 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n**3 + 22*m*
n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1) + B*d**2*e**m*x*x**m*x**(
3*n)/(m**4 + 6*m**3*n + 4*m**3 + 11*m**2*n**2 + 18*m**2*n + 6*m**2 + 6*m*n*
*3 + 22*m*n**2 + 18*m*n + 4*m + 6*n**3 + 11*n**2 + 6*n + 1), True))
```

3.9 $\int (ex)^m (a+bx^n)^3 (A+Bx^n) (c+dx^n)^3 \, dx$

Optimal. Leaf size=410

$$\frac{a^3Ac^3(ex)^{m+1}}{e(m+1)}+\frac{3acx^{2n+1}(ex)^m\left(A\left(a^2d^2+3abcd+b^2c^2\right)+aBc(ad+bc)\right)}{m+2n+1}+\frac{3bdx^{5n+1}(ex)^m\left(a^2Bd^2+abd(Ad+\right.}{m+5n+1}$$

Rubi [A] time = 0.62, antiderivative size = 410, normalized size of antiderivative = 1.00, number of steps used = 16, number of rules used = 3, integrand size = 31, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.097, Rules used = {570, 20, 30}

$$\frac{3acx^{2n+1}(ex)^m\left(A\left(a^2d^2+3abcd+b^2c^2\right)+aBc(ad+bc)\right)}{m+2n+1}+\frac{x^{3n+1}(ex)^m\left(A\left(9a^2bcd^2+a^3d^3+9ab^2c^2d+b^3c^3\right)+3aBc\left(a^2d^2+3abcd+b^2c^2\right)\right)}{m+3n+1}+\frac{x^{4n+1}(ex)^m\left(3a^2bd^2(Ad+3Bc)+a^3Bd^3+9ab^2cd(Ad+Bc)+b^3c^2(3Ad+Bc)\right)}{m+4n+1}+\frac{3bdx^{5n+1}(ex)^m\left(a^2Bd^2+abd(Ad+3Bc)+b^2c(Ad+Bc)\right)}{m+5n+1}+\frac{a^2c^2x^{n+1}(ex)^m(3A(ad+bc)+aBc)}{m+n+1}+\frac{a^3Ac^3(ex)^{m+1}}{e(m+1)}+\frac{b^2d^2x^{6n+1}(ex)^m(3aBd+Abd+3bBc)}{m+6n+1}+\frac{b^3Bd^3x^{7n+1}(ex)^m}{m+7n+1}$$

Antiderivative was successfully verified.

```
[In] Int[(e*x)^m*(a + b*x^n)^3*(A + B*x^n)*(c + d*x^n)^3,x]
```

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

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 30

```
Int[(x_)^(m_.), x_Symbol] :> Simp[x^(m + 1)/(m + 1), x] /; FreeQ[m, x] && N
eQ[m, -1]
```

Rule 570

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

Rubi steps

$$\int (ex)^m (a+bx^n)^3 (A+Bx^n) (c+dx^n)^3 \, dx = \int \left(a^3Ac^3(ex)^m+a^2c^2(aBc+3A(bc+ad))x^n(ex)^m+3ac\left(aBc(b\right.\right.$$
$$=\frac{a^3Ac^3(ex)^{1+m}}{e(1+m)}+\left(b^3Bd^3\right)\int x^{7n}(ex)^m\,dx+\left(b^2d^2(3bBc+Abd+\right.$$
$$=\frac{a^3Ac^3(ex)^{1+m}}{e(1+m)}+\left(b^3Bd^3x^{-m}(ex)^m\right)\int x^{m+7n}\,dx+\left(b^2d^2(3bBc+\right.$$
$$=\frac{a^2c^2(aBc+3A(bc+ad))x^{1+n}(ex)^m}{1+m+n}+\frac{3ac\left(aBc(bc+ad)+A\left(b^2\right.\right.}{1+}$$

Mathematica [A] time = 1.43, size = 358, normalized size = 0.87

$$x(ex)^m\left(\frac{a^3Ac^3}{m+1}+\frac{3acx^{2n}\left(A\left(a^2d^2+3abcd+b^2c^2\right)+aBc(ad+bc)\right)}{m+2n+1}+\frac{3bdx^{5n}\left(a^2Bd^2+abd(Ad+3Bc)+b^2c(Ad+Bc)\right)}{m+5n+1}+\frac{a^2c^2x^n(3A(ad+bc)+aBc)}{m+n+1}+\frac{x^{4n}\left(a^3Bd^3+3a^2bd^2(Ad+3Bc)+9ab^2cd(Ad+Bc)+b^3c^2(3Ad+Bc)\right)}{m+4n+1}+\frac{x^{3n}\left(3aBc\left(a^2d^2+3abcd+b^2c^2\right)+A\left(a^3d^3+9a^2bcd^2+9ab^2c^2d+b^3c^3\right)\right)}{m+3n+1}+\frac{b^2d^2x^{6n}(3aBd+Abd+3bBc)}{m+6n+1}+\frac{b^3Bd^3x^{7n}}{m+7n+1}\right)$$

Antiderivative was successfully verified.

```
[In] Integrate[(e*x)^m*(a + b*x^n)^3*(A + B*x^n)*(c + d*x^n)^3,x]
```

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

IntegrateAlgebraic [F] time = 1.25, size = 0, normalized size = 0.00

$$\int (ex)^m\left(a+bx^n\right)^3\left(A+Bx^n\right)\left(c+dx^n\right)^3\,dx$$

Verification is not applicable to the result.

```
[In] IntegrateAlgebraic[(e*x)^m*(a + b*x^n)^3*(A + B*x^n)*(c + d*x^n)^3,x]
```

```
[Out] Defer[IntegrateAlgebraic][(e*x)^m*(a + b*x^n)^3*(A + B*x^n)*(c + d*x^n)^3,
x]
```

fricas [B] time = 0.74, size = 11628, normalized size = 28.36

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)^3*(A+B*x^n)*(c+d*x^n)^3,x, algorithm="fricas")
```

```
[Out] ((B*b^3*d^3*m^7 + 7*B*b^3*d^3*m^6 + 21*B*b^3*d^3*m^5 + 35*B*b^3*d^3*m^4 + 3
5*B*b^3*d^3*m^3 + 21*B*b^3*d^3*m^2 + 7*B*b^3*d^3*m + B*b^3*d^3 + 720*(B*b^3
*d^3*m + B*b^3*d^3)*n^6 + 1764*(B*b^3*d^3*m^2 + 2*B*b^3*d^3*m + B*b^3*d^3)*
n^5 + 1624*(B*b^3*d^3*m^3 + 3*B*b^3*d^3*m^2 + 3*B*b^3*d^3*m + B*b^3*d^3)*n^
4 + 735*(B*b^3*d^3*m^4 + 4*B*b^3*d^3*m^3 + 6*B*b^3*d^3*m^2 + 4*B*b^3*d^3*m
+ B*b^3*d^3)*n^3 + 175*(B*b^3*d^3*m^5 + 5*B*b^3*d^3*m^4 + 10*B*b^3*d^3*m^3
+ 10*B*b^3*d^3*m^2 + 5*B*b^3*d^3*m + B*b^3*d^3)*n^2 + 21*(B*b^3*d^3*m^6 + 6
*B*b^3*d^3*m^5 + 15*B*b^3*d^3*m^4 + 20*B*b^3*d^3*m^3 + 15*B*b^3*d^3*m^2 + 6
*B*b^3*d^3*m + B*b^3*d^3)*n)*x*x^(7*n)*e^(m*log(e) + m*log(x)) + ((3*B*b^3*
c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m^7 + 3*B*b^3*c*d^2 + 7*(3*B*b^3*c*d^2 + (
3*B*a*b^2 + A*b^3)*d^3)*m^6 + 840*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3
+ (3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m)*n^6 + 21*(3*B*b^3*c*d^2 + (3
*B*a*b^2 + A*b^3)*d^3)*m^5 + 2038*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3
+ (3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m^2 + 2*(3*B*b^3*c*d^2 + (3*B*a
*b^2 + A*b^3)*d^3)*m)*n^5 + 35*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m^
4 + 1849*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3 + (3*B*b^3*c*d^2 + (3*B*a
*b^2 + A*b^3)*d^3)*m^3 + 3*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m^2 +
3*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m)*n^4 + (3*B*a*b^2 + A*b^3)*d^
3 + 35*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m^3 + 820*(3*B*b^3*c*d^2 +
 (3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m^4 + (3*B*a*b^2 + A*b^3)*d^3 +
4*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m^3 + 6*(3*B*b^3*c*d^2 + (3*B*a
*b^2 + A*b^3)*d^3)*m^2 + 4*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m)*n^3
 + 21*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m^2 + 190*(3*B*b^3*c*d^2 +
(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m^5 + 5*(3*B*b^3*c*d^2 + (3*B*a*b
^2 + A*b^3)*d^3)*m^4 + (3*B*a*b^2 + A*b^3)*d^3 + 10*(3*B*b^3*c*d^2 + (3*B*a
*b^2 + A*b^3)*d^3)*m^3 + 10*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m^2 +
```

```
 5*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m)*n^2 + 7*(3*B*b^3*c*d^2 + (3
*B*a*b^2 + A*b^3)*d^3)*m + 22*(3*B*b^3*c*d^2 + (3*B*b^3*c*d^2 + (3*B*a*b^2
+ A*b^3)*d^3)*m^6 + 6*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m^5 + 15*(3
*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m^4 + (3*B*a*b^2 + A*b^3)*d^3 + 20*
(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m^3 + 15*(3*B*b^3*c*d^2 + (3*B*a*
b^2 + A*b^3)*d^3)*m^2 + 6*(3*B*b^3*c*d^2 + (3*B*a*b^2 + A*b^3)*d^3)*m)*n)*x
*x^(6*n)*e^(m*log(e) + m*log(x)) + 3*((B*b^3*c^2*d + (3*B*a*b^2 + A*b^3)*c*
d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^7 + B*b^3*c^2*d + 7*(B*b^3*c^2*d + (3*B*a*
b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^6 + 1008*(B*b^3*c^2*d + (3*
B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3 + (B*b^3*c^2*d + (3*B*a*b^
2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m)*n^6 + 21*(B*b^3*c^2*d + (3*B
*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^5 + 2412*(B*b^3*c^2*d +
(3*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3 + (B*b^3*c^2*d + (3*B*a
*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^2 + 2*(B*b^3*c^2*d + (3*B*
a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m)*n^5 + 35*(B*b^3*c^2*d +
(3*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^4 + 2144*(B*b^3*c^2*
d + (3*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3 + (B*b^3*c^2*d + (3
*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^3 + 3*(B*b^3*c^2*d + (
3*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^2 + 3*(B*b^3*c^2*d +
(3*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m)*n^4 + (3*B*a*b^2 +
A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3 + 35*(B*b^3*c^2*d + (3*B*a*b^2 + A*b
^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^3 + 925*(B*b^3*c^2*d + (B*b^3*c^2*d
+ (3*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^4 + (3*B*a*b^2 + A
*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3 + 4*(B*b^3*c^2*d + (3*B*a*b^2 + A*b^3
)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^3 + 6*(B*b^3*c^2*d + (3*B*a*b^2 + A*b^
3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^2 + 4*(B*b^3*c^2*d + (3*B*a*b^2 + A*b
^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m)*n^3 + 21*(B*b^3*c^2*d + (3*B*a*b^2
+ A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^2 + 207*(B*b^3*c^2*d + (B*b^3*c
^2*d + (3*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^5 + 5*(B*b^3*
c^2*d + (3*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^4 + (3*B*a*b
^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3 + 10*(B*b^3*c^2*d + (3*B*a*b^2
+ A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^3 + 10*(B*b^3*c^2*d + (3*B*a*b^
2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^2 + 5*(B*b^3*c^2*d + (3*B*a*b
^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m)*n^2 + 7*(B*b^3*c^2*d + (3*B
*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m + 23*(B*b^3*c^2*d + (B*b
^3*c^2*d + (3*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^6 + 6*(B*
b^3*c^2*d + (3*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^5 + 15*(
B*b^3*c^2*d + (3*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^4 + (3
*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3 + 20*(B*b^3*c^2*d + (3*B*
a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^3 + 15*(B*b^3*c^2*d + (3*
B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m^2 + 6*(B*b^3*c^2*d + (3
*B*a*b^2 + A*b^3)*c*d^2 + (B*a^2*b + A*a*b^2)*d^3)*m)*n)*x*x^(5*n)*e^(m*log
(e) + m*log(x)) + ((B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b +
A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m^7 + B*b^3*c^3 + 7*(B*b^3*c^3 +
3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^
2*b)*d^3)*m^6 + 1260*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b
+ A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3 + (B*b^3*c^3 + 3*(3*B*a*b^2 + A*
b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m)*n^6
+ 21*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2
 + (B*a^3 + 3*A*a^2*b)*d^3)*m^5 + 2952*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c
^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3 + (B*b^3*c^3 +
 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a
^2*b)*d^3)*m^2 + 2*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b +
A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m)*n^5 + 35*(B*b^3*c^3 + 3*(3*B*a
*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3
)*m^4 + 2545*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^
2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3 + (B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2
*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m^3 + 3*(B*b^3*
c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 +
```

3*A*a^2*b)*d^3)*m^2 + 3*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2
*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m)*n^4 + 3*(3*B*a*b^2 + A*b^
3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3 + 35*(B*b^
3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3
+ 3*A*a^2*b)*d^3)*m^3 + 1056*(B*b^3*c^3 + (B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3
)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m^4 + 3*(3
*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)
d^3 + 4(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c
*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m^3 + 6*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*
c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m^2 + 4*(B*b
^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3
 + 3*A*a^2*b)*d^3)*m)*n^3 + 21*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9
*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m^2 + 226*(B*b^3*c^3
+ (B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 +
(B*a^3 + 3*A*a^2*b)*d^3)*m^5 + 5*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d +
 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m^4 + 3*(3*B*a*b^2
+ A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3 + 10
*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (
B*a^3 + 3*A*a^2*b)*d^3)*m^3 + 10*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d +
 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m^2 + 5*(B*b^3*c^3
+ 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*
a^2*b)*d^3)*m)*n^2 + 7*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*
b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m + 24*(B*b^3*c^3 + (B*b^3*c^
3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*
A*a^2*b)*d^3)*m^6 + 6*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b
 + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m^5 + 15*(B*b^3*c^3 + 3*(3*B*a
*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3
)*m^4 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3
+ 3*A*a^2*b)*d^3 + 20*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b
 + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m^3 + 15*(B*b^3*c^3 + 3*(3*B*a
*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3
)*m^2 + 6*(B*b^3*c^3 + 3*(3*B*a*b^2 + A*b^3)*c^2*d + 9*(B*a^2*b + A*a*b^2)*
c*d^2 + (B*a^3 + 3*A*a^2*b)*d^3)*m)*n)*x*x^(4*n)*e^(m*log(e) + m*log(x)) +
((A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*
a^3 + 3*A*a^2*b)*c*d^2)*m^7 + A*a^3*d^3 + 7*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3
)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m^6 + 16
80*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(
B*a^3 + 3*A*a^2*b)*c*d^2 + (A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*
b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m)*n^6 + 21*(A*a^3*d^3 +
(3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*
b)*c*d^2)*m^5 + 3796*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*
a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2 + (A*a^3*d^3 + (3*B*a*b^2 + A*b^
3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m^2 + 2
*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*
a^3 + 3*A*a^2*b)*c*d^2)*m)*n^5 + 35*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 +
9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m^4 + 3112*(A*a^
3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 +
3*A*a^2*b)*c*d^2 + (A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*
b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m^3 + 3*(A*a^3*d^3 + (3*B*a*b^2 +
 A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m^
2 + 3*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d +
3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m)*n^4 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b
+ A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2 + 35*(A*a^3*d^3 + (3*B*a*b^2
 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*
m^3 + 1219*(A*a^3*d^3 + (A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b +
 A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m^4 + (3*B*a*b^2 + A*b^3)*c^
3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2 + 4*(A*a^3*d^
3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*
a^2*b)*c*d^2)*m^3 + 6*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A

```
*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m^2 + 4*(A*a^3*d^3 + (3*B*a*b^
2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)
*m)*n^3 + 21*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c
^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m^2 + 247*(A*a^3*d^3 + (A*a^3*d^3 + (3*
B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*
c*d^2)*m^5 + 5*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)
*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m^4 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*
a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2 + 10*(A*a^3*d^3 + (3*B
*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c
*d^2)*m^3 + 10*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)
*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m^2 + 5*(A*a^3*d^3 + (3*B*a*b^2 + A*b
^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m)*n^2
 + 7*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3
*(B*a^3 + 3*A*a^2*b)*c*d^2)*m + 25*(A*a^3*d^3 + (A*a^3*d^3 + (3*B*a*b^2 + A
*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m^6
+ 6*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*
(B*a^3 + 3*A*a^2*b)*c*d^2)*m^5 + 15*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 +
9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m^4 + (3*B*a*b^2
 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2 +
 20*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*
(B*a^3 + 3*A*a^2*b)*c*d^2)*m^3 + 15*(A*a^3*d^3 + (3*B*a*b^2 + A*b^3)*c^3 +
9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A*a^2*b)*c*d^2)*m^2 + 6*(A*a^3*d
^3 + (3*B*a*b^2 + A*b^3)*c^3 + 9*(B*a^2*b + A*a*b^2)*c^2*d + 3*(B*a^3 + 3*A
*a^2*b)*c*d^2)*m)*n)*x*x^(3*n)*e^(m*log(e) + m*log(x)) + 3*((A*a^3*c*d^2 +
(B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m^7 + A*a^3*c*d^2 + 7*
(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m^6 + 2
520*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d + (A
*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m)*n^6 +
21*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m^5
+ 5274*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d +
 (A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m^2 +
2*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m)*n^
5 + 35*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*
m^4 + 3929*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2
*d + (A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m^
3 + 3*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m
^2 + 3*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*
m)*n^4 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d + 35*(A*a^3*c*
d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m^3 + 1420*(A*a^
3*c*d^2 + (A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*
d)*m^4 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d + 4*(A*a^3*c*d
^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m^3 + 6*(A*a^3*c*
d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m^2 + 4*(A*a^3*c
*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m)*n^3 + 21*(A*
a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m^2 + 270*
(A*a^3*c*d^2 + (A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)
*c^2*d)*m^5 + 5*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b
)*c^2*d)*m^4 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d + 10*(A*
a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m^3 + 10*(
A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m^2 + 5*
(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m)*n^2
+ 7*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m +
 26*(A*a^3*c*d^2 + (A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^
2*b)*c^2*d)*m^6 + 6*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a
^2*b)*c^2*d)*m^5 + 15*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A
*a^2*b)*c^2*d)*m^4 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d +
20*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m^3
+ 15*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m^
2 + 6*(A*a^3*c*d^2 + (B*a^2*b + A*a*b^2)*c^3 + (B*a^3 + 3*A*a^2*b)*c^2*d)*m
```

```
)*n)*x*x^(2*n)*e^(m*log(e) + m*log(x)) + ((3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2
*b)*c^3)*m^7 + 3*A*a^3*c^2*d + 7*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*
m^6 + 5040*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3 + (3*A*a^3*c^2*d + (B*a
^3 + 3*A*a^2*b)*c^3)*m)*n^6 + 21*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*
m^5 + 8028*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3 + (3*A*a^3*c^2*d + (B*a
^3 + 3*A*a^2*b)*c^3)*m^2 + 2*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*m)*n
^5 + 35*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*m^4 + 5104*(3*A*a^3*c^2*d
 + (B*a^3 + 3*A*a^2*b)*c^3 + (3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*m^3
+ 3*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*m^2 + 3*(3*A*a^3*c^2*d + (B*a
^3 + 3*A*a^2*b)*c^3)*m)*n^4 + (B*a^3 + 3*A*a^2*b)*c^3 + 35*(3*A*a^3*c^2*d +
 (B*a^3 + 3*A*a^2*b)*c^3)*m^3 + 1665*(3*A*a^3*c^2*d + (3*A*a^3*c^2*d + (B*a
^3 + 3*A*a^2*b)*c^3)*m^4 + (B*a^3 + 3*A*a^2*b)*c^3 + 4*(3*A*a^3*c^2*d + (B*
a^3 + 3*A*a^2*b)*c^3)*m^3 + 6*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*m^2
 + 4*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*m)*n^3 + 21*(3*A*a^3*c^2*d +
 (B*a^3 + 3*A*a^2*b)*c^3)*m^2 + 295*(3*A*a^3*c^2*d + (3*A*a^3*c^2*d + (B*a^
3 + 3*A*a^2*b)*c^3)*m^5 + 5*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*m^4 +
 (B*a^3 + 3*A*a^2*b)*c^3 + 10*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*m^3
 + 10*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*m^2 + 5*(3*A*a^3*c^2*d + (B
*a^3 + 3*A*a^2*b)*c^3)*m)*n^2 + 7*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)
*m + 27*(3*A*a^3*c^2*d + (3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*m^6 + 6*
(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*m^5 + 15*(3*A*a^3*c^2*d + (B*a^3
+ 3*A*a^2*b)*c^3)*m^4 + (B*a^3 + 3*A*a^2*b)*c^3 + 20*(3*A*a^3*c^2*d + (B*a^
3 + 3*A*a^2*b)*c^3)*m^3 + 15*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*m^2
+ 6*(3*A*a^3*c^2*d + (B*a^3 + 3*A*a^2*b)*c^3)*m)*n)*x*x^n*e^(m*log(e) + m*l
og(x)) + (A*a^3*c^3*m^7 + 5040*A*a^3*c^3*n^7 + 7*A*a^3*c^3*m^6 + 21*A*a^3*c
^3*m^5 + 35*A*a^3*c^3*m^4 + 35*A*a^3*c^3*m^3 + 21*A*a^3*c^3*m^2 + 7*A*a^3*c
^3*m + A*a^3*c^3 + 13068*(A*a^3*c^3*m + A*a^3*c^3)*n^6 + 13132*(A*a^3*c^3*m
^2 + 2*A*a^3*c^3*m + A*a^3*c^3)*n^5 + 6769*(A*a^3*c^3*m^3 + 3*A*a^3*c^3*m^2
 + 3*A*a^3*c^3*m + A*a^3*c^3)*n^4 + 1960*(A*a^3*c^3*m^4 + 4*A*a^3*c^3*m^3 +
 6*A*a^3*c^3*m^2 + 4*A*a^3*c^3*m + A*a^3*c^3)*n^3 + 322*(A*a^3*c^3*m^5 + 5*
A*a^3*c^3*m^4 + 10*A*a^3*c^3*m^3 + 10*A*a^3*c^3*m^2 + 5*A*a^3*c^3*m + A*a^3
*c^3)*n^2 + 28*(A*a^3*c^3*m^6 + 6*A*a^3*c^3*m^5 + 15*A*a^3*c^3*m^4 + 20*A*a
^3*c^3*m^3 + 15*A*a^3*c^3*m^2 + 6*A*a^3*c^3*m + A*a^3*c^3)*n)*x*e^(m*log(e)
 + m*log(x)))/(m^8 + 5040*(m + 1)*n^7 + 8*m^7 + 13068*(m^2 + 2*m + 1)*n^6 +
 28*m^6 + 13132*(m^3 + 3*m^2 + 3*m + 1)*n^5 + 56*m^5 + 6769*(m^4 + 4*m^3 +
6*m^2 + 4*m + 1)*n^4 + 70*m^4 + 1960*(m^5 + 5*m^4 + 10*m^3 + 10*m^2 + 5*m +
 1)*n^3 + 56*m^3 + 322*(m^6 + 6*m^5 + 15*m^4 + 20*m^3 + 15*m^2 + 6*m + 1)*n
^2 + 28*m^2 + 28*(m^7 + 7*m^6 + 21*m^5 + 35*m^4 + 35*m^3 + 21*m^2 + 7*m + 1
)*n + 8*m + 1)
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)^3*(A+B*x^n)*(c+d*x^n)^3,x, algorithm="giac")
```

```
[Out] Timed out
```

maple [C] time = 0.34, size = 20937, normalized size = 51.07

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(b*x^n+a)^3*(B*x^n+A)*(d*x^n+c)^3,x)
```

```
[Out] result too large to display
```

maxima [B] time = 1.47, size = 1032, normalized size = 2.52

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)^3*(A+B*x^n)*(c+d*x^n)^3,x, algorithm="maxima")
```

```
[Out] B*b^3*d^3*e^m*x*e^(m*log(x) + 7*n*log(x))/(m + 7*n + 1) + 3*B*b^3*c*d^2*e^m
*x*e^(m*log(x) + 6*n*log(x))/(m + 6*n + 1) + 3*B*a*b^2*d^3*e^m*x*e^(m*log(x
) + 6*n*log(x))/(m + 6*n + 1) + A*b^3*d^3*e^m*x*e^(m*log(x) + 6*n*log(x))/(
m + 6*n + 1) + 3*B*b^3*c^2*d*e^m*x*e^(m*log(x) + 5*n*log(x))/(m + 5*n + 1)
+ 9*B*a*b^2*c*d^2*e^m*x*e^(m*log(x) + 5*n*log(x))/(m + 5*n + 1) + 3*A*b^3*c
*d^2*e^m*x*e^(m*log(x) + 5*n*log(x))/(m + 5*n + 1) + 3*B*a^2*b*d^3*e^m*x*e^
(m*log(x) + 5*n*log(x))/(m + 5*n + 1) + 3*A*a*b^2*d^3*e^m*x*e^(m*log(x) + 5
*n*log(x))/(m + 5*n + 1) + B*b^3*c^3*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4
*n + 1) + 9*B*a*b^2*c^2*d*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) + 3
*A*b^3*c^2*d*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) + 9*B*a^2*b*c*d^
2*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) + 9*A*a*b^2*c*d^2*e^m*x*e^(
m*log(x) + 4*n*log(x))/(m + 4*n + 1) + B*a^3*d^3*e^m*x*e^(m*log(x) + 4*n*lo
g(x))/(m + 4*n + 1) + 3*A*a^2*b*d^3*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*
n + 1) + 3*B*a*b^2*c^3*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + A*b^
3*c^3*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + 9*B*a^2*b*c^2*d*e^m*x
*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + 9*A*a*b^2*c^2*d*e^m*x*e^(m*log(x
) + 3*n*log(x))/(m + 3*n + 1) + 3*B*a^3*c*d^2*e^m*x*e^(m*log(x) + 3*n*log(x
))/(m + 3*n + 1) + 9*A*a^2*b*c*d^2*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n
 + 1) + A*a^3*d^3*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + 3*B*a^2*b
*c^3*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 3*A*a*b^2*c^3*e^m*x*e^
(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 3*B*a^3*c^2*d*e^m*x*e^(m*log(x) + 2
*n*log(x))/(m + 2*n + 1) + 9*A*a^2*b*c^2*d*e^m*x*e^(m*log(x) + 2*n*log(x))/
(m + 2*n + 1) + 3*A*a^3*c*d^2*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1)
 + B*a^3*c^3*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1) + 3*A*a^2*b*c^3*e^m*
x*e^(m*log(x) + n*log(x))/(m + n + 1) + 3*A*a^3*c^2*d*e^m*x*e^(m*log(x) + n
*log(x))/(m + n + 1) + (e*x)^(m + 1)*A*a^3*c^3/(e*(m + 1))
```

mupad [B] time = 7.49, size = 2949, normalized size = 7.19

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(A + B*x^n)*(a + b*x^n)^3*(c + d*x^n)^3,x)
```

```
[Out] (x*x^(3*n)*(e*x)^m*(A*a^3*d^3 + A*b^3*c^3 + 3*B*a*b^2*c^3 + 3*B*a^3*c*d^2 +
 9*A*a*b^2*c^2*d + 9*A*a^2*b*c*d^2 + 9*B*a^2*b*c^2*d)*(6*m + 25*n + 125*m*n
 + 988*m*n^2 + 250*m^2*n + 3657*m*n^3 + 250*m^3*n + 6224*m*n^4 + 125*m^4*n
+ 3796*m*n^5 + 25*m^5*n + 15*m^2 + 20*m^3 + 15*m^4 + 6*m^5 + m^6 + 247*n^2
+ 1219*n^3 + 3112*n^4 + 3796*n^5 + 1680*n^6 + 1482*m^2*n^2 + 3657*m^2*n^3 +
 988*m^3*n^2 + 3112*m^2*n^4 + 1219*m^3*n^3 + 247*m^4*n^2 + 1))/(7*m + 28*n
+ 168*m*n + 1610*m*n^2 + 420*m^2*n + 7840*m*n^3 + 560*m^3*n + 20307*m*n^4 +
 420*m^4*n + 26264*m*n^5 + 168*m^5*n + 13068*m*n^6 + 28*m^6*n + 21*m^2 + 35
*m^3 + 35*m^4 + 21*m^5 + 7*m^6 + m^7 + 322*n^2 + 1960*n^3 + 6769*n^4 + 1313
2*n^5 + 13068*n^6 + 5040*n^7 + 3220*m^2*n^2 + 11760*m^2*n^3 + 3220*m^3*n^2
+ 20307*m^2*n^4 + 7840*m^3*n^3 + 1610*m^4*n^2 + 13132*m^2*n^5 + 6769*m^3*n^
4 + 1960*m^4*n^3 + 322*m^5*n^2 + 1) + (x*x^(4*n)*(e*x)^m*(B*a^3*d^3 + B*b^3
*c^3 + 3*A*a^2*b*d^3 + 3*A*b^3*c^2*d + 9*A*a*b^2*c*d^2 + 9*B*a*b^2*c^2*d +
9*B*a^2*b*c*d^2)*(6*m + 24*n + 120*m*n + 904*m*n^2 + 240*m^2*n + 3168*m*n^3
 + 240*m^3*n + 5090*m*n^4 + 120*m^4*n + 2952*m*n^5 + 24*m^5*n + 15*m^2 + 20
*m^3 + 15*m^4 + 6*m^5 + m^6 + 226*n^2 + 1056*n^3 + 2545*n^4 + 2952*n^5 + 12
60*n^6 + 1356*m^2*n^2 + 3168*m^2*n^3 + 904*m^3*n^2 + 2545*m^2*n^4 + 1056*m^
3*n^3 + 226*m^4*n^2 + 1))/(7*m + 28*n + 168*m*n + 1610*m*n^2 + 420*m^2*n +
7840*m*n^3 + 560*m^3*n + 20307*m*n^4 + 420*m^4*n + 26264*m*n^5 + 168*m^5*n
+ 13068*m*n^6 + 28*m^6*n + 21*m^2 + 35*m^3 + 35*m^4 + 21*m^5 + 7*m^6 + m^7
+ 322*n^2 + 1960*n^3 + 6769*n^4 + 13132*n^5 + 13068*n^6 + 5040*n^7 + 3220*m
```

```
^2*n^2 + 11760*m^2*n^3 + 3220*m^3*n^2 + 20307*m^2*n^4 + 7840*m^3*n^3 + 1610
*m^4*n^2 + 13132*m^2*n^5 + 6769*m^3*n^4 + 1960*m^4*n^3 + 322*m^5*n^2 + 1) +
 (A*a^3*c^3*x*(e*x)^m)/(m + 1) + (a^2*c^2*x*x^n*(e*x)^m*(3*A*a*d + 3*A*b*c
+ B*a*c)*(6*m + 27*n + 135*m*n + 1180*m*n^2 + 270*m^2*n + 4995*m*n^3 + 270*
m^3*n + 10208*m*n^4 + 135*m^4*n + 8028*m*n^5 + 27*m^5*n + 15*m^2 + 20*m^3 +
 15*m^4 + 6*m^5 + m^6 + 295*n^2 + 1665*n^3 + 5104*n^4 + 8028*n^5 + 5040*n^6
 + 1770*m^2*n^2 + 4995*m^2*n^3 + 1180*m^3*n^2 + 5104*m^2*n^4 + 1665*m^3*n^3
 + 295*m^4*n^2 + 1))/(7*m + 28*n + 168*m*n + 1610*m*n^2 + 420*m^2*n + 7840*
m*n^3 + 560*m^3*n + 20307*m*n^4 + 420*m^4*n + 26264*m*n^5 + 168*m^5*n + 130
68*m*n^6 + 28*m^6*n + 21*m^2 + 35*m^3 + 35*m^4 + 21*m^5 + 7*m^6 + m^7 + 322
*n^2 + 1960*n^3 + 6769*n^4 + 13132*n^5 + 13068*n^6 + 5040*n^7 + 3220*m^2*n^
2 + 11760*m^2*n^3 + 3220*m^3*n^2 + 20307*m^2*n^4 + 7840*m^3*n^3 + 1610*m^4*
n^2 + 13132*m^2*n^5 + 6769*m^3*n^4 + 1960*m^4*n^3 + 322*m^5*n^2 + 1) + (B*b
^3*d^3*x*x^(7*n)*(e*x)^m*(6*m + 21*n + 105*m*n + 700*m*n^2 + 210*m^2*n + 22
05*m*n^3 + 210*m^3*n + 3248*m*n^4 + 105*m^4*n + 1764*m*n^5 + 21*m^5*n + 15*
m^2 + 20*m^3 + 15*m^4 + 6*m^5 + m^6 + 175*n^2 + 735*n^3 + 1624*n^4 + 1764*n
^5 + 720*n^6 + 1050*m^2*n^2 + 2205*m^2*n^3 + 700*m^3*n^2 + 1624*m^2*n^4 + 7
35*m^3*n^3 + 175*m^4*n^2 + 1))/(7*m + 28*n + 168*m*n + 1610*m*n^2 + 420*m^2
*n + 7840*m*n^3 + 560*m^3*n + 20307*m*n^4 + 420*m^4*n + 26264*m*n^5 + 168*m
^5*n + 13068*m*n^6 + 28*m^6*n + 21*m^2 + 35*m^3 + 35*m^4 + 21*m^5 + 7*m^6 +
 m^7 + 322*n^2 + 1960*n^3 + 6769*n^4 + 13132*n^5 + 13068*n^6 + 5040*n^7 + 3
220*m^2*n^2 + 11760*m^2*n^3 + 3220*m^3*n^2 + 20307*m^2*n^4 + 7840*m^3*n^3 +
 1610*m^4*n^2 + 13132*m^2*n^5 + 6769*m^3*n^4 + 1960*m^4*n^3 + 322*m^5*n^2 +
 1) + (3*a*c*x*x^(2*n)*(e*x)^m*(A*a^2*d^2 + A*b^2*c^2 + B*a*b*c^2 + B*a^2*c
*d + 3*A*a*b*c*d)*(6*m + 26*n + 130*m*n + 1080*m*n^2 + 260*m^2*n + 4260*m*n
^3 + 260*m^3*n + 7858*m*n^4 + 130*m^4*n + 5274*m*n^5 + 26*m^5*n + 15*m^2 +
20*m^3 + 15*m^4 + 6*m^5 + m^6 + 270*n^2 + 1420*n^3 + 3929*n^4 + 5274*n^5 +
2520*n^6 + 1620*m^2*n^2 + 4260*m^2*n^3 + 1080*m^3*n^2 + 3929*m^2*n^4 + 1420
*m^3*n^3 + 270*m^4*n^2 + 1))/(7*m + 28*n + 168*m*n + 1610*m*n^2 + 420*m^2*n
 + 7840*m*n^3 + 560*m^3*n + 20307*m*n^4 + 420*m^4*n + 26264*m*n^5 + 168*m^5
*n + 13068*m*n^6 + 28*m^6*n + 21*m^2 + 35*m^3 + 35*m^4 + 21*m^5 + 7*m^6 + m
^7 + 322*n^2 + 1960*n^3 + 6769*n^4 + 13132*n^5 + 13068*n^6 + 5040*n^7 + 322
0*m^2*n^2 + 11760*m^2*n^3 + 3220*m^3*n^2 + 20307*m^2*n^4 + 7840*m^3*n^3 + 1
610*m^4*n^2 + 13132*m^2*n^5 + 6769*m^3*n^4 + 1960*m^4*n^3 + 322*m^5*n^2 + 1
) + (3*b*d*x*x^(5*n)*(e*x)^m*(B*a^2*d^2 + B*b^2*c^2 + A*a*b*d^2 + A*b^2*c*d
 + 3*B*a*b*c*d)*(6*m + 23*n + 115*m*n + 828*m*n^2 + 230*m^2*n + 2775*m*n^3
+ 230*m^3*n + 4288*m*n^4 + 115*m^4*n + 2412*m*n^5 + 23*m^5*n + 15*m^2 + 20*
m^3 + 15*m^4 + 6*m^5 + m^6 + 207*n^2 + 925*n^3 + 2144*n^4 + 2412*n^5 + 1008
*n^6 + 1242*m^2*n^2 + 2775*m^2*n^3 + 828*m^3*n^2 + 2144*m^2*n^4 + 925*m^3*n
^3 + 207*m^4*n^2 + 1))/(7*m + 28*n + 168*m*n + 1610*m*n^2 + 420*m^2*n + 784
0*m*n^3 + 560*m^3*n + 20307*m*n^4 + 420*m^4*n + 26264*m*n^5 + 168*m^5*n + 1
3068*m*n^6 + 28*m^6*n + 21*m^2 + 35*m^3 + 35*m^4 + 21*m^5 + 7*m^6 + m^7 + 3
22*n^2 + 1960*n^3 + 6769*n^4 + 13132*n^5 + 13068*n^6 + 5040*n^7 + 3220*m^2*
n^2 + 11760*m^2*n^3 + 3220*m^3*n^2 + 20307*m^2*n^4 + 7840*m^3*n^3 + 1610*m^
4*n^2 + 13132*m^2*n^5 + 6769*m^3*n^4 + 1960*m^4*n^3 + 322*m^5*n^2 + 1) + (b
^2*d^2*x*x^(6*n)*(e*x)^m*(A*b*d + 3*B*a*d + 3*B*b*c)*(6*m + 22*n + 110*m*n
+ 760*m*n^2 + 220*m^2*n + 2460*m*n^3 + 220*m^3*n + 3698*m*n^4 + 110*m^4*n +
 2038*m*n^5 + 22*m^5*n + 15*m^2 + 20*m^3 + 15*m^4 + 6*m^5 + m^6 + 190*n^2 +
 820*n^3 + 1849*n^4 + 2038*n^5 + 840*n^6 + 1140*m^2*n^2 + 2460*m^2*n^3 + 76
0*m^3*n^2 + 1849*m^2*n^4 + 820*m^3*n^3 + 190*m^4*n^2 + 1))/(7*m + 28*n + 16
8*m*n + 1610*m*n^2 + 420*m^2*n + 7840*m*n^3 + 560*m^3*n + 20307*m*n^4 + 420
*m^4*n + 26264*m*n^5 + 168*m^5*n + 13068*m*n^6 + 28*m^6*n + 21*m^2 + 35*m^3
 + 35*m^4 + 21*m^5 + 7*m^6 + m^7 + 322*n^2 + 1960*n^3 + 6769*n^4 + 13132*n^
5 + 13068*n^6 + 5040*n^7 + 3220*m^2*n^2 + 11760*m^2*n^3 + 3220*m^3*n^2 + 20
307*m^2*n^4 + 7840*m^3*n^3 + 1610*m^4*n^2 + 13132*m^2*n^5 + 6769*m^3*n^4 +
1960*m^4*n^3 + 322*m^5*n^2 + 1)
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)**m*(a+b*x**n)**3*(A+B*x**n)*(c+d*x**n)**3,x)
```

```
[Out] Timed out
```

3.10 $\int (ex)^m (a+bx^n)^2 (A+Bx^n) (c+dx^n)^3 \, dx$

Optimal. Leaf size=310

$$\frac{cx^{2n+1}(ex)^m \left(A\left(3a^2d^2+6abcd+b^2c^2\right)+aBc(3ad+2bc)\right)}{m+2n+1}+\frac{x^{3n+1}(ex)^m \left(a^2d^2(Ad+3Bc)+6abcd(Ad+Bc)+b^2c\right.}{m+3n+1}$$

Rubi [A] time = 0.41, antiderivative size = 310, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 3, integrand size = 31, $\frac{\text{number of rules}}{\text{integrand size}} = 0.097$, Rules used = {570, 20, 30}

$$\frac{cx^{2n+1}(ex)^m \left(A\left(3a^2d^2+6abcd+b^2c^2\right)+aBc(3ad+2bc)\right)}{m+2n+1}+\frac{x^{3n+1}(ex)^m \left(a^2d^2(Ad+3Bc)+6abcd(Ad+Bc)+b^2c^2(3Ad+Bc)\right)}{m+3n+1}+\frac{dx^{4n+1}(ex)^m \left(a^2Bd^2+2abd(Ad+3Bc)+3b^2c(Ad+Bc)\right)}{m+4n+1}+\frac{a^2Ac^3(ex)^{m+1}}{e(m+1)}+\frac{ac^2x^{n+1}(ex)^m(3aAd+aBc+2Abc)}{m+n+1}+\frac{bd^2x^{5n+1}(ex)^m(2aBd+Abd+3bBc)}{m+5n+1}+\frac{b^2Bd^3x^{6n+1}(ex)^m}{m+6n+1}$$

Antiderivative was successfully verified.

```
[In] Int[(e*x)^m*(a + b*x^n)^2*(A + B*x^n)*(c + d*x^n)^3,x]
```

```
[Out] (a*c^2*(2*A*b*c + a*B*c + 3*a*A*d)*x^(1 + n)*(e*x)^m)/(1 + m + n) + (c*(a*B
*c*(2*b*c + 3*a*d) + A*(b^2*c^2 + 6*a*b*c*d + 3*a^2*d^2))*x^(1 + 2*n)*(e*x)
^m)/(1 + m + 2*n) + ((6*a*b*c*d*(B*c + A*d) + a^2*d^2*(3*B*c + A*d) + b^2*c
^2*(B*c + 3*A*d))*x^(1 + 3*n)*(e*x)^m)/(1 + m + 3*n) + (d*(a^2*B*d^2 + 3*b^
2*c*(B*c + A*d) + 2*a*b*d*(3*B*c + A*d))*x^(1 + 4*n)*(e*x)^m)/(1 + m + 4*n)
 + (b*d^2*(3*b*B*c + A*b*d + 2*a*B*d)*x^(1 + 5*n)*(e*x)^m)/(1 + m + 5*n) +
(b^2*B*d^3*x^(1 + 6*n)*(e*x)^m)/(1 + m + 6*n) + (a^2*A*c^3*(e*x)^(1 + m))/(
e*(1 + m))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 30

```
Int[(x_)^(m_.), x_Symbol] :> Simp[x^(m + 1)/(m + 1), x] /; FreeQ[m, x] && N
eQ[m, -1]
```

Rule 570

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

Rubi steps

$$\int (ex)^m (a+bx^n)^2 (A+Bx^n) (c+dx^n)^3 \, dx = \int \left(a^2Ac^3(ex)^m + ac^2(2Abc+aBc+3aAd)x^n(ex)^m + c\left(aBc(2bc+\right.\right.$$
$$= \frac{a^2Ac^3(ex)^{1+m}}{e(1+m)} + \left(b^2Bd^3\right)\int x^{6n}(ex)^m \, dx + \left(ac^2(2Abc+aBc+3aAd\right.$$
$$= \frac{a^2Ac^3(ex)^{1+m}}{e(1+m)} + \left(b^2Bd^3x^{-m}(ex)^m\right)\int x^{m+6n} \, dx + \left(ac^2(2Abc+aBc\right.$$
$$= \frac{ac^2(2Abc+aBc+3aAd)x^{1+n}(ex)^m}{1+m+n} + \frac{c\left(aBc(2bc+3ad)+A\left(b^2c^2\right.\right.}{1+m}$$

Mathematica [A] time = 1.52, size = 265, normalized size = 0.85

$$x(ex)^m\left(\frac{cx^{2n}\left(A\left(3a^2d^2+6abcd+b^2c^2\right)+aBc(3ad+2bc)\right)}{m+2n+1}+\frac{x^{3n}\left(a^2d^2(Ad+3Bc)+6abcd(Ad+Bc)+b^2c^2(3Ad+Bc)\right)}{m+3n+1}+\frac{dx^{4n}\left(a^2Bd^2+2abd(Ad+3Bc)+3b^2c(Ad+Bc)\right)}{m+4n+1}+\frac{a^2Ac^3}{m+1}+\frac{ac^2x^n(3aAd+aBc+2Abc)}{m+n+1}+\frac{bd^2x^{5n}(2aBd+Abd+3bBc)}{m+5n+1}+\frac{b^2Bd^3x^{6n}}{m+6n+1}\right)$$

Antiderivative was successfully verified.

```
[In] Integrate[(e*x)^m*(a + b*x^n)^2*(A + B*x^n)*(c + d*x^n)^3,x]
```

```
[Out] x*(e*x)^m*((a^2*A*c^3)/(1 + m) + (a*c^2*(2*A*b*c + a*B*c + 3*a*A*d)*x^n)/(1
 + m + n) + (c*(a*B*c*(2*b*c + 3*a*d) + A*(b^2*c^2 + 6*a*b*c*d + 3*a^2*d^2)
)*x^(2*n))/(1 + m + 2*n) + ((6*a*b*c*d*(B*c + A*d) + a^2*d^2*(3*B*c + A*d)
+ b^2*c^2*(B*c + 3*A*d))*x^(3*n))/(1 + m + 3*n) + (d*(a^2*B*d^2 + 3*b^2*c*(
B*c + A*d) + 2*a*b*d*(3*B*c + A*d))*x^(4*n))/(1 + m + 4*n) + (b*d^2*(3*b*B*
c + A*b*d + 2*a*B*d)*x^(5*n))/(1 + m + 5*n) + (b^2*B*d^3*x^(6*n))/(1 + m +
6*n))
```

IntegrateAlgebraic [F] time = 1.01, size = 0, normalized size = 0.00

$$\int (ex)^m\left(a+bx^n\right)^2\left(A+Bx^n\right)\left(c+dx^n\right)^3\,dx$$

Verification is not applicable to the result.

```
[In] IntegrateAlgebraic[(e*x)^m*(a + b*x^n)^2*(A + B*x^n)*(c + d*x^n)^3,x]
```

```
[Out] Defer[IntegrateAlgebraic][(e*x)^m*(a + b*x^n)^2*(A + B*x^n)*(c + d*x^n)^3,
x]
```

fricas [B] time = 0.58, size = 6557, normalized size = 21.15

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

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

```
[Out] ((B*b^2*d^3*m^6 + 6*B*b^2*d^3*m^5 + 15*B*b^2*d^3*m^4 + 20*B*b^2*d^3*m^3 + 1
5*B*b^2*d^3*m^2 + 6*B*b^2*d^3*m + B*b^2*d^3 + 120*(B*b^2*d^3*m + B*b^2*d^3)
*n^5 + 274*(B*b^2*d^3*m^2 + 2*B*b^2*d^3*m + B*b^2*d^3)*n^4 + 225*(B*b^2*d^3
*m^3 + 3*B*b^2*d^3*m^2 + 3*B*b^2*d^3*m + B*b^2*d^3)*n^3 + 85*(B*b^2*d^3*m^4
 + 4*B*b^2*d^3*m^3 + 6*B*b^2*d^3*m^2 + 4*B*b^2*d^3*m + B*b^2*d^3)*n^2 + 15*
(B*b^2*d^3*m^5 + 5*B*b^2*d^3*m^4 + 10*B*b^2*d^3*m^3 + 10*B*b^2*d^3*m^2 + 5*
B*b^2*d^3*m + B*b^2*d^3)*n)*x*x^(6*n)*e^(m*log(e) + m*log(x)) + ((3*B*b^2*c
*d^2 + (2*B*a*b + A*b^2)*d^3)*m^6 + 3*B*b^2*c*d^2 + 6*(3*B*b^2*c*d^2 + (2*B
*a*b + A*b^2)*d^3)*m^5 + 144*(3*B*b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3 + (3*B*
b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3)*m)*n^5 + 15*(3*B*b^2*c*d^2 + (2*B*a*b +
A*b^2)*d^3)*m^4 + 324*(3*B*b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3 + (3*B*b^2*c*d
^2 + (2*B*a*b + A*b^2)*d^3)*m^2 + 2*(3*B*b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3)
*m)*n^4 + (2*B*a*b + A*b^2)*d^3 + 20*(3*B*b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3
)*m^3 + 260*(3*B*b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3 + (3*B*b^2*c*d^2 + (2*B*
a*b + A*b^2)*d^3)*m^3 + 3*(3*B*b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3)*m^2 + 3*(
3*B*b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3)*m)*n^3 + 15*(3*B*b^2*c*d^2 + (2*B*a*
b + A*b^2)*d^3)*m^2 + 95*(3*B*b^2*c*d^2 + (3*B*b^2*c*d^2 + (2*B*a*b + A*b^2
)*d^3)*m^4 + (2*B*a*b + A*b^2)*d^3 + 4*(3*B*b^2*c*d^2 + (2*B*a*b + A*b^2)*d
^3)*m^3 + 6*(3*B*b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3)*m^2 + 4*(3*B*b^2*c*d^2
+ (2*B*a*b + A*b^2)*d^3)*m)*n^2 + 6*(3*B*b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3)
*m + 16*(3*B*b^2*c*d^2 + (3*B*b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3)*m^5 + 5*(3
*B*b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3)*m^4 + (2*B*a*b + A*b^2)*d^3 + 10*(3*B
*b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3)*m^3 + 10*(3*B*b^2*c*d^2 + (2*B*a*b + A*
b^2)*d^3)*m^2 + 5*(3*B*b^2*c*d^2 + (2*B*a*b + A*b^2)*d^3)*m)*n)*x*x^(5*n)*e
^(m*log(e) + m*log(x)) + ((3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2 + (B*a
^2 + 2*A*a*b)*d^3)*m^6 + 3*B*b^2*c^2*d + 6*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*
b^2)*c*d^2 + (B*a^2 + 2*A*a*b)*d^3)*m^5 + 180*(3*B*b^2*c^2*d + 3*(2*B*a*b +
```

```
 A*b^2)*c*d^2 + (B*a^2 + 2*A*a*b)*d^3 + (3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2
)*c*d^2 + (B*a^2 + 2*A*a*b)*d^3)*m)*n^5 + 15*(3*B*b^2*c^2*d + 3*(2*B*a*b +
A*b^2)*c*d^2 + (B*a^2 + 2*A*a*b)*d^3)*m^4 + 396*(3*B*b^2*c^2*d + 3*(2*B*a*b
 + A*b^2)*c*d^2 + (B*a^2 + 2*A*a*b)*d^3 + (3*B*b^2*c^2*d + 3*(2*B*a*b + A*b
^2)*c*d^2 + (B*a^2 + 2*A*a*b)*d^3)*m^2 + 2*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*
b^2)*c*d^2 + (B*a^2 + 2*A*a*b)*d^3)*m)*n^4 + 3*(2*B*a*b + A*b^2)*c*d^2 + (B
*a^2 + 2*A*a*b)*d^3 + 20*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2 + (B*a^
2 + 2*A*a*b)*d^3)*m^3 + 307*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2 + (B
*a^2 + 2*A*a*b)*d^3 + (3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2 + (B*a^2 +
 2*A*a*b)*d^3)*m^3 + 3*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2 + (B*a^2
+ 2*A*a*b)*d^3)*m^2 + 3*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2 + (B*a^2
 + 2*A*a*b)*d^3)*m)*n^3 + 15*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2 + (
B*a^2 + 2*A*a*b)*d^3)*m^2 + 107*(3*B*b^2*c^2*d + (3*B*b^2*c^2*d + 3*(2*B*a*
b + A*b^2)*c*d^2 + (B*a^2 + 2*A*a*b)*d^3)*m^4 + 3*(2*B*a*b + A*b^2)*c*d^2 +
 (B*a^2 + 2*A*a*b)*d^3 + 4*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2 + (B*
a^2 + 2*A*a*b)*d^3)*m^3 + 6*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2 + (B
*a^2 + 2*A*a*b)*d^3)*m^2 + 4*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2 + (
B*a^2 + 2*A*a*b)*d^3)*m)*n^2 + 6*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2
 + (B*a^2 + 2*A*a*b)*d^3)*m + 17*(3*B*b^2*c^2*d + (3*B*b^2*c^2*d + 3*(2*B*a
*b + A*b^2)*c*d^2 + (B*a^2 + 2*A*a*b)*d^3)*m^5 + 5*(3*B*b^2*c^2*d + 3*(2*B*
a*b + A*b^2)*c*d^2 + (B*a^2 + 2*A*a*b)*d^3)*m^4 + 3*(2*B*a*b + A*b^2)*c*d^2
 + (B*a^2 + 2*A*a*b)*d^3 + 10*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2 +
(B*a^2 + 2*A*a*b)*d^3)*m^3 + 10*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2
+ (B*a^2 + 2*A*a*b)*d^3)*m^2 + 5*(3*B*b^2*c^2*d + 3*(2*B*a*b + A*b^2)*c*d^2
 + (B*a^2 + 2*A*a*b)*d^3)*m)*n)*x*x^(4*n)*e^(m*log(e) + m*log(x)) + ((B*b^2
*c^3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m
^6 + B*b^2*c^3 + A*a^2*d^3 + 6*(B*b^2*c^3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)
*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m^5 + 240*(B*b^2*c^3 + A*a^2*d^3 + 3*(2
*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2 + (B*b^2*c^3 + A*a^2*d^3
+ 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m)*n^5 + 15*(B*b^2
*c^3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m
^4 + 508*(B*b^2*c^3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*
A*a*b)*c*d^2 + (B*b^2*c^3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^
2 + 2*A*a*b)*c*d^2)*m^2 + 2*(B*b^2*c^3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^
2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m)*n^4 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*
a^2 + 2*A*a*b)*c*d^2 + 20*(B*b^2*c^3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*
d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m^3 + 372*(B*b^2*c^3 + A*a^2*d^3 + 3*(2*B*a*
b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2 + (B*b^2*c^3 + A*a^2*d^3 + 3*(
2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m^3 + 3*(B*b^2*c^3 + A*
a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m^2 + 3*(B
*b^2*c^3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^
2)*m)*n^3 + 15*(B*b^2*c^3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^
2 + 2*A*a*b)*c*d^2)*m^2 + 121*(B*b^2*c^3 + A*a^2*d^3 + (B*b^2*c^3 + A*a^2*d
^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m^4 + 3*(2*B*a*
b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2 + 4*(B*b^2*c^3 + A*a^2*d^3 + 3
*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m^3 + 6*(B*b^2*c^3 +
A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m^2 + 4*
(B*b^2*c^3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*
d^2)*m)*n^2 + 6*(B*b^2*c^3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a
^2 + 2*A*a*b)*c*d^2)*m + 18*(B*b^2*c^3 + A*a^2*d^3 + (B*b^2*c^3 + A*a^2*d^3
 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m^5 + 5*(B*b^2*c^
3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m^4
+ 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2 + 10*(B*b^2*c^3 + A
*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*d^2)*m^3 + 10*
(B*b^2*c^3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2 + 2*A*a*b)*c*
d^2)*m^2 + 5*(B*b^2*c^3 + A*a^2*d^3 + 3*(2*B*a*b + A*b^2)*c^2*d + 3*(B*a^2
+ 2*A*a*b)*c*d^2)*m)*n)*x*x^(3*n)*e^(m*log(e) + m*log(x)) + ((3*A*a^2*c*d^2
 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m^6 + 3*A*a^2*c*d^2 +
 6*(3*A*a^2*c*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m^5
```

```
+ 360*(3*A*a^2*c*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d +
(3*A*a^2*c*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m)*n^5
+ 15*(3*A*a^2*c*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m^
4 + 702*(3*A*a^2*c*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d
+ (3*A*a^2*c*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m^2 +
 2*(3*A*a^2*c*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m)*n
^4 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d + 20*(3*A*a^2*c*d^2
+ (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m^3 + 461*(3*A*a^2*c*d
^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d + (3*A*a^2*c*d^2 + (
2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m^3 + 3*(3*A*a^2*c*d^2 +
(2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m^2 + 3*(3*A*a^2*c*d^2 +
 (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m)*n^3 + 15*(3*A*a^2*c*
d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m^2 + 137*(3*A*a^2
*c*d^2 + (3*A*a^2*c*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d
)*m^4 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d + 4*(3*A*a^2*c*d^
2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m^3 + 6*(3*A*a^2*c*d
^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m^2 + 4*(3*A*a^2*c*
d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m)*n^2 + 6*(3*A*a^
2*c*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m + 19*(3*A*a^
2*c*d^2 + (3*A*a^2*c*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*
d)*m^5 + 5*(3*A*a^2*c*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2
*d)*m^4 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d + 10*(3*A*a^2*c
*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m^3 + 10*(3*A*a^2
*c*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m^2 + 5*(3*A*a^
2*c*d^2 + (2*B*a*b + A*b^2)*c^3 + 3*(B*a^2 + 2*A*a*b)*c^2*d)*m)*n)*x*x^(2*n
)*e^(m*log(e) + m*log(x)) + ((3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m^6 +
3*A*a^2*c^2*d + 6*(3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m^5 + 720*(3*A*a^
2*c^2*d + (B*a^2 + 2*A*a*b)*c^3 + (3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m
)*n^5 + 15*(3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m^4 + 1044*(3*A*a^2*c^2*
d + (B*a^2 + 2*A*a*b)*c^3 + (3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m^2 + 2
*(3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m)*n^4 + (B*a^2 + 2*A*a*b)*c^3 + 2
0*(3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m^3 + 580*(3*A*a^2*c^2*d + (B*a^2
 + 2*A*a*b)*c^3 + (3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m^3 + 3*(3*A*a^2*
c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m^2 + 3*(3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c
^3)*m)*n^3 + 15*(3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m^2 + 155*(3*A*a^2*
c^2*d + (3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m^4 + (B*a^2 + 2*A*a*b)*c^3
 + 4*(3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m^3 + 6*(3*A*a^2*c^2*d + (B*a^
2 + 2*A*a*b)*c^3)*m^2 + 4*(3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m)*n^2 +
6*(3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m + 20*(3*A*a^2*c^2*d + (3*A*a^2*
c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m^5 + 5*(3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c
^3)*m^4 + (B*a^2 + 2*A*a*b)*c^3 + 10*(3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3
)*m^3 + 10*(3*A*a^2*c^2*d + (B*a^2 + 2*A*a*b)*c^3)*m^2 + 5*(3*A*a^2*c^2*d +
 (B*a^2 + 2*A*a*b)*c^3)*m)*n)*x*x^n*e^(m*log(e) + m*log(x)) + (A*a^2*c^3*m^
6 + 720*A*a^2*c^3*n^6 + 6*A*a^2*c^3*m^5 + 15*A*a^2*c^3*m^4 + 20*A*a^2*c^3*m
^3 + 15*A*a^2*c^3*m^2 + 6*A*a^2*c^3*m + A*a^2*c^3 + 1764*(A*a^2*c^3*m + A*a
^2*c^3)*n^5 + 1624*(A*a^2*c^3*m^2 + 2*A*a^2*c^3*m + A*a^2*c^3)*n^4 + 735*(A
*a^2*c^3*m^3 + 3*A*a^2*c^3*m^2 + 3*A*a^2*c^3*m + A*a^2*c^3)*n^3 + 175*(A*a^
2*c^3*m^4 + 4*A*a^2*c^3*m^3 + 6*A*a^2*c^3*m^2 + 4*A*a^2*c^3*m + A*a^2*c^3)*
n^2 + 21*(A*a^2*c^3*m^5 + 5*A*a^2*c^3*m^4 + 10*A*a^2*c^3*m^3 + 10*A*a^2*c^3
*m^2 + 5*A*a^2*c^3*m + A*a^2*c^3)*n)*x*e^(m*log(e) + m*log(x)))/(m^7 + 720*
(m + 1)*n^6 + 7*m^6 + 1764*(m^2 + 2*m + 1)*n^5 + 21*m^5 + 1624*(m^3 + 3*m^2
 + 3*m + 1)*n^4 + 35*m^4 + 735*(m^4 + 4*m^3 + 6*m^2 + 4*m + 1)*n^3 + 35*m^3
 + 175*(m^5 + 5*m^4 + 10*m^3 + 10*m^2 + 5*m + 1)*n^2 + 21*m^2 + 21*(m^6 + 6
*m^5 + 15*m^4 + 20*m^3 + 15*m^2 + 6*m + 1)*n + 7*m + 1)
```

giac [B] time = 1.30, size = 15358, normalized size = 49.54

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

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

```
[Out] (B*b^2*d^3*m^6*x*x^m*x^(6*n)*e^m + 15*B*b^2*d^3*m^5*n*x*x^m*x^(6*n)*e^m + 8
5*B*b^2*d^3*m^4*n^2*x*x^m*x^(6*n)*e^m + 225*B*b^2*d^3*m^3*n^3*x*x^m*x^(6*n)
*e^m + 274*B*b^2*d^3*m^2*n^4*x*x^m*x^(6*n)*e^m + 120*B*b^2*d^3*m*n^5*x*x^m*
x^(6*n)*e^m + 3*B*b^2*c*d^2*m^6*x*x^m*x^(5*n)*e^m + 2*B*a*b*d^3*m^6*x*x^m*x
^(5*n)*e^m + A*b^2*d^3*m^6*x*x^m*x^(5*n)*e^m + 48*B*b^2*c*d^2*m^5*n*x*x^m*x
^(5*n)*e^m + 32*B*a*b*d^3*m^5*n*x*x^m*x^(5*n)*e^m + 16*A*b^2*d^3*m^5*n*x*x^
m*x^(5*n)*e^m + 285*B*b^2*c*d^2*m^4*n^2*x*x^m*x^(5*n)*e^m + 190*B*a*b*d^3*m
^4*n^2*x*x^m*x^(5*n)*e^m + 95*A*b^2*d^3*m^4*n^2*x*x^m*x^(5*n)*e^m + 780*B*b
^2*c*d^2*m^3*n^3*x*x^m*x^(5*n)*e^m + 520*B*a*b*d^3*m^3*n^3*x*x^m*x^(5*n)*e^
m + 260*A*b^2*d^3*m^3*n^3*x*x^m*x^(5*n)*e^m + 972*B*b^2*c*d^2*m^2*n^4*x*x^m
*x^(5*n)*e^m + 648*B*a*b*d^3*m^2*n^4*x*x^m*x^(5*n)*e^m + 324*A*b^2*d^3*m^2*
n^4*x*x^m*x^(5*n)*e^m + 432*B*b^2*c*d^2*m*n^5*x*x^m*x^(5*n)*e^m + 288*B*a*b
*d^3*m*n^5*x*x^m*x^(5*n)*e^m + 144*A*b^2*d^3*m*n^5*x*x^m*x^(5*n)*e^m + 3*B*
b^2*c^2*d*m^6*x*x^m*x^(4*n)*e^m + 6*B*a*b*c*d^2*m^6*x*x^m*x^(4*n)*e^m + 3*A
*b^2*c*d^2*m^6*x*x^m*x^(4*n)*e^m + B*a^2*d^3*m^6*x*x^m*x^(4*n)*e^m + 2*A*a*
b*d^3*m^6*x*x^m*x^(4*n)*e^m + 51*B*b^2*c^2*d*m^5*n*x*x^m*x^(4*n)*e^m + 102*
B*a*b*c*d^2*m^5*n*x*x^m*x^(4*n)*e^m + 51*A*b^2*c*d^2*m^5*n*x*x^m*x^(4*n)*e^
m + 17*B*a^2*d^3*m^5*n*x*x^m*x^(4*n)*e^m + 34*A*a*b*d^3*m^5*n*x*x^m*x^(4*n)
*e^m + 321*B*b^2*c^2*d*m^4*n^2*x*x^m*x^(4*n)*e^m + 642*B*a*b*c*d^2*m^4*n^2*
x*x^m*x^(4*n)*e^m + 321*A*b^2*c*d^2*m^4*n^2*x*x^m*x^(4*n)*e^m + 107*B*a^2*d
^3*m^4*n^2*x*x^m*x^(4*n)*e^m + 214*A*a*b*d^3*m^4*n^2*x*x^m*x^(4*n)*e^m + 92
1*B*b^2*c^2*d*m^3*n^3*x*x^m*x^(4*n)*e^m + 1842*B*a*b*c*d^2*m^3*n^3*x*x^m*x^
(4*n)*e^m + 921*A*b^2*c*d^2*m^3*n^3*x*x^m*x^(4*n)*e^m + 307*B*a^2*d^3*m^3*n
^3*x*x^m*x^(4*n)*e^m + 614*A*a*b*d^3*m^3*n^3*x*x^m*x^(4*n)*e^m + 1188*B*b^2
*c^2*d*m^2*n^4*x*x^m*x^(4*n)*e^m + 2376*B*a*b*c*d^2*m^2*n^4*x*x^m*x^(4*n)*e
^m + 1188*A*b^2*c*d^2*m^2*n^4*x*x^m*x^(4*n)*e^m + 396*B*a^2*d^3*m^2*n^4*x*x
^m*x^(4*n)*e^m + 792*A*a*b*d^3*m^2*n^4*x*x^m*x^(4*n)*e^m + 540*B*b^2*c^2*d*
m*n^5*x*x^m*x^(4*n)*e^m + 1080*B*a*b*c*d^2*m*n^5*x*x^m*x^(4*n)*e^m + 540*A*
b^2*c*d^2*m*n^5*x*x^m*x^(4*n)*e^m + 180*B*a^2*d^3*m*n^5*x*x^m*x^(4*n)*e^m +
 360*A*a*b*d^3*m*n^5*x*x^m*x^(4*n)*e^m + B*b^2*c^3*m^6*x*x^m*x^(3*n)*e^m +
6*B*a*b*c^2*d*m^6*x*x^m*x^(3*n)*e^m + 3*A*b^2*c^2*d*m^6*x*x^m*x^(3*n)*e^m +
 3*B*a^2*c*d^2*m^6*x*x^m*x^(3*n)*e^m + 6*A*a*b*c*d^2*m^6*x*x^m*x^(3*n)*e^m
+ A*a^2*d^3*m^6*x*x^m*x^(3*n)*e^m + 18*B*b^2*c^3*m^5*n*x*x^m*x^(3*n)*e^m +
108*B*a*b*c^2*d*m^5*n*x*x^m*x^(3*n)*e^m + 54*A*b^2*c^2*d*m^5*n*x*x^m*x^(3*n
)*e^m + 54*B*a^2*c*d^2*m^5*n*x*x^m*x^(3*n)*e^m + 108*A*a*b*c*d^2*m^5*n*x*x^
m*x^(3*n)*e^m + 18*A*a^2*d^3*m^5*n*x*x^m*x^(3*n)*e^m + 121*B*b^2*c^3*m^4*n^
2*x*x^m*x^(3*n)*e^m + 726*B*a*b*c^2*d*m^4*n^2*x*x^m*x^(3*n)*e^m + 363*A*b^2
*c^2*d*m^4*n^2*x*x^m*x^(3*n)*e^m + 363*B*a^2*c*d^2*m^4*n^2*x*x^m*x^(3*n)*e^
m + 726*A*a*b*c*d^2*m^4*n^2*x*x^m*x^(3*n)*e^m + 121*A*a^2*d^3*m^4*n^2*x*x^m
*x^(3*n)*e^m + 372*B*b^2*c^3*m^3*n^3*x*x^m*x^(3*n)*e^m + 2232*B*a*b*c^2*d*m
^3*n^3*x*x^m*x^(3*n)*e^m + 1116*A*b^2*c^2*d*m^3*n^3*x*x^m*x^(3*n)*e^m + 111
6*B*a^2*c*d^2*m^3*n^3*x*x^m*x^(3*n)*e^m + 2232*A*a*b*c*d^2*m^3*n^3*x*x^m*x^
(3*n)*e^m + 372*A*a^2*d^3*m^3*n^3*x*x^m*x^(3*n)*e^m + 508*B*b^2*c^3*m^2*n^4
*x*x^m*x^(3*n)*e^m + 3048*B*a*b*c^2*d*m^2*n^4*x*x^m*x^(3*n)*e^m + 1524*A*b^
2*c^2*d*m^2*n^4*x*x^m*x^(3*n)*e^m + 1524*B*a^2*c*d^2*m^2*n^4*x*x^m*x^(3*n)*
e^m + 3048*A*a*b*c*d^2*m^2*n^4*x*x^m*x^(3*n)*e^m + 508*A*a^2*d^3*m^2*n^4*x*
x^m*x^(3*n)*e^m + 240*B*b^2*c^3*m*n^5*x*x^m*x^(3*n)*e^m + 1440*B*a*b*c^2*d*
m*n^5*x*x^m*x^(3*n)*e^m + 720*A*b^2*c^2*d*m*n^5*x*x^m*x^(3*n)*e^m + 720*B*a
^2*c*d^2*m*n^5*x*x^m*x^(3*n)*e^m + 1440*A*a*b*c*d^2*m*n^5*x*x^m*x^(3*n)*e^m
 + 240*A*a^2*d^3*m*n^5*x*x^m*x^(3*n)*e^m + 2*B*a*b*c^3*m^6*x*x^m*x^(2*n)*e^
m + A*b^2*c^3*m^6*x*x^m*x^(2*n)*e^m + 3*B*a^2*c^2*d*m^6*x*x^m*x^(2*n)*e^m +
 6*A*a*b*c^2*d*m^6*x*x^m*x^(2*n)*e^m + 3*A*a^2*c*d^2*m^6*x*x^m*x^(2*n)*e^m
+ 38*B*a*b*c^3*m^5*n*x*x^m*x^(2*n)*e^m + 19*A*b^2*c^3*m^5*n*x*x^m*x^(2*n)*e
^m + 57*B*a^2*c^2*d*m^5*n*x*x^m*x^(2*n)*e^m + 114*A*a*b*c^2*d*m^5*n*x*x^m*x
^(2*n)*e^m + 57*A*a^2*c*d^2*m^5*n*x*x^m*x^(2*n)*e^m + 274*B*a*b*c^3*m^4*n^2
*x*x^m*x^(2*n)*e^m + 137*A*b^2*c^3*m^4*n^2*x*x^m*x^(2*n)*e^m + 411*B*a^2*c^
2*d*m^4*n^2*x*x^m*x^(2*n)*e^m + 822*A*a*b*c^2*d*m^4*n^2*x*x^m*x^(2*n)*e^m +
 411*A*a^2*c*d^2*m^4*n^2*x*x^m*x^(2*n)*e^m + 922*B*a*b*c^3*m^3*n^3*x*x^m*x^
```

(2*n)*e^m + 461*A*b^2*c^3*m^3*n^3*x*x^m*x^(2*n)*e^m + 1383*B*a^2*c^2*d*m^3*
n^3*x*x^m*x^(2*n)*e^m + 2766*A*a*b*c^2*d*m^3*n^3*x*x^m*x^(2*n)*e^m + 1383*A
*a^2*c*d^2*m^3*n^3*x*x^m*x^(2*n)*e^m + 1404*B*a*b*c^3*m^2*n^4*x*x^m*x^(2*n)
*e^m + 702*A*b^2*c^3*m^2*n^4*x*x^m*x^(2*n)*e^m + 2106*B*a^2*c^2*d*m^2*n^4*x
*x^m*x^(2*n)*e^m + 4212*A*a*b*c^2*d*m^2*n^4*x*x^m*x^(2*n)*e^m + 2106*A*a^2*
c*d^2*m^2*n^4*x*x^m*x^(2*n)*e^m + 720*B*a*b*c^3*m*n^5*x*x^m*x^(2*n)*e^m + 3
60*A*b^2*c^3*m*n^5*x*x^m*x^(2*n)*e^m + 1080*B*a^2*c^2*d*m*n^5*x*x^m*x^(2*n)
*e^m + 2160*A*a*b*c^2*d*m*n^5*x*x^m*x^(2*n)*e^m + 1080*A*a^2*c*d^2*m*n^5*x*
x^m*x^(2*n)*e^m + B*a^2*c^3*m^6*x*x^m*x^n*e^m + 2*A*a*b*c^3*m^6*x*x^m*x^n*e
^m + 3*A*a^2*c^2*d*m^6*x*x^m*x^n*e^m + 20*B*a^2*c^3*m^5*n*x*x^m*x^n*e^m + 4
0*A*a*b*c^3*m^5*n*x*x^m*x^n*e^m + 60*A*a^2*c^2*d*m^5*n*x*x^m*x^n*e^m + 155*
B*a^2*c^3*m^4*n^2*x*x^m*x^n*e^m + 310*A*a*b*c^3*m^4*n^2*x*x^m*x^n*e^m + 465
*A*a^2*c^2*d*m^4*n^2*x*x^m*x^n*e^m + 580*B*a^2*c^3*m^3*n^3*x*x^m*x^n*e^m +
1160*A*a*b*c^3*m^3*n^3*x*x^m*x^n*e^m + 1740*A*a^2*c^2*d*m^3*n^3*x*x^m*x^n*e
^m + 1044*B*a^2*c^3*m^2*n^4*x*x^m*x^n*e^m + 2088*A*a*b*c^3*m^2*n^4*x*x^m*x^
n*e^m + 3132*A*a^2*c^2*d*m^2*n^4*x*x^m*x^n*e^m + 720*B*a^2*c^3*m*n^5*x*x^m*
x^n*e^m + 1440*A*a*b*c^3*m*n^5*x*x^m*x^n*e^m + 2160*A*a^2*c^2*d*m*n^5*x*x^m
*x^n*e^m + A*a^2*c^3*m^6*x*x^m*e^m + 21*A*a^2*c^3*m^5*n*x*x^m*e^m + 175*A*a
^2*c^3*m^4*n^2*x*x^m*e^m + 735*A*a^2*c^3*m^3*n^3*x*x^m*e^m + 1624*A*a^2*c^3
*m^2*n^4*x*x^m*e^m + 1764*A*a^2*c^3*m*n^5*x*x^m*e^m + 720*A*a^2*c^3*n^6*x*x
^m*e^m + 6*B*b^2*d^3*m^5*x*x^m*x^(6*n)*e^m + 75*B*b^2*d^3*m^4*n*x*x^m*x^(6*
n)*e^m + 340*B*b^2*d^3*m^3*n^2*x*x^m*x^(6*n)*e^m + 675*B*b^2*d^3*m^2*n^3*x*
x^m*x^(6*n)*e^m + 548*B*b^2*d^3*m*n^4*x*x^m*x^(6*n)*e^m + 120*B*b^2*d^3*n^5
*x*x^m*x^(6*n)*e^m + 18*B*b^2*c*d^2*m^5*x*x^m*x^(5*n)*e^m + 12*B*a*b*d^3*m^
5*x*x^m*x^(5*n)*e^m + 6*A*b^2*d^3*m^5*x*x^m*x^(5*n)*e^m + 240*B*b^2*c*d^2*m
^4*n*x*x^m*x^(5*n)*e^m + 160*B*a*b*d^3*m^4*n*x*x^m*x^(5*n)*e^m + 80*A*b^2*d
^3*m^4*n*x*x^m*x^(5*n)*e^m + 1140*B*b^2*c*d^2*m^3*n^2*x*x^m*x^(5*n)*e^m + 7
60*B*a*b*d^3*m^3*n^2*x*x^m*x^(5*n)*e^m + 380*A*b^2*d^3*m^3*n^2*x*x^m*x^(5*n
)*e^m + 2340*B*b^2*c*d^2*m^2*n^3*x*x^m*x^(5*n)*e^m + 1560*B*a*b*d^3*m^2*n^3
*x*x^m*x^(5*n)*e^m + 780*A*b^2*d^3*m^2*n^3*x*x^m*x^(5*n)*e^m + 1944*B*b^2*c
*d^2*m*n^4*x*x^m*x^(5*n)*e^m + 1296*B*a*b*d^3*m*n^4*x*x^m*x^(5*n)*e^m + 648
*A*b^2*d^3*m*n^4*x*x^m*x^(5*n)*e^m + 432*B*b^2*c*d^2*n^5*x*x^m*x^(5*n)*e^m
+ 288*B*a*b*d^3*n^5*x*x^m*x^(5*n)*e^m + 144*A*b^2*d^3*n^5*x*x^m*x^(5*n)*e^m
+ 18*B*b^2*c^2*d*m^5*x*x^m*x^(4*n)*e^m + 36*B*a*b*c*d^2*m^5*x*x^m*x^(4*n)*
e^m + 18*A*b^2*c*d^2*m^5*x*x^m*x^(4*n)*e^m + 6*B*a^2*d^3*m^5*x*x^m*x^(4*n)*
e^m + 12*A*a*b*d^3*m^5*x*x^m*x^(4*n)*e^m + 255*B*b^2*c^2*d*m^4*n*x*x^m*x^(4
*n)*e^m + 510*B*a*b*c*d^2*m^4*n*x*x^m*x^(4*n)*e^m + 255*A*b^2*c*d^2*m^4*n*x
*x^m*x^(4*n)*e^m + 85*B*a^2*d^3*m^4*n*x*x^m*x^(4*n)*e^m + 170*A*a*b*d^3*m^4
*n*x*x^m*x^(4*n)*e^m + 1284*B*b^2*c^2*d*m^3*n^2*x*x^m*x^(4*n)*e^m + 2568*B*
a*b*c*d^2*m^3*n^2*x*x^m*x^(4*n)*e^m + 1284*A*b^2*c*d^2*m^3*n^2*x*x^m*x^(4*n
)*e^m + 428*B*a^2*d^3*m^3*n^2*x*x^m*x^(4*n)*e^m + 856*A*a*b*d^3*m^3*n^2*x*x
^m*x^(4*n)*e^m + 2763*B*b^2*c^2*d*m^2*n^3*x*x^m*x^(4*n)*e^m + 5526*B*a*b*c*
d^2*m^2*n^3*x*x^m*x^(4*n)*e^m + 2763*A*b^2*c*d^2*m^2*n^3*x*x^m*x^(4*n)*e^m
+ 921*B*a^2*d^3*m^2*n^3*x*x^m*x^(4*n)*e^m + 1842*A*a*b*d^3*m^2*n^3*x*x^m*x^
(4*n)*e^m + 2376*B*b^2*c^2*d*m*n^4*x*x^m*x^(4*n)*e^m + 4752*B*a*b*c*d^2*m*n
^4*x*x^m*x^(4*n)*e^m + 2376*A*b^2*c*d^2*m*n^4*x*x^m*x^(4*n)*e^m + 792*B*a^2
*d^3*m*n^4*x*x^m*x^(4*n)*e^m + 1584*A*a*b*d^3*m*n^4*x*x^m*x^(4*n)*e^m + 540
*B*b^2*c^2*d*n^5*x*x^m*x^(4*n)*e^m + 1080*B*a*b*c*d^2*n^5*x*x^m*x^(4*n)*e^m
+ 540*A*b^2*c*d^2*n^5*x*x^m*x^(4*n)*e^m + 180*B*a^2*d^3*n^5*x*x^m*x^(4*n)*
e^m + 360*A*a*b*d^3*n^5*x*x^m*x^(4*n)*e^m + 6*B*b^2*c^3*m^5*x*x^m*x^(3*n)*e
^m + 36*B*a*b*c^2*d*m^5*x*x^m*x^(3*n)*e^m + 18*A*b^2*c^2*d*m^5*x*x^m*x^(3*n
)*e^m + 18*B*a^2*c*d^2*m^5*x*x^m*x^(3*n)*e^m + 36*A*a*b*c*d^2*m^5*x*x^m*x^(
3*n)*e^m + 6*A*a^2*d^3*m^5*x*x^m*x^(3*n)*e^m + 90*B*b^2*c^3*m^4*n*x*x^m*x^(
3*n)*e^m + 540*B*a*b*c^2*d*m^4*n*x*x^m*x^(3*n)*e^m + 270*A*b^2*c^2*d*m^4*n*
x*x^m*x^(3*n)*e^m + 270*B*a^2*c*d^2*m^4*n*x*x^m*x^(3*n)*e^m + 540*A*a*b*c*d
^2*m^4*n*x*x^m*x^(3*n)*e^m + 90*A*a^2*d^3*m^4*n*x*x^m*x^(3*n)*e^m + 484*B*b
^2*c^3*m^3*n^2*x*x^m*x^(3*n)*e^m + 2904*B*a*b*c^2*d*m^3*n^2*x*x^m*x^(3*n)*e
^m + 1452*A*b^2*c^2*d*m^3*n^2*x*x^m*x^(3*n)*e^m + 1452*B*a^2*c*d^2*m^3*n^2*
x*x^m*x^(3*n)*e^m + 2904*A*a*b*c*d^2*m^3*n^2*x*x^m*x^(3*n)*e^m + 484*A*a^2*

```
d^3*m^3*n^2*x*x^m*x^(3*n)*e^m + 1116*B*b^2*c^3*m^2*n^3*x*x^m*x^(3*n)*e^m +
6696*B*a*b*c^2*d*m^2*n^3*x*x^m*x^(3*n)*e^m + 3348*A*b^2*c^2*d*m^2*n^3*x*x^m
*x^(3*n)*e^m + 3348*B*a^2*c*d^2*m^2*n^3*x*x^m*x^(3*n)*e^m + 6696*A*a*b*c*d^
2*m^2*n^3*x*x^m*x^(3*n)*e^m + 1116*A*a^2*d^3*m^2*n^3*x*x^m*x^(3*n)*e^m + 10
16*B*b^2*c^3*m*n^4*x*x^m*x^(3*n)*e^m + 6096*B*a*b*c^2*d*m*n^4*x*x^m*x^(3*n)
*e^m + 3048*A*b^2*c^2*d*m*n^4*x*x^m*x^(3*n)*e^m + 3048*B*a^2*c*d^2*m*n^4*x*
x^m*x^(3*n)*e^m + 6096*A*a*b*c*d^2*m*n^4*x*x^m*x^(3*n)*e^m + 1016*A*a^2*d^3
*m*n^4*x*x^m*x^(3*n)*e^m + 240*B*b^2*c^3*n^5*x*x^m*x^(3*n)*e^m + 1440*B*a*b
*c^2*d*n^5*x*x^m*x^(3*n)*e^m + 720*A*b^2*c^2*d*n^5*x*x^m*x^(3*n)*e^m + 720*
B*a^2*c*d^2*n^5*x*x^m*x^(3*n)*e^m + 1440*A*a*b*c*d^2*n^5*x*x^m*x^(3*n)*e^m
 + 240*A*a^2*d^3*n^5*x*x^m*x^(3*n)*e^m + 12*B*a*b*c^3*m^5*x*x^m*x^(2*n)*e^m
 + 6*A*b^2*c^3*m^5*x*x^m*x^(2*n)*e^m + 18*B*a^2*c^2*d*m^5*x*x^m*x^(2*n)*e^m
 + 36*A*a*b*c^2*d*m^5*x*x^m*x^(2*n)*e^m + 18*A*a^2*c*d^2*m^5*x*x^m*x^(2*n)*e
^m + 190*B*a*b*c^3*m^4*n*x*x^m*x^(2*n)*e^m + 95*A*b^2*c^3*m^4*n*x*x^m*x^(2*
n)*e^m + 285*B*a^2*c^2*d*m^4*n*x*x^m*x^(2*n)*e^m + 570*A*a*b*c^2*d*m^4*n*x*
x^m*x^(2*n)*e^m + 285*A*a^2*c*d^2*m^4*n*x*x^m*x^(2*n)*e^m + 1096*B*a*b*c^3*
m^3*n^2*x*x^m*x^(2*n)*e^m + 548*A*b^2*c^3*m^3*n^2*x*x^m*x^(2*n)*e^m + 1644*
B*a^2*c^2*d*m^3*n^2*x*x^m*x^(2*n)*e^m + 3288*A*a*b*c^2*d*m^3*n^2*x*x^m*x^(2
*n)*e^m + 1644*A*a^2*c*d^2*m^3*n^2*x*x^m*x^(2*n)*e^m + 2766*B*a*b*c^3*m^2*n
^3*x*x^m*x^(2*n)*e^m + 1383*A*b^2*c^3*m^2*n^3*x*x^m*x^(2*n)*e^m + 4149*B*a^
2*c^2*d*m^2*n^3*x*x^m*x^(2*n)*e^m + 8298*A*a*b*c^2*d*m^2*n^3*x*x^m*x^(2*n)*
e^m + 4149*A*a^2*c*d^2*m^2*n^3*x*x^m*x^(2*n)*e^m + 2808*B*a*b*c^3*m*n^4*x*x
^m*x^(2*n)*e^m + 1404*A*b^2*c^3*m*n^4*x*x^m*x^(2*n)*e^m + 4212*B*a^2*c^2*d*
m*n^4*x*x^m*x^(2*n)*e^m + 8424*A*a*b*c^2*d*m*n^4*x*x^m*x^(2*n)*e^m + 4212*A
*a^2*c*d^2*m*n^4*x*x^m*x^(2*n)*e^m + 720*B*a*b*c^3*n^5*x*x^m*x^(2*n)*e^m +
360*A*b^2*c^3*n^5*x*x^m*x^(2*n)*e^m + 1080*B*a^2*c^2*d*n^5*x*x^m*x^(2*n)*e^
m + 2160*A*a*b*c^2*d*n^5*x*x^m*x^(2*n)*e^m + 1080*A*a^2*c*d^2*n^5*x*x^m*x^(
2*n)*e^m + 6*B*a^2*c^3*m^5*x*x^m*x^n*e^m + 12*A*a*b*c^3*m^5*x*x^m*x^n*e^m +
 18*A*a^2*c^2*d*m^5*x*x^m*x^n*e^m + 100*B*a^2*c^3*m^4*n*x*x^m*x^n*e^m + 200
*A*a*b*c^3*m^4*n*x*x^m*x^n*e^m + 300*A*a^2*c^2*d*m^4*n*x*x^m*x^n*e^m + 620*
B*a^2*c^3*m^3*n^2*x*x^m*x^n*e^m + 1240*A*a*b*c^3*m^3*n^2*x*x^m*x^n*e^m + 18
60*A*a^2*c^2*d*m^3*n^2*x*x^m*x^n*e^m + 1740*B*a^2*c^3*m^2*n^3*x*x^m*x^n*e^m
 + 3480*A*a*b*c^3*m^2*n^3*x*x^m*x^n*e^m + 5220*A*a^2*c^2*d*m^2*n^3*x*x^m*x^
n*e^m + 2088*B*a^2*c^3*m*n^4*x*x^m*x^n*e^m + 4176*A*a*b*c^3*m*n^4*x*x^m*x^n
*e^m + 6264*A*a^2*c^2*d*m*n^4*x*x^m*x^n*e^m + 720*B*a^2*c^3*n^5*x*x^m*x^n*e
^m + 1440*A*a*b*c^3*n^5*x*x^m*x^n*e^m + 2160*A*a^2*c^2*d*n^5*x*x^m*x^n*e^m
+ 6*A*a^2*c^3*m^5*x*x^m*e^m + 105*A*a^2*c^3*m^4*n*x*x^m*e^m + 700*A*a^2*c^3
*m^3*n^2*x*x^m*e^m + 2205*A*a^2*c^3*m^2*n^3*x*x^m*e^m + 3248*A*a^2*c^3*m*n^
4*x*x^m*e^m + 1764*A*a^2*c^3*n^5*x*x^m*e^m + 15*B*b^2*d^3*m^4*x*x^m*x^(6*n)
*e^m + 150*B*b^2*d^3*m^3*n*x*x^m*x^(6*n)*e^m + 510*B*b^2*d^3*m^2*n^2*x*x^m*
x^(6*n)*e^m + 675*B*b^2*d^3*m*n^3*x*x^m*x^(6*n)*e^m + 274*B*b^2*d^3*n^4*x*x
^m*x^(6*n)*e^m + 45*B*b^2*c*d^2*m^4*x*x^m*x^(5*n)*e^m + 30*B*a*b*d^3*m^4*x*
x^m*x^(5*n)*e^m + 15*A*b^2*d^3*m^4*x*x^m*x^(5*n)*e^m + 480*B*b^2*c*d^2*m^3*
n*x*x^m*x^(5*n)*e^m + 320*B*a*b*d^3*m^3*n*x*x^m*x^(5*n)*e^m + 160*A*b^2*d^3
*m^3*n*x*x^m*x^(5*n)*e^m + 1710*B*b^2*c*d^2*m^2*n^2*x*x^m*x^(5*n)*e^m + 114
0*B*a*b*d^3*m^2*n^2*x*x^m*x^(5*n)*e^m + 570*A*b^2*d^3*m^2*n^2*x*x^m*x^(5*n)
*e^m + 2340*B*b^2*c*d^2*m*n^3*x*x^m*x^(5*n)*e^m + 1560*B*a*b*d^3*m*n^3*x*x^
m*x^(5*n)*e^m + 780*A*b^2*d^3*m*n^3*x*x^m*x^(5*n)*e^m + 972*B*b^2*c*d^2*n^4
*x*x^m*x^(5*n)*e^m + 648*B*a*b*d^3*n^4*x*x^m*x^(5*n)*e^m + 324*A*b^2*d^3*n^
4*x*x^m*x^(5*n)*e^m + 45*B*b^2*c^2*d*m^4*x*x^m*x^(4*n)*e^m + 90*B*a*b*c*d^2
*m^4*x*x^m*x^(4*n)*e^m + 45*A*b^2*c*d^2*m^4*x*x^m*x^(4*n)*e^m + 15*B*a^2*d^
3*m^4*x*x^m*x^(4*n)*e^m + 30*A*a*b*d^3*m^4*x*x^m*x^(4*n)*e^m + 510*B*b^2*c^
2*d*m^3*n*x*x^m*x^(4*n)*e^m + 1020*B*a*b*c*d^2*m^3*n*x*x^m*x^(4*n)*e^m + 51
0*A*b^2*c*d^2*m^3*n*x*x^m*x^(4*n)*e^m + 170*B*a^2*d^3*m^3*n*x*x^m*x^(4*n)*e
^m + 340*A*a*b*d^3*m^3*n*x*x^m*x^(4*n)*e^m + 1926*B*b^2*c^2*d*m^2*n^2*x*x^m
*x^(4*n)*e^m + 3852*B*a*b*c*d^2*m^2*n^2*x*x^m*x^(4*n)*e^m + 1926*A*b^2*c*d^
2*m^2*n^2*x*x^m*x^(4*n)*e^m + 642*B*a^2*d^3*m^2*n^2*x*x^m*x^(4*n)*e^m + 128
4*A*a*b*d^3*m^2*n^2*x*x^m*x^(4*n)*e^m + 2763*B*b^2*c^2*d*m*n^3*x*x^m*x^(4*n
)*e^m + 5526*B*a*b*c*d^2*m*n^3*x*x^m*x^(4*n)*e^m + 2763*A*b^2*c*d^2*m*n^3*x
```

*x^m*x^(4*n)*e^m + 921*B*a^2*d^3*m*n^3*x*x^m*x^(4*n)*e^m + 1842*A*a*b*d^3*m
*n^3*x*x^m*x^(4*n)*e^m + 1188*B*b^2*c^2*d*n^4*x*x^m*x^(4*n)*e^m + 2376*B*a*
b*c*d^2*n^4*x*x^m*x^(4*n)*e^m + 1188*A*b^2*c*d^2*n^4*x*x^m*x^(4*n)*e^m + 39
6*B*a^2*d^3*n^4*x*x^m*x^(4*n)*e^m + 792*A*a*b*d^3*n^4*x*x^m*x^(4*n)*e^m + 1
5*B*b^2*c^3*m^4*x*x^m*x^(3*n)*e^m + 90*B*a*b*c^2*d*m^4*x*x^m*x^(3*n)*e^m +
45*A*b^2*c^2*d*m^4*x*x^m*x^(3*n)*e^m + 45*B*a^2*c*d^2*m^4*x*x^m*x^(3*n)*e^m
 + 90*A*a*b*c*d^2*m^4*x*x^m*x^(3*n)*e^m + 15*A*a^2*d^3*m^4*x*x^m*x^(3*n)*e^
m + 180*B*b^2*c^3*m^3*n*x*x^m*x^(3*n)*e^m + 1080*B*a*b*c^2*d*m^3*n*x*x^m*x^
(3*n)*e^m + 540*A*b^2*c^2*d*m^3*n*x*x^m*x^(3*n)*e^m + 540*B*a^2*c*d^2*m^3*n
*x*x^m*x^(3*n)*e^m + 1080*A*a*b*c*d^2*m^3*n*x*x^m*x^(3*n)*e^m + 180*A*a^2*d
^3*m^3*n*x*x^m*x^(3*n)*e^m + 726*B*b^2*c^3*m^2*n^2*x*x^m*x^(3*n)*e^m + 4356
*B*a*b*c^2*d*m^2*n^2*x*x^m*x^(3*n)*e^m + 2178*A*b^2*c^2*d*m^2*n^2*x*x^m*x^(
3*n)*e^m + 2178*B*a^2*c*d^2*m^2*n^2*x*x^m*x^(3*n)*e^m + 4356*A*a*b*c*d^2*m^
2*n^2*x*x^m*x^(3*n)*e^m + 726*A*a^2*d^3*m^2*n^2*x*x^m*x^(3*n)*e^m + 1116*B*
b^2*c^3*m*n^3*x*x^m*x^(3*n)*e^m + 6696*B*a*b*c^2*d*m*n^3*x*x^m*x^(3*n)*e^m
+ 3348*A*b^2*c^2*d*m*n^3*x*x^m*x^(3*n)*e^m + 3348*B*a^2*c*d^2*m*n^3*x*x^m*x
^(3*n)*e^m + 6696*A*a*b*c*d^2*m*n^3*x*x^m*x^(3*n)*e^m + 1116*A*a^2*d^3*m*n^
3*x*x^m*x^(3*n)*e^m + 508*B*b^2*c^3*n^4*x*x^m*x^(3*n)*e^m + 3048*B*a*b*c^2*
d*n^4*x*x^m*x^(3*n)*e^m + 1524*A*b^2*c^2*d*n^4*x*x^m*x^(3*n)*e^m + 1524*B*a
^2*c*d^2*n^4*x*x^m*x^(3*n)*e^m + 3048*A*a*b*c*d^2*n^4*x*x^m*x^(3*n)*e^m + 5
08*A*a^2*d^3*n^4*x*x^m*x^(3*n)*e^m + 30*B*a*b*c^3*m^4*x*x^m*x^(2*n)*e^m + 1
5*A*b^2*c^3*m^4*x*x^m*x^(2*n)*e^m + 45*B*a^2*c^2*d*m^4*x*x^m*x^(2*n)*e^m +
90*A*a*b*c^2*d*m^4*x*x^m*x^(2*n)*e^m + 45*A*a^2*c*d^2*m^4*x*x^m*x^(2*n)*e^m
 + 380*B*a*b*c^3*m^3*n*x*x^m*x^(2*n)*e^m + 190*A*b^2*c^3*m^3*n*x*x^m*x^(2*n
)*e^m + 570*B*a^2*c^2*d*m^3*n*x*x^m*x^(2*n)*e^m + 1140*A*a*b*c^2*d*m^3*n*x*
x^m*x^(2*n)*e^m + 570*A*a^2*c*d^2*m^3*n*x*x^m*x^(2*n)*e^m + 1644*B*a*b*c^3*
m^2*n^2*x*x^m*x^(2*n)*e^m + 822*A*b^2*c^3*m^2*n^2*x*x^m*x^(2*n)*e^m + 2466*
B*a^2*c^2*d*m^2*n^2*x*x^m*x^(2*n)*e^m + 4932*A*a*b*c^2*d*m^2*n^2*x*x^m*x^(2
*n)*e^m + 2466*A*a^2*c*d^2*m^2*n^2*x*x^m*x^(2*n)*e^m + 2766*B*a*b*c^3*m*n^3
*x*x^m*x^(2*n)*e^m + 1383*A*b^2*c^3*m*n^3*x*x^m*x^(2*n)*e^m + 4149*B*a^2*c^
2*d*m*n^3*x*x^m*x^(2*n)*e^m + 8298*A*a*b*c^2*d*m*n^3*x*x^m*x^(2*n)*e^m + 41
49*A*a^2*c*d^2*m*n^3*x*x^m*x^(2*n)*e^m + 1404*B*a*b*c^3*n^4*x*x^m*x^(2*n)*e
^m + 702*A*b^2*c^3*n^4*x*x^m*x^(2*n)*e^m + 2106*B*a^2*c^2*d*n^4*x*x^m*x^(2*
n)*e^m + 4212*A*a*b*c^2*d*n^4*x*x^m*x^(2*n)*e^m + 2106*A*a^2*c*d^2*n^4*x*x^
m*x^(2*n)*e^m + 15*B*a^2*c^3*m^4*x*x^m*x^n*e^m + 30*A*a*b*c^3*m^4*x*x^m*x^n
*e^m + 45*A*a^2*c^2*d*m^4*x*x^m*x^n*e^m + 200*B*a^2*c^3*m^3*n*x*x^m*x^n*e^m
 + 400*A*a*b*c^3*m^3*n*x*x^m*x^n*e^m + 600*A*a^2*c^2*d*m^3*n*x*x^m*x^n*e^m
+ 930*B*a^2*c^3*m^2*n^2*x*x^m*x^n*e^m + 1860*A*a*b*c^3*m^2*n^2*x*x^m*x^n*e^
m + 2790*A*a^2*c^2*d*m^2*n^2*x*x^m*x^n*e^m + 1740*B*a^2*c^3*m*n^3*x*x^m*x^n
*e^m + 3480*A*a*b*c^3*m*n^3*x*x^m*x^n*e^m + 5220*A*a^2*c^2*d*m*n^3*x*x^m*x^
n*e^m + 1044*B*a^2*c^3*n^4*x*x^m*x^n*e^m + 2088*A*a*b*c^3*n^4*x*x^m*x^n*e^m
 + 3132*A*a^2*c^2*d*n^4*x*x^m*x^n*e^m + 15*A*a^2*c^3*m^4*x*x^m*e^m + 210*A*
a^2*c^3*m^3*n*x*x^m*e^m + 1050*A*a^2*c^3*m^2*n^2*x*x^m*e^m + 2205*A*a^2*c^3
*m*n^3*x*x^m*e^m + 1624*A*a^2*c^3*n^4*x*x^m*e^m + 20*B*b^2*d^3*m^3*x*x^m*x^
(6*n)*e^m + 150*B*b^2*d^3*m^2*n*x*x^m*x^(6*n)*e^m + 340*B*b^2*d^3*m*n^2*x*x
^m*x^(6*n)*e^m + 225*B*b^2*d^3*n^3*x*x^m*x^(6*n)*e^m + 60*B*b^2*c*d^2*m^3*x
*x^m*x^(5*n)*e^m + 40*B*a*b*d^3*m^3*x*x^m*x^(5*n)*e^m + 20*A*b^2*d^3*m^3*x*
x^m*x^(5*n)*e^m + 480*B*b^2*c*d^2*m^2*n*x*x^m*x^(5*n)*e^m + 320*B*a*b*d^3*m
^2*n*x*x^m*x^(5*n)*e^m + 160*A*b^2*d^3*m^2*n*x*x^m*x^(5*n)*e^m + 1140*B*b^2
*c*d^2*m*n^2*x*x^m*x^(5*n)*e^m + 760*B*a*b*d^3*m*n^2*x*x^m*x^(5*n)*e^m + 38
0*A*b^2*d^3*m*n^2*x*x^m*x^(5*n)*e^m + 780*B*b^2*c*d^2*n^3*x*x^m*x^(5*n)*e^m
 + 520*B*a*b*d^3*n^3*x*x^m*x^(5*n)*e^m + 260*A*b^2*d^3*n^3*x*x^m*x^(5*n)*e^
m + 60*B*b^2*c^2*d*m^3*x*x^m*x^(4*n)*e^m + 120*B*a*b*c*d^2*m^3*x*x^m*x^(4*n
)*e^m + 60*A*b^2*c*d^2*m^3*x*x^m*x^(4*n)*e^m + 20*B*a^2*d^3*m^3*x*x^m*x^(4*
n)*e^m + 40*A*a*b*d^3*m^3*x*x^m*x^(4*n)*e^m + 510*B*b^2*c^2*d*m^2*n*x*x^m*x
^(4*n)*e^m + 1020*B*a*b*c*d^2*m^2*n*x*x^m*x^(4*n)*e^m + 510*A*b^2*c*d^2*m^2
*n*x*x^m*x^(4*n)*e^m + 170*B*a^2*d^3*m^2*n*x*x^m*x^(4*n)*e^m + 340*A*a*b*d^
3*m^2*n*x*x^m*x^(4*n)*e^m + 1284*B*b^2*c^2*d*m*n^2*x*x^m*x^(4*n)*e^m + 2568
*B*a*b*c*d^2*m*n^2*x*x^m*x^(4*n)*e^m + 1284*A*b^2*c*d^2*m*n^2*x*x^m*x^(4*n)

*e^m + 428*B*a^2*d^3*m*n^2*x*x^m*x^(4*n)*e^m + 856*A*a*b*d^3*m*n^2*x*x^m*x^
(4*n)*e^m + 921*B*b^2*c^2*d*n^3*x*x^m*x^(4*n)*e^m + 1842*B*a*b*c*d^2*n^3*x*
x^m*x^(4*n)*e^m + 921*A*b^2*c*d^2*n^3*x*x^m*x^(4*n)*e^m + 307*B*a^2*d^3*n^3
*x*x^m*x^(4*n)*e^m + 614*A*a*b*d^3*n^3*x*x^m*x^(4*n)*e^m + 20*B*b^2*c^3*m^3
*x*x^m*x^(3*n)*e^m + 120*B*a*b*c^2*d*m^3*x*x^m*x^(3*n)*e^m + 60*A*b^2*c^2*d
*m^3*x*x^m*x^(3*n)*e^m + 60*B*a^2*c*d^2*m^3*x*x^m*x^(3*n)*e^m + 120*A*a*b*c
*d^2*m^3*x*x^m*x^(3*n)*e^m + 20*A*a^2*d^3*m^3*x*x^m*x^(3*n)*e^m + 180*B*b^2
*c^3*m^2*n*x*x^m*x^(3*n)*e^m + 1080*B*a*b*c^2*d*m^2*n*x*x^m*x^(3*n)*e^m + 5
40*A*b^2*c^2*d*m^2*n*x*x^m*x^(3*n)*e^m + 540*B*a^2*c*d^2*m^2*n*x*x^m*x^(3*n
)*e^m + 1080*A*a*b*c*d^2*m^2*n*x*x^m*x^(3*n)*e^m + 180*A*a^2*d^3*m^2*n*x*x^
m*x^(3*n)*e^m + 484*B*b^2*c^3*m*n^2*x*x^m*x^(3*n)*e^m + 2904*B*a*b*c^2*d*m*
n^2*x*x^m*x^(3*n)*e^m + 1452*A*b^2*c^2*d*m*n^2*x*x^m*x^(3*n)*e^m + 1452*B*a
^2*c*d^2*m*n^2*x*x^m*x^(3*n)*e^m + 2904*A*a*b*c*d^2*m*n^2*x*x^m*x^(3*n)*e^m
 + 484*A*a^2*d^3*m*n^2*x*x^m*x^(3*n)*e^m + 372*B*b^2*c^3*n^3*x*x^m*x^(3*n)*
e^m + 2232*B*a*b*c^2*d*n^3*x*x^m*x^(3*n)*e^m + 1116*A*b^2*c^2*d*n^3*x*x^m*x
^(3*n)*e^m + 1116*B*a^2*c*d^2*n^3*x*x^m*x^(3*n)*e^m + 2232*A*a*b*c*d^2*n^3*
x*x^m*x^(3*n)*e^m + 372*A*a^2*d^3*n^3*x*x^m*x^(3*n)*e^m + 40*B*a*b*c^3*m^3*
x*x^m*x^(2*n)*e^m + 20*A*b^2*c^3*m^3*x*x^m*x^(2*n)*e^m + 60*B*a^2*c^2*d*m^3
*x*x^m*x^(2*n)*e^m + 120*A*a*b*c^2*d*m^3*x*x^m*x^(2*n)*e^m + 60*A*a^2*c*d^2
*m^3*x*x^m*x^(2*n)*e^m + 380*B*a*b*c^3*m^2*n*x*x^m*x^(2*n)*e^m + 190*A*b^2*
c^3*m^2*n*x*x^m*x^(2*n)*e^m + 570*B*a^2*c^2*d*m^2*n*x*x^m*x^(2*n)*e^m + 114
0*A*a*b*c^2*d*m^2*n*x*x^m*x^(2*n)*e^m + 570*A*a^2*c*d^2*m^2*n*x*x^m*x^(2*n)
*e^m + 1096*B*a*b*c^3*m*n^2*x*x^m*x^(2*n)*e^m + 548*A*b^2*c^3*m*n^2*x*x^m*x
^(2*n)*e^m + 1644*B*a^2*c^2*d*m*n^2*x*x^m*x^(2*n)*e^m + 3288*A*a*b*c^2*d*m*
n^2*x*x^m*x^(2*n)*e^m + 1644*A*a^2*c*d^2*m*n^2*x*x^m*x^(2*n)*e^m + 922*B*a*
b*c^3*n^3*x*x^m*x^(2*n)*e^m + 461*A*b^2*c^3*n^3*x*x^m*x^(2*n)*e^m + 1383*B*
a^2*c^2*d*n^3*x*x^m*x^(2*n)*e^m + 2766*A*a*b*c^2*d*n^3*x*x^m*x^(2*n)*e^m +
1383*A*a^2*c*d^2*n^3*x*x^m*x^(2*n)*e^m + 20*B*a^2*c^3*m^3*x*x^m*x^n*e^m + 4
0*A*a*b*c^3*m^3*x*x^m*x^n*e^m + 60*A*a^2*c^2*d*m^3*x*x^m*x^n*e^m + 200*B*a^
2*c^3*m^2*n*x*x^m*x^n*e^m + 400*A*a*b*c^3*m^2*n*x*x^m*x^n*e^m + 600*A*a^2*c
^2*d*m^2*n*x*x^m*x^n*e^m + 620*B*a^2*c^3*m*n^2*x*x^m*x^n*e^m + 1240*A*a*b*c
^3*m*n^2*x*x^m*x^n*e^m + 1860*A*a^2*c^2*d*m*n^2*x*x^m*x^n*e^m + 580*B*a^2*c
^3*n^3*x*x^m*x^n*e^m + 1160*A*a*b*c^3*n^3*x*x^m*x^n*e^m + 1740*A*a^2*c^2*d*
n^3*x*x^m*x^n*e^m + 20*A*a^2*c^3*m^3*x*x^m*e^m + 210*A*a^2*c^3*m^2*n*x*x^m*
e^m + 700*A*a^2*c^3*m*n^2*x*x^m*e^m + 735*A*a^2*c^3*n^3*x*x^m*e^m + 15*B*b^
2*d^3*m^2*x*x^m*x^(6*n)*e^m + 75*B*b^2*d^3*m*n*x*x^m*x^(6*n)*e^m + 85*B*b^2
*d^3*n^2*x*x^m*x^(6*n)*e^m + 45*B*b^2*c*d^2*m^2*x*x^m*x^(5*n)*e^m + 30*B*a*
b*d^3*m^2*x*x^m*x^(5*n)*e^m + 15*A*b^2*d^3*m^2*x*x^m*x^(5*n)*e^m + 240*B*b^
2*c*d^2*m*n*x*x^m*x^(5*n)*e^m + 160*B*a*b*d^3*m*n*x*x^m*x^(5*n)*e^m + 80*A*
b^2*d^3*m*n*x*x^m*x^(5*n)*e^m + 285*B*b^2*c*d^2*n^2*x*x^m*x^(5*n)*e^m + 190
*B*a*b*d^3*n^2*x*x^m*x^(5*n)*e^m + 95*A*b^2*d^3*n^2*x*x^m*x^(5*n)*e^m + 45*
B*b^2*c^2*d*m^2*x*x^m*x^(4*n)*e^m + 90*B*a*b*c*d^2*m^2*x*x^m*x^(4*n)*e^m +
45*A*b^2*c*d^2*m^2*x*x^m*x^(4*n)*e^m + 15*B*a^2*d^3*m^2*x*x^m*x^(4*n)*e^m +
 30*A*a*b*d^3*m^2*x*x^m*x^(4*n)*e^m + 255*B*b^2*c^2*d*m*n*x*x^m*x^(4*n)*e^m
 + 510*B*a*b*c*d^2*m*n*x*x^m*x^(4*n)*e^m + 255*A*b^2*c*d^2*m*n*x*x^m*x^(4*n
)*e^m + 85*B*a^2*d^3*m*n*x*x^m*x^(4*n)*e^m + 170*A*a*b*d^3*m*n*x*x^m*x^(4*n
)*e^m + 321*B*b^2*c^2*d*n^2*x*x^m*x^(4*n)*e^m + 642*B*a*b*c*d^2*n^2*x*x^m*x
^(4*n)*e^m + 321*A*b^2*c*d^2*n^2*x*x^m*x^(4*n)*e^m + 107*B*a^2*d^3*n^2*x*x^
m*x^(4*n)*e^m + 214*A*a*b*d^3*n^2*x*x^m*x^(4*n)*e^m + 15*B*b^2*c^3*m^2*x*x^
m*x^(3*n)*e^m + 90*B*a*b*c^2*d*m^2*x*x^m*x^(3*n)*e^m + 45*A*b^2*c^2*d*m^2*x
*x^m*x^(3*n)*e^m + 45*B*a^2*c*d^2*m^2*x*x^m*x^(3*n)*e^m + 90*A*a*b*c*d^2*m^
2*x*x^m*x^(3*n)*e^m + 15*A*a^2*d^3*m^2*x*x^m*x^(3*n)*e^m + 90*B*b^2*c^3*m*n
*x*x^m*x^(3*n)*e^m + 540*B*a*b*c^2*d*m*n*x*x^m*x^(3*n)*e^m + 270*A*b^2*c^2*
d*m*n*x*x^m*x^(3*n)*e^m + 270*B*a^2*c*d^2*m*n*x*x^m*x^(3*n)*e^m + 540*A*a*b
*c*d^2*m*n*x*x^m*x^(3*n)*e^m + 90*A*a^2*d^3*m*n*x*x^m*x^(3*n)*e^m + 121*B*b
^2*c^3*n^2*x*x^m*x^(3*n)*e^m + 726*B*a*b*c^2*d*n^2*x*x^m*x^(3*n)*e^m + 363*
A*b^2*c^2*d*n^2*x*x^m*x^(3*n)*e^m + 363*B*a^2*c*d^2*n^2*x*x^m*x^(3*n)*e^m +
 726*A*a*b*c*d^2*n^2*x*x^m*x^(3*n)*e^m + 121*A*a^2*d^3*n^2*x*x^m*x^(3*n)*e^
m + 30*B*a*b*c^3*m^2*x*x^m*x^(2*n)*e^m + 15*A*b^2*c^3*m^2*x*x^m*x^(2*n)*e^m

```
 + 45*B*a^2*c^2*d*m^2*x*x^m*x^(2*n)*e^m + 90*A*a*b*c^2*d*m^2*x*x^m*x^(2*n)*
e^m + 45*A*a^2*c*d^2*m^2*x*x^m*x^(2*n)*e^m + 190*B*a*b*c^3*m*n*x*x^m*x^(2*n
)*e^m + 95*A*b^2*c^3*m*n*x*x^m*x^(2*n)*e^m + 285*B*a^2*c^2*d*m*n*x*x^m*x^(2
*n)*e^m + 570*A*a*b*c^2*d*m*n*x*x^m*x^(2*n)*e^m + 285*A*a^2*c*d^2*m*n*x*x^m
*x^(2*n)*e^m + 274*B*a*b*c^3*n^2*x*x^m*x^(2*n)*e^m + 137*A*b^2*c^3*n^2*x*x^
m*x^(2*n)*e^m + 411*B*a^2*c^2*d*n^2*x*x^m*x^(2*n)*e^m + 822*A*a*b*c^2*d*n^2
*x*x^m*x^(2*n)*e^m + 411*A*a^2*c*d^2*n^2*x*x^m*x^(2*n)*e^m + 15*B*a^2*c^3*m
^2*x*x^m*x^n*e^m + 30*A*a*b*c^3*m^2*x*x^m*x^n*e^m + 45*A*a^2*c^2*d*m^2*x*x^
m*x^n*e^m + 100*B*a^2*c^3*m*n*x*x^m*x^n*e^m + 200*A*a*b*c^3*m*n*x*x^m*x^n*e
^m + 300*A*a^2*c^2*d*m*n*x*x^m*x^n*e^m + 155*B*a^2*c^3*n^2*x*x^m*x^n*e^m +
310*A*a*b*c^3*n^2*x*x^m*x^n*e^m + 465*A*a^2*c^2*d*n^2*x*x^m*x^n*e^m + 15*A*
a^2*c^3*m^2*x*x^m*e^m + 105*A*a^2*c^3*m*n*x*x^m*e^m + 175*A*a^2*c^3*n^2*x*x
^m*e^m + 6*B*b^2*d^3*m*x*x^m*x^(6*n)*e^m + 15*B*b^2*d^3*n*x*x^m*x^(6*n)*e^m
 + 18*B*b^2*c*d^2*m*x*x^m*x^(5*n)*e^m + 12*B*a*b*d^3*m*x*x^m*x^(5*n)*e^m +
6*A*b^2*d^3*m*x*x^m*x^(5*n)*e^m + 48*B*b^2*c*d^2*n*x*x^m*x^(5*n)*e^m + 32*B
*a*b*d^3*n*x*x^m*x^(5*n)*e^m + 16*A*b^2*d^3*n*x*x^m*x^(5*n)*e^m + 18*B*b^2*
c^2*d*m*x*x^m*x^(4*n)*e^m + 36*B*a*b*c*d^2*m*x*x^m*x^(4*n)*e^m + 18*A*b^2*c
*d^2*m*x*x^m*x^(4*n)*e^m + 6*B*a^2*d^3*m*x*x^m*x^(4*n)*e^m + 12*A*a*b*d^3*m
*x*x^m*x^(4*n)*e^m + 51*B*b^2*c^2*d*n*x*x^m*x^(4*n)*e^m + 102*B*a*b*c*d^2*n
*x*x^m*x^(4*n)*e^m + 51*A*b^2*c*d^2*n*x*x^m*x^(4*n)*e^m + 17*B*a^2*d^3*n*x*
x^m*x^(4*n)*e^m + 34*A*a*b*d^3*n*x*x^m*x^(4*n)*e^m + 6*B*b^2*c^3*m*x*x^m*x^
(3*n)*e^m + 36*B*a*b*c^2*d*m*x*x^m*x^(3*n)*e^m + 18*A*b^2*c^2*d*m*x*x^m*x^(
3*n)*e^m + 18*B*a^2*c*d^2*m*x*x^m*x^(3*n)*e^m + 36*A*a*b*c*d^2*m*x*x^m*x^(3
*n)*e^m + 6*A*a^2*d^3*m*x*x^m*x^(3*n)*e^m + 18*B*b^2*c^3*n*x*x^m*x^(3*n)*e^
m + 108*B*a*b*c^2*d*n*x*x^m*x^(3*n)*e^m + 54*A*b^2*c^2*d*n*x*x^m*x^(3*n)*e^
m + 54*B*a^2*c*d^2*n*x*x^m*x^(3*n)*e^m + 108*A*a*b*c*d^2*n*x*x^m*x^(3*n)*e^
m + 18*A*a^2*d^3*n*x*x^m*x^(3*n)*e^m + 12*B*a*b*c^3*m*x*x^m*x^(2*n)*e^m + 6
*A*b^2*c^3*m*x*x^m*x^(2*n)*e^m + 18*B*a^2*c^2*d*m*x*x^m*x^(2*n)*e^m + 36*A*
a*b*c^2*d*m*x*x^m*x^(2*n)*e^m + 18*A*a^2*c*d^2*m*x*x^m*x^(2*n)*e^m + 38*B*a
*b*c^3*n*x*x^m*x^(2*n)*e^m + 19*A*b^2*c^3*n*x*x^m*x^(2*n)*e^m + 57*B*a^2*c^
2*d*n*x*x^m*x^(2*n)*e^m + 114*A*a*b*c^2*d*n*x*x^m*x^(2*n)*e^m + 57*A*a^2*c*
d^2*n*x*x^m*x^(2*n)*e^m + 6*B*a^2*c^3*m*x*x^m*x^n*e^m + 12*A*a*b*c^3*m*x*x^
m*x^n*e^m + 18*A*a^2*c^2*d*m*x*x^m*x^n*e^m + 20*B*a^2*c^3*n*x*x^m*x^n*e^m +
 40*A*a*b*c^3*n*x*x^m*x^n*e^m + 60*A*a^2*c^2*d*n*x*x^m*x^n*e^m + 6*A*a^2*c^
3*m*x*x^m*e^m + 21*A*a^2*c^3*n*x*x^m*e^m + B*b^2*d^3*x*x^m*x^(6*n)*e^m + 3*
B*b^2*c*d^2*x*x^m*x^(5*n)*e^m + 2*B*a*b*d^3*x*x^m*x^(5*n)*e^m + A*b^2*d^3*x
*x^m*x^(5*n)*e^m + 3*B*b^2*c^2*d*x*x^m*x^(4*n)*e^m + 6*B*a*b*c*d^2*x*x^m*x^
(4*n)*e^m + 3*A*b^2*c*d^2*x*x^m*x^(4*n)*e^m + B*a^2*d^3*x*x^m*x^(4*n)*e^m +
 2*A*a*b*d^3*x*x^m*x^(4*n)*e^m + B*b^2*c^3*x*x^m*x^(3*n)*e^m + 6*B*a*b*c^2*
d*x*x^m*x^(3*n)*e^m + 3*A*b^2*c^2*d*x*x^m*x^(3*n)*e^m + 3*B*a^2*c*d^2*x*x^m
*x^(3*n)*e^m + 6*A*a*b*c*d^2*x*x^m*x^(3*n)*e^m + A*a^2*d^3*x*x^m*x^(3*n)*e^
m + 2*B*a*b*c^3*x*x^m*x^(2*n)*e^m + A*b^2*c^3*x*x^m*x^(2*n)*e^m + 3*B*a^2*c
^2*d*x*x^m*x^(2*n)*e^m + 6*A*a*b*c^2*d*x*x^m*x^(2*n)*e^m + 3*A*a^2*c*d^2*x*
x^m*x^(2*n)*e^m + B*a^2*c^3*x*x^m*x^n*e^m + 2*A*a*b*c^3*x*x^m*x^n*e^m + 3*A
*a^2*c^2*d*x*x^m*x^n*e^m + A*a^2*c^3*x*x^m*e^m)/(m^7 + 21*m^6*n + 175*m^5*n
^2 + 735*m^4*n^3 + 1624*m^3*n^4 + 1764*m^2*n^5 + 720*m*n^6 + 7*m^6 + 126*m^
5*n + 875*m^4*n^2 + 2940*m^3*n^3 + 4872*m^2*n^4 + 3528*m*n^5 + 720*n^6 + 21
*m^5 + 315*m^4*n + 1750*m^3*n^2 + 4410*m^2*n^3 + 4872*m*n^4 + 1764*n^5 + 35
*m^4 + 420*m^3*n + 1750*m^2*n^2 + 2940*m*n^3 + 1624*n^4 + 35*m^3 + 315*m^2*
n + 875*m*n^2 + 735*n^3 + 21*m^2 + 126*m*n + 175*n^2 + 7*m + 21*n + 1)
```

maple [C] time = 0.25, size = 11389, normalized size = 36.74

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(b*x^n+a)^2*(B*x^n+A)*(d*x^n+c)^3,x)
```

```
[Out] result too large to display
```

maxima [B] time = 0.92, size = 748, normalized size = 2.41

Verification of antiderivative is not currently implemented for this CAS.

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

```
[Out] B*b^2*d^3*e^m*x*e^(m*log(x) + 6*n*log(x))/(m + 6*n + 1) + 3*B*b^2*c*d^2*e^m
*x*e^(m*log(x) + 5*n*log(x))/(m + 5*n + 1) + 2*B*a*b*d^3*e^m*x*e^(m*log(x)
+ 5*n*log(x))/(m + 5*n + 1) + A*b^2*d^3*e^m*x*e^(m*log(x) + 5*n*log(x))/(m
+ 5*n + 1) + 3*B*b^2*c^2*d*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) +
6*B*a*b*c*d^2*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) + 3*A*b^2*c*d^2
*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) + B*a^2*d^3*e^m*x*e^(m*log(x
) + 4*n*log(x))/(m + 4*n + 1) + 2*A*a*b*d^3*e^m*x*e^(m*log(x) + 4*n*log(x))
/(m + 4*n + 1) + B*b^2*c^3*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) +
6*B*a*b*c^2*d*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + 3*A*b^2*c^2*d
*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + 3*B*a^2*c*d^2*e^m*x*e^(m*l
og(x) + 3*n*log(x))/(m + 3*n + 1) + 6*A*a*b*c*d^2*e^m*x*e^(m*log(x) + 3*n*l
og(x))/(m + 3*n + 1) + A*a^2*d^3*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n +
 1) + 2*B*a*b*c^3*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + A*b^2*c^3
*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 3*B*a^2*c^2*d*e^m*x*e^(m*l
og(x) + 2*n*log(x))/(m + 2*n + 1) + 6*A*a*b*c^2*d*e^m*x*e^(m*log(x) + 2*n*l
og(x))/(m + 2*n + 1) + 3*A*a^2*c*d^2*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2
*n + 1) + B*a^2*c^3*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1) + 2*A*a*b*c^3
*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1) + 3*A*a^2*c^2*d*e^m*x*e^(m*log(x
) + n*log(x))/(m + n + 1) + (e*x)^(m + 1)*A*a^2*c^3/(e*(m + 1))
```

mupad [B] time = 6.41, size = 1882, normalized size = 6.07

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(A + B*x^n)*(a + b*x^n)^2*(c + d*x^n)^3,x)
```

```
[Out] (x*x^(3*n)*(e*x)^m*(A*a^2*d^3 + B*b^2*c^3 + 3*A*b^2*c^2*d + 3*B*a^2*c*d^2 +
 6*A*a*b*c*d^2 + 6*B*a*b*c^2*d)*(5*m + 18*n + 72*m*n + 363*m*n^2 + 108*m^2*
n + 744*m*n^3 + 72*m^3*n + 508*m*n^4 + 18*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 +
 m^5 + 121*n^2 + 372*n^3 + 508*n^4 + 240*n^5 + 363*m^2*n^2 + 372*m^2*n^3 +
121*m^3*n^2 + 1))/(6*m + 21*n + 105*m*n + 700*m*n^2 + 210*m^2*n + 2205*m*n^
3 + 210*m^3*n + 3248*m*n^4 + 105*m^4*n + 1764*m*n^5 + 21*m^5*n + 15*m^2 + 2
0*m^3 + 15*m^4 + 6*m^5 + m^6 + 175*n^2 + 735*n^3 + 1624*n^4 + 1764*n^5 + 72
0*n^6 + 1050*m^2*n^2 + 2205*m^2*n^3 + 700*m^3*n^2 + 1624*m^2*n^4 + 735*m^3*
n^3 + 175*m^4*n^2 + 1) + (A*a^2*c^3*x*(e*x)^m)/(m + 1) + (c*x*x^(2*n)*(e*x)
^m*(3*A*a^2*d^2 + A*b^2*c^2 + 2*B*a*b*c^2 + 3*B*a^2*c*d + 6*A*a*b*c*d)*(5*m
 + 19*n + 76*m*n + 411*m*n^2 + 114*m^2*n + 922*m*n^3 + 76*m^3*n + 702*m*n^4
 + 19*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 137*n^2 + 461*n^3 + 702*n^4 +
 360*n^5 + 411*m^2*n^2 + 461*m^2*n^3 + 137*m^3*n^2 + 1))/(6*m + 21*n + 105*
m*n + 700*m*n^2 + 210*m^2*n + 2205*m*n^3 + 210*m^3*n + 3248*m*n^4 + 105*m^4
*n + 1764*m*n^5 + 21*m^5*n + 15*m^2 + 20*m^3 + 15*m^4 + 6*m^5 + m^6 + 175*n
^2 + 735*n^3 + 1624*n^4 + 1764*n^5 + 720*n^6 + 1050*m^2*n^2 + 2205*m^2*n^3
+ 700*m^3*n^2 + 1624*m^2*n^4 + 735*m^3*n^3 + 175*m^4*n^2 + 1) + (d*x*x^(4*n
)*(e*x)^m*(B*a^2*d^2 + 3*B*b^2*c^2 + 2*A*a*b*d^2 + 3*A*b^2*c*d + 6*B*a*b*c*
d)*(5*m + 17*n + 68*m*n + 321*m*n^2 + 102*m^2*n + 614*m*n^3 + 68*m^3*n + 39
6*m*n^4 + 17*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 107*n^2 + 307*n^3 + 39
6*n^4 + 180*n^5 + 321*m^2*n^2 + 307*m^2*n^3 + 107*m^3*n^2 + 1))/(6*m + 21*n
 + 105*m*n + 700*m*n^2 + 210*m^2*n + 2205*m*n^3 + 210*m^3*n + 3248*m*n^4 +
105*m^4*n + 1764*m*n^5 + 21*m^5*n + 15*m^2 + 20*m^3 + 15*m^4 + 6*m^5 + m^6
+ 175*n^2 + 735*n^3 + 1624*n^4 + 1764*n^5 + 720*n^6 + 1050*m^2*n^2 + 2205*m
^2*n^3 + 700*m^3*n^2 + 1624*m^2*n^4 + 735*m^3*n^3 + 175*m^4*n^2 + 1) + (a*c
^2*x*x^n*(e*x)^m*(3*A*a*d + 2*A*b*c + B*a*c)*(5*m + 20*n + 80*m*n + 465*m*n
```

```
^2 + 120*m^2*n + 1160*m*n^3 + 80*m^3*n + 1044*m*n^4 + 20*m^4*n + 10*m^2 + 1
0*m^3 + 5*m^4 + m^5 + 155*n^2 + 580*n^3 + 1044*n^4 + 720*n^5 + 465*m^2*n^2
+ 580*m^2*n^3 + 155*m^3*n^2 + 1))/(6*m + 21*n + 105*m*n + 700*m*n^2 + 210*m
^2*n + 2205*m*n^3 + 210*m^3*n + 3248*m*n^4 + 105*m^4*n + 1764*m*n^5 + 21*m^
5*n + 15*m^2 + 20*m^3 + 15*m^4 + 6*m^5 + m^6 + 175*n^2 + 735*n^3 + 1624*n^4
 + 1764*n^5 + 720*n^6 + 1050*m^2*n^2 + 2205*m^2*n^3 + 700*m^3*n^2 + 1624*m^
2*n^4 + 735*m^3*n^3 + 175*m^4*n^2 + 1) + (b*d^2*x*x^(5*n)*(e*x)^m*(A*b*d +
2*B*a*d + 3*B*b*c)*(5*m + 16*n + 64*m*n + 285*m*n^2 + 96*m^2*n + 520*m*n^3
+ 64*m^3*n + 324*m*n^4 + 16*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 95*n^2
+ 260*n^3 + 324*n^4 + 144*n^5 + 285*m^2*n^2 + 260*m^2*n^3 + 95*m^3*n^2 + 1)
)/(6*m + 21*n + 105*m*n + 700*m*n^2 + 210*m^2*n + 2205*m*n^3 + 210*m^3*n +
3248*m*n^4 + 105*m^4*n + 1764*m*n^5 + 21*m^5*n + 15*m^2 + 20*m^3 + 15*m^4 +
 6*m^5 + m^6 + 175*n^2 + 735*n^3 + 1624*n^4 + 1764*n^5 + 720*n^6 + 1050*m^2
*n^2 + 2205*m^2*n^3 + 700*m^3*n^2 + 1624*m^2*n^4 + 735*m^3*n^3 + 175*m^4*n^
2 + 1) + (B*b^2*d^3*x*x^(6*n)*(e*x)^m*(5*m + 15*n + 60*m*n + 255*m*n^2 + 90
*m^2*n + 450*m*n^3 + 60*m^3*n + 274*m*n^4 + 15*m^4*n + 10*m^2 + 10*m^3 + 5*
m^4 + m^5 + 85*n^2 + 225*n^3 + 274*n^4 + 120*n^5 + 255*m^2*n^2 + 225*m^2*n^
3 + 85*m^3*n^2 + 1))/(6*m + 21*n + 105*m*n + 700*m*n^2 + 210*m^2*n + 2205*m
*n^3 + 210*m^3*n + 3248*m*n^4 + 105*m^4*n + 1764*m*n^5 + 21*m^5*n + 15*m^2
+ 20*m^3 + 15*m^4 + 6*m^5 + m^6 + 175*n^2 + 735*n^3 + 1624*n^4 + 1764*n^5 +
 720*n^6 + 1050*m^2*n^2 + 2205*m^2*n^3 + 700*m^3*n^2 + 1624*m^2*n^4 + 735*m
^3*n^3 + 175*m^4*n^2 + 1)
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)**m*(a+b*x**n)**2*(A+B*x**n)*(c+d*x**n)**3,x)
```

```
[Out] Timed out
```

3.11 $\int (ex)^m (a+bx^n)(A+Bx^n)(c+dx^n)^3 \, dx$

Optimal. Leaf size=210

$$\frac{c^2x^{n+1}(ex)^m(3aAd+aBc+Abc)}{m+n+1}+\frac{d^2x^{4n+1}(ex)^m(aBd+Abd+3bBc)}{m+4n+1}+\frac{cx^{2n+1}(ex)^m(3ad(Ad+Bc)+bc(3Ad+Bc)}{m+2n+1}$$

Rubi [A] time = 0.26, antiderivative size = 210, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 3, integrand size = 29, $\frac{\text{number of rules}}{\text{integrand size}} = 0.103$, Rules used = {570, 20, 30}

$$\frac{c^2x^{n+1}(ex)^m(3aAd+aBc+Abc)}{m+n+1}+\frac{d^2x^{4n+1}(ex)^m(aBd+Abd+3bBc)}{m+4n+1}+\frac{cx^{2n+1}(ex)^m(3ad(Ad+Bc)+bc(3Ad+Bc))}{m+2n+1}+\frac{dx^{3n+1}(ex)^m(ad(Ad+3Bc)+3bc(Ad+Bc))}{m+3n+1}+\frac{aAc^3(ex)^{m+1}}{e(m+1)}+\frac{bBd^3x^{5n+1}(ex)^m}{m+5n+1}$$

Antiderivative was successfully verified.

```
[In] Int[(e*x)^m*(a + b*x^n)*(A + B*x^n)*(c + d*x^n)^3,x]
```

```
[Out] (c^2*(A*b*c + a*B*c + 3*a*A*d)*x^(1 + n)*(e*x)^m)/(1 + m + n) + (c*(3*a*d*(
B*c + A*d) + b*c*(B*c + 3*A*d))*x^(1 + 2*n)*(e*x)^m)/(1 + m + 2*n) + (d*(3*
b*c*(B*c + A*d) + a*d*(3*B*c + A*d))*x^(1 + 3*n)*(e*x)^m)/(1 + m + 3*n) + (
d^2*(3*b*B*c + A*b*d + a*B*d)*x^(1 + 4*n)*(e*x)^m)/(1 + m + 4*n) + (b*B*d^3
*x^(1 + 5*n)*(e*x)^m)/(1 + m + 5*n) + (a*A*c^3*(e*x)^(1 + m))/(e*(1 + m))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 30

```
Int[(x_)^(m_.), x_Symbol] :> Simp[x^(m + 1)/(m + 1), x] /; FreeQ[m, x] && N
eQ[m, -1]
```

Rule 570

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

Rubi steps

$$\begin{aligned}
\int (ex)^m (a+bx^n)(A+Bx^n)(c+dx^n)^3 \, dx &= \int \left(aAc^3(ex)^m+c^2(Abc+aBc+3aAd)x^n(ex)^m+c(3ad(Bc+Ad)+\right. \\
&= \frac{aAc^3(ex)^{1+m}}{e(1+m)}+\left(bBd^3\right)\int x^{5n}(ex)^m\,dx+\left(c^2(Abc+aBc+3aAd)\right)\int \\
&= \frac{aAc^3(ex)^{1+m}}{e(1+m)}+\left(bBd^3x^{-m}(ex)^m\right)\int x^{m+5n}\,dx+\left(c^2(Abc+aBc+3aA\right. \\
&= \frac{c^2(Abc+aBc+3aAd)x^{1+n}(ex)^m}{1+m+n}+\frac{c(3ad(Bc+Ad)+bc(Bc+3Ad)}{1+m+2n}
\end{aligned}$$

Mathematica [A] time = 0.75, size = 172, normalized size = 0.82

$$x(ex)^m\left(\frac{c^2x^n(3aAd+aBc+Abc)}{m+n+1}+\frac{d^2x^{4n}(aBd+Abd+3bBc)}{m+4n+1}+\frac{cx^{2n}(3ad(Ad+Bc)+bc(3Ad+Bc))}{m+2n+1}+\frac{dx^{3n}(ad(Ad+3Bc)+3bc(Ad+Bc))}{m+3n+1}+\frac{aAc^3}{m+1}+\frac{bBd^3x^{5n}}{m+5n+1}\right)$$

Antiderivative was successfully verified.

```
[In] Integrate[(e*x)^m*(a + b*x^n)*(A + B*x^n)*(c + d*x^n)^3,x]
```

```
[Out] x*(e*x)^m*((a*A*c^3)/(1 + m) + (c^2*(A*b*c + a*B*c + 3*a*A*d)*x^n)/(1 + m +
 n) + (c*(3*a*d*(B*c + A*d) + b*c*(B*c + 3*A*d))*x^(2*n))/(1 + m + 2*n) + (
d*(3*b*c*(B*c + A*d) + a*d*(3*B*c + A*d))*x^(3*n))/(1 + m + 3*n) + (d^2*(3*
b*B*c + A*b*d + a*B*d)*x^(4*n))/(1 + m + 4*n) + (b*B*d^3*x^(5*n))/(1 + m +
5*n))
```

IntegrateAlgebraic [F]   time = 0.64, size = 0, normalized size = 0.00

$$\int (ex)^m (a + bx^n) (A + Bx^n) (c + dx^n)^3 \, dx$$

Verification is not applicable to the result.

```
[In] IntegrateAlgebraic[(e*x)^m*(a + b*x^n)*(A + B*x^n)*(c + d*x^n)^3,x]
```

```
[Out] Defer[IntegrateAlgebraic][(e*x)^m*(a + b*x^n)*(A + B*x^n)*(c + d*x^n)^3, x]
```

fricas [B]   time = 0.50, size = 2833, normalized size = 13.49

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)*(A+B*x^n)*(c+d*x^n)^3,x, algorithm="fricas")
```

```
[Out] ((B*b*d^3*m^5 + 5*B*b*d^3*m^4 + 10*B*b*d^3*m^3 + 10*B*b*d^3*m^2 + 5*B*b*d^3
*m + B*b*d^3 + 24*(B*b*d^3*m + B*b*d^3)*n^4 + 50*(B*b*d^3*m^2 + 2*B*b*d^3*m
 + B*b*d^3)*n^3 + 35*(B*b*d^3*m^3 + 3*B*b*d^3*m^2 + 3*B*b*d^3*m + B*b*d^3)*
n^2 + 10*(B*b*d^3*m^4 + 4*B*b*d^3*m^3 + 6*B*b*d^3*m^2 + 4*B*b*d^3*m + B*b*d
^3)*n)*x*x^(5*n)*e^(m*log(e) + m*log(x)) + ((3*B*b*c*d^2 + (B*a + A*b)*d^3)
*m^5 + 3*B*b*c*d^2 + 5*(3*B*b*c*d^2 + (B*a + A*b)*d^3)*m^4 + 30*(3*B*b*c*d^
2 + (B*a + A*b)*d^3 + (3*B*b*c*d^2 + (B*a + A*b)*d^3)*m)*n^4 + (B*a + A*b)*
d^3 + 10*(3*B*b*c*d^2 + (B*a + A*b)*d^3)*m^3 + 61*(3*B*b*c*d^2 + (B*a + A*b
)*d^3 + (3*B*b*c*d^2 + (B*a + A*b)*d^3)*m^2 + 2*(3*B*b*c*d^2 + (B*a + A*b)*
d^3)*m)*n^3 + 10*(3*B*b*c*d^2 + (B*a + A*b)*d^3)*m^2 + 41*(3*B*b*c*d^2 + (B
*a + A*b)*d^3 + (3*B*b*c*d^2 + (B*a + A*b)*d^3)*m^3 + 3*(3*B*b*c*d^2 + (B*a
 + A*b)*d^3)*m^2 + 3*(3*B*b*c*d^2 + (B*a + A*b)*d^3)*m)*n^2 + 5*(3*B*b*c*d^
2 + (B*a + A*b)*d^3)*m + 11*(3*B*b*c*d^2 + (3*B*b*c*d^2 + (B*a + A*b)*d^3)*
m^4 + (B*a + A*b)*d^3 + 4*(3*B*b*c*d^2 + (B*a + A*b)*d^3)*m^3 + 6*(3*B*b*c*
d^2 + (B*a + A*b)*d^3)*m^2 + 4*(3*B*b*c*d^2 + (B*a + A*b)*d^3)*m)*n)*x*x^(4
*n)*e^(m*log(e) + m*log(x)) + ((3*B*b*c^2*d + A*a*d^3 + 3*(B*a + A*b)*c*d^2
)*m^5 + 3*B*b*c^2*d + A*a*d^3 + 5*(3*B*b*c^2*d + A*a*d^3 + 3*(B*a + A*b)*c*
d^2)*m^4 + 40*(3*B*b*c^2*d + A*a*d^3 + 3*(B*a + A*b)*c*d^2 + (3*B*b*c^2*d +
 A*a*d^3 + 3*(B*a + A*b)*c*d^2)*m)*n^4 + 3*(B*a + A*b)*c*d^2 + 10*(3*B*b*c^
2*d + A*a*d^3 + 3*(B*a + A*b)*c*d^2)*m^3 + 78*(3*B*b*c^2*d + A*a*d^3 + 3*(B
*a + A*b)*c*d^2 + (3*B*b*c^2*d + A*a*d^3 + 3*(B*a + A*b)*c*d^2)*m^2 + 2*(3*
B*b*c^2*d + A*a*d^3 + 3*(B*a + A*b)*c*d^2)*m)*n^3 + 10*(3*B*b*c^2*d + A*a*d
^3 + 3*(B*a + A*b)*c*d^2)*m^2 + 49*(3*B*b*c^2*d + A*a*d^3 + 3*(B*a + A*b)*c
*d^2 + (3*B*b*c^2*d + A*a*d^3 + 3*(B*a + A*b)*c*d^2)*m^3 + 3*(3*B*b*c^2*d +
 A*a*d^3 + 3*(B*a + A*b)*c*d^2)*m^2 + 3*(3*B*b*c^2*d + A*a*d^3 + 3*(B*a + A
*b)*c*d^2)*m)*n^2 + 5*(3*B*b*c^2*d + A*a*d^3 + 3*(B*a + A*b)*c*d^2)*m + 12*
(3*B*b*c^2*d + A*a*d^3 + (3*B*b*c^2*d + A*a*d^3 + 3*(B*a + A*b)*c*d^2)*m^4
+ 3*(B*a + A*b)*c*d^2 + 4*(3*B*b*c^2*d + A*a*d^3 + 3*(B*a + A*b)*c*d^2)*m^3
 + 6*(3*B*b*c^2*d + A*a*d^3 + 3*(B*a + A*b)*c*d^2)*m^2 + 4*(3*B*b*c^2*d + A
*a*d^3 + 3*(B*a + A*b)*c*d^2)*m)*n)*x*x^(3*n)*e^(m*log(e) + m*log(x)) + ((B
*b*c^3 + 3*A*a*c*d^2 + 3*(B*a + A*b)*c^2*d)*m^5 + B*b*c^3 + 3*A*a*c*d^2 + 5
*(B*b*c^3 + 3*A*a*c*d^2 + 3*(B*a + A*b)*c^2*d)*m^4 + 60*(B*b*c^3 + 3*A*a*c*
d^2 + 3*(B*a + A*b)*c^2*d + (B*b*c^3 + 3*A*a*c*d^2 + 3*(B*a + A*b)*c^2*d)*m
)*n^4 + 3*(B*a + A*b)*c^2*d + 10*(B*b*c^3 + 3*A*a*c*d^2 + 3*(B*a + A*b)*c^2
```

```
*d)*m^3 + 107*(B*b*c^3 + 3*A*a*c*d^2 + 3*(B*a + A*b)*c^2*d + (B*b*c^3 + 3*A
*a*c*d^2 + 3*(B*a + A*b)*c^2*d)*m^2 + 2*(B*b*c^3 + 3*A*a*c*d^2 + 3*(B*a + A
*b)*c^2*d)*m)*n^3 + 10*(B*b*c^3 + 3*A*a*c*d^2 + 3*(B*a + A*b)*c^2*d)*m^2 +
59*(B*b*c^3 + 3*A*a*c*d^2 + 3*(B*a + A*b)*c^2*d + (B*b*c^3 + 3*A*a*c*d^2 +
3*(B*a + A*b)*c^2*d)*m^3 + 3*(B*b*c^3 + 3*A*a*c*d^2 + 3*(B*a + A*b)*c^2*d)*
m^2 + 3*(B*b*c^3 + 3*A*a*c*d^2 + 3*(B*a + A*b)*c^2*d)*m)*n^2 + 5*(B*b*c^3 +
 3*A*a*c*d^2 + 3*(B*a + A*b)*c^2*d)*m + 13*(B*b*c^3 + 3*A*a*c*d^2 + (B*b*c^
3 + 3*A*a*c*d^2 + 3*(B*a + A*b)*c^2*d)*m^4 + 3*(B*a + A*b)*c^2*d + 4*(B*b*c
^3 + 3*A*a*c*d^2 + 3*(B*a + A*b)*c^2*d)*m^3 + 6*(B*b*c^3 + 3*A*a*c*d^2 + 3*
(B*a + A*b)*c^2*d)*m^2 + 4*(B*b*c^3 + 3*A*a*c*d^2 + 3*(B*a + A*b)*c^2*d)*m)
*n)*x*x^(2*n)*e^(m*log(e) + m*log(x)) + ((3*A*a*c^2*d + (B*a + A*b)*c^3)*m^
5 + 3*A*a*c^2*d + 5*(3*A*a*c^2*d + (B*a + A*b)*c^3)*m^4 + 120*(3*A*a*c^2*d
+ (B*a + A*b)*c^3 + (3*A*a*c^2*d + (B*a + A*b)*c^3)*m)*n^4 + (B*a + A*b)*c^
3 + 10*(3*A*a*c^2*d + (B*a + A*b)*c^3)*m^3 + 154*(3*A*a*c^2*d + (B*a + A*b)
*c^3 + (3*A*a*c^2*d + (B*a + A*b)*c^3)*m^2 + 2*(3*A*a*c^2*d + (B*a + A*b)*c
^3)*m)*n^3 + 10*(3*A*a*c^2*d + (B*a + A*b)*c^3)*m^2 + 71*(3*A*a*c^2*d + (B*
a + A*b)*c^3 + (3*A*a*c^2*d + (B*a + A*b)*c^3)*m^3 + 3*(3*A*a*c^2*d + (B*a
+ A*b)*c^3)*m^2 + 3*(3*A*a*c^2*d + (B*a + A*b)*c^3)*m)*n^2 + 5*(3*A*a*c^2*d
 + (B*a + A*b)*c^3)*m + 14*(3*A*a*c^2*d + (3*A*a*c^2*d + (B*a + A*b)*c^3)*m
^4 + (B*a + A*b)*c^3 + 4*(3*A*a*c^2*d + (B*a + A*b)*c^3)*m^3 + 6*(3*A*a*c^2
*d + (B*a + A*b)*c^3)*m^2 + 4*(3*A*a*c^2*d + (B*a + A*b)*c^3)*m)*n)*x*x^n*e
^(m*log(e) + m*log(x)) + (A*a*c^3*m^5 + 120*A*a*c^3*n^5 + 5*A*a*c^3*m^4 + 1
0*A*a*c^3*m^3 + 10*A*a*c^3*m^2 + 5*A*a*c^3*m + A*a*c^3 + 274*(A*a*c^3*m + A
*a*c^3)*n^4 + 225*(A*a*c^3*m^2 + 2*A*a*c^3*m + A*a*c^3)*n^3 + 85*(A*a*c^3*m
^3 + 3*A*a*c^3*m^2 + 3*A*a*c^3*m + A*a*c^3)*n^2 + 15*(A*a*c^3*m^4 + 4*A*a*c
^3*m^3 + 6*A*a*c^3*m^2 + 4*A*a*c^3*m + A*a*c^3)*n)*x*e^(m*log(e) + m*log(x)
))/(m^6 + 120*(m + 1)*n^5 + 6*m^5 + 274*(m^2 + 2*m + 1)*n^4 + 15*m^4 + 225*
(m^3 + 3*m^2 + 3*m + 1)*n^3 + 20*m^3 + 85*(m^4 + 4*m^3 + 6*m^2 + 4*m + 1)*n
^2 + 15*m^2 + 15*(m^5 + 5*m^4 + 10*m^3 + 10*m^2 + 5*m + 1)*n + 6*m + 1)
```

giac [B] time = 1.06, size = 6927, normalized size = 32.99

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)*(A+B*x^n)*(c+d*x^n)^3,x, algorithm="giac")
```

```
[Out] (B*b*d^3*m^5*x*x^m*x^(5*n)*e^m + 10*B*b*d^3*m^4*n*x*x^m*x^(5*n)*e^m + 35*B*
b*d^3*m^3*n^2*x*x^m*x^(5*n)*e^m + 50*B*b*d^3*m^2*n^3*x*x^m*x^(5*n)*e^m + 24
*B*b*d^3*m*n^4*x*x^m*x^(5*n)*e^m + 3*B*b*c*d^2*m^5*x*x^m*x^(4*n)*e^m + B*a*
d^3*m^5*x*x^m*x^(4*n)*e^m + A*b*d^3*m^5*x*x^m*x^(4*n)*e^m + 33*B*b*c*d^2*m^
4*n*x*x^m*x^(4*n)*e^m + 11*B*a*d^3*m^4*n*x*x^m*x^(4*n)*e^m + 11*A*b*d^3*m^4
*n*x*x^m*x^(4*n)*e^m + 123*B*b*c*d^2*m^3*n^2*x*x^m*x^(4*n)*e^m + 41*B*a*d^3
*m^3*n^2*x*x^m*x^(4*n)*e^m + 41*A*b*d^3*m^3*n^2*x*x^m*x^(4*n)*e^m + 183*B*b
*c*d^2*m^2*n^3*x*x^m*x^(4*n)*e^m + 61*B*a*d^3*m^2*n^3*x*x^m*x^(4*n)*e^m + 6
1*A*b*d^3*m^2*n^3*x*x^m*x^(4*n)*e^m + 90*B*b*c*d^2*m*n^4*x*x^m*x^(4*n)*e^m
+ 30*B*a*d^3*m*n^4*x*x^m*x^(4*n)*e^m + 30*A*b*d^3*m*n^4*x*x^m*x^(4*n)*e^m +
 3*B*b*c^2*d*m^5*x*x^m*x^(3*n)*e^m + 3*B*a*c*d^2*m^5*x*x^m*x^(3*n)*e^m + 3*
A*b*c*d^2*m^5*x*x^m*x^(3*n)*e^m + A*a*d^3*m^5*x*x^m*x^(3*n)*e^m + 36*B*b*c^
2*d*m^4*n*x*x^m*x^(3*n)*e^m + 36*B*a*c*d^2*m^4*n*x*x^m*x^(3*n)*e^m + 36*A*b
*c*d^2*m^4*n*x*x^m*x^(3*n)*e^m + 12*A*a*d^3*m^4*n*x*x^m*x^(3*n)*e^m + 147*B
*b*c^2*d*m^3*n^2*x*x^m*x^(3*n)*e^m + 147*B*a*c*d^2*m^3*n^2*x*x^m*x^(3*n)*e^
m + 147*A*b*c*d^2*m^3*n^2*x*x^m*x^(3*n)*e^m + 49*A*a*d^3*m^3*n^2*x*x^m*x^(3
*n)*e^m + 234*B*b*c^2*d*m^2*n^3*x*x^m*x^(3*n)*e^m + 234*B*a*c*d^2*m^2*n^3*x
*x^m*x^(3*n)*e^m + 234*A*b*c*d^2*m^2*n^3*x*x^m*x^(3*n)*e^m + 78*A*a*d^3*m^2
*n^3*x*x^m*x^(3*n)*e^m + 120*B*b*c^2*d*m*n^4*x*x^m*x^(3*n)*e^m + 120*B*a*c*
d^2*m*n^4*x*x^m*x^(3*n)*e^m + 120*A*b*c*d^2*m*n^4*x*x^m*x^(3*n)*e^m + 40*A*
a*d^3*m*n^4*x*x^m*x^(3*n)*e^m + B*b*c^3*m^5*x*x^m*x^(2*n)*e^m + 3*B*a*c^2*d
*m^5*x*x^m*x^(2*n)*e^m + 3*A*b*c^2*d*m^5*x*x^m*x^(2*n)*e^m + 3*A*a*c*d^2*m^
5*x*x^m*x^(2*n)*e^m + 13*B*b*c^3*m^4*n*x*x^m*x^(2*n)*e^m + 39*B*a*c^2*d*m^4
```

```
*n*x*x^m*x^(2*n)*e^m + 39*A*b*c^2*d*m^4*n*x*x^m*x^(2*n)*e^m + 39*A*a*c*d^2*
m^4*n*x*x^m*x^(2*n)*e^m + 59*B*b*c^3*m^3*n^2*x*x^m*x^(2*n)*e^m + 177*B*a*c^
2*d*m^3*n^2*x*x^m*x^(2*n)*e^m + 177*A*b*c^2*d*m^3*n^2*x*x^m*x^(2*n)*e^m + 1
77*A*a*c*d^2*m^3*n^2*x*x^m*x^(2*n)*e^m + 107*B*b*c^3*m^2*n^3*x*x^m*x^(2*n)*
e^m + 321*B*a*c^2*d*m^2*n^3*x*x^m*x^(2*n)*e^m + 321*A*b*c^2*d*m^2*n^3*x*x^m
*x^(2*n)*e^m + 321*A*a*c*d^2*m^2*n^3*x*x^m*x^(2*n)*e^m + 60*B*b*c^3*m*n^4*x
*x^m*x^(2*n)*e^m + 180*B*a*c^2*d*m*n^4*x*x^m*x^(2*n)*e^m + 180*A*b*c^2*d*m*
n^4*x*x^m*x^(2*n)*e^m + 180*A*a*c*d^2*m*n^4*x*x^m*x^(2*n)*e^m + B*a*c^3*m^5
*x*x^m*x^n*e^m + A*b*c^3*m^5*x*x^m*x^n*e^m + 3*A*a*c^2*d*m^5*x*x^m*x^n*e^m
 + 14*B*a*c^3*m^4*n*x*x^m*x^n*e^m + 14*A*b*c^3*m^4*n*x*x^m*x^n*e^m + 42*A*a*
c^2*d*m^4*n*x*x^m*x^n*e^m + 71*B*a*c^3*m^3*n^2*x*x^m*x^n*e^m + 71*A*b*c^3*m
^3*n^2*x*x^m*x^n*e^m + 213*A*a*c^2*d*m^3*n^2*x*x^m*x^n*e^m + 154*B*a*c^3*m^
2*n^3*x*x^m*x^n*e^m + 154*A*b*c^3*m^2*n^3*x*x^m*x^n*e^m + 462*A*a*c^2*d*m^2
*n^3*x*x^m*x^n*e^m + 120*B*a*c^3*m*n^4*x*x^m*x^n*e^m + 120*A*b*c^3*m*n^4*x*
x^m*x^n*e^m + 360*A*a*c^2*d*m*n^4*x*x^m*x^n*e^m + A*a*c^3*m^5*x*x^m*e^m + 1
5*A*a*c^3*m^4*n*x*x^m*e^m + 85*A*a*c^3*m^3*n^2*x*x^m*e^m + 225*A*a*c^3*m^2*
n^3*x*x^m*e^m + 274*A*a*c^3*m*n^4*x*x^m*e^m + 120*A*a*c^3*n^5*x*x^m*e^m + 5
*B*b*d^3*m^4*x*x^m*x^(5*n)*e^m + 40*B*b*d^3*m^3*n*x*x^m*x^(5*n)*e^m + 105*B
*b*d^3*m^2*n^2*x*x^m*x^(5*n)*e^m + 100*B*b*d^3*m*n^3*x*x^m*x^(5*n)*e^m + 24
*B*b*d^3*n^4*x*x^m*x^(5*n)*e^m + 15*B*b*c*d^2*m^4*x*x^m*x^(4*n)*e^m + 5*B*a
*d^3*m^4*x*x^m*x^(4*n)*e^m + 5*A*b*d^3*m^4*x*x^m*x^(4*n)*e^m + 132*B*b*c*d^
2*m^3*n*x*x^m*x^(4*n)*e^m + 44*B*a*d^3*m^3*n*x*x^m*x^(4*n)*e^m + 44*A*b*d^3
*m^3*n*x*x^m*x^(4*n)*e^m + 369*B*b*c*d^2*m^2*n^2*x*x^m*x^(4*n)*e^m + 123*B*
a*d^3*m^2*n^2*x*x^m*x^(4*n)*e^m + 123*A*b*d^3*m^2*n^2*x*x^m*x^(4*n)*e^m + 3
66*B*b*c*d^2*m*n^3*x*x^m*x^(4*n)*e^m + 122*B*a*d^3*m*n^3*x*x^m*x^(4*n)*e^m
 + 122*A*b*d^3*m*n^3*x*x^m*x^(4*n)*e^m + 90*B*b*c*d^2*n^4*x*x^m*x^(4*n)*e^m
 + 30*B*a*d^3*n^4*x*x^m*x^(4*n)*e^m + 30*A*b*d^3*n^4*x*x^m*x^(4*n)*e^m + 15*
B*b*c^2*d*m^4*x*x^m*x^(3*n)*e^m + 15*B*a*c*d^2*m^4*x*x^m*x^(3*n)*e^m + 15*A
*b*c*d^2*m^4*x*x^m*x^(3*n)*e^m + 5*A*a*d^3*m^4*x*x^m*x^(3*n)*e^m + 144*B*b*
c^2*d*m^3*n*x*x^m*x^(3*n)*e^m + 144*B*a*c*d^2*m^3*n*x*x^m*x^(3*n)*e^m + 144
*A*b*c*d^2*m^3*n*x*x^m*x^(3*n)*e^m + 48*A*a*d^3*m^3*n*x*x^m*x^(3*n)*e^m + 4
41*B*b*c^2*d*m^2*n^2*x*x^m*x^(3*n)*e^m + 441*B*a*c*d^2*m^2*n^2*x*x^m*x^(3*n
)*e^m + 441*A*b*c*d^2*m^2*n^2*x*x^m*x^(3*n)*e^m + 147*A*a*d^3*m^2*n^2*x*x^m
*x^(3*n)*e^m + 468*B*b*c^2*d*m*n^3*x*x^m*x^(3*n)*e^m + 468*B*a*c*d^2*m*n^3*
x*x^m*x^(3*n)*e^m + 468*A*b*c*d^2*m*n^3*x*x^m*x^(3*n)*e^m + 156*A*a*d^3*m*n
^3*x*x^m*x^(3*n)*e^m + 120*B*b*c^2*d*n^4*x*x^m*x^(3*n)*e^m + 120*B*a*c*d^2*
n^4*x*x^m*x^(3*n)*e^m + 120*A*b*c*d^2*n^4*x*x^m*x^(3*n)*e^m + 40*A*a*d^3*n^
4*x*x^m*x^(3*n)*e^m + 5*B*b*c^3*m^4*x*x^m*x^(2*n)*e^m + 15*B*a*c^2*d*m^4*x*
x^m*x^(2*n)*e^m + 15*A*b*c^2*d*m^4*x*x^m*x^(2*n)*e^m + 15*A*a*c*d^2*m^4*x*x
^m*x^(2*n)*e^m + 52*B*b*c^3*m^3*n*x*x^m*x^(2*n)*e^m + 156*B*a*c^2*d*m^3*n*x
*x^m*x^(2*n)*e^m + 156*A*b*c^2*d*m^3*n*x*x^m*x^(2*n)*e^m + 156*A*a*c*d^2*m^
3*n*x*x^m*x^(2*n)*e^m + 177*B*b*c^3*m^2*n^2*x*x^m*x^(2*n)*e^m + 531*B*a*c^2
*d*m^2*n^2*x*x^m*x^(2*n)*e^m + 531*A*b*c^2*d*m^2*n^2*x*x^m*x^(2*n)*e^m + 53
1*A*a*c*d^2*m^2*n^2*x*x^m*x^(2*n)*e^m + 214*B*b*c^3*m*n^3*x*x^m*x^(2*n)*e^m
 + 642*B*a*c^2*d*m*n^3*x*x^m*x^(2*n)*e^m + 642*A*b*c^2*d*m*n^3*x*x^m*x^(2*n
)*e^m + 642*A*a*c*d^2*m*n^3*x*x^m*x^(2*n)*e^m + 60*B*b*c^3*n^4*x*x^m*x^(2*n
)*e^m + 180*B*a*c^2*d*n^4*x*x^m*x^(2*n)*e^m + 180*A*b*c^2*d*n^4*x*x^m*x^(2*
n)*e^m + 180*A*a*c*d^2*n^4*x*x^m*x^(2*n)*e^m + 5*B*a*c^3*m^4*x*x^m*x^n*e^m
 + 5*A*b*c^3*m^4*x*x^m*x^n*e^m + 15*A*a*c^2*d*m^4*x*x^m*x^n*e^m + 56*B*a*c^3
*m^3*n*x*x^m*x^n*e^m + 56*A*b*c^3*m^3*n*x*x^m*x^n*e^m + 168*A*a*c^2*d*m^3*n
*x*x^m*x^n*e^m + 213*B*a*c^3*m^2*n^2*x*x^m*x^n*e^m + 213*A*b*c^3*m^2*n^2*x*
x^m*x^n*e^m + 639*A*a*c^2*d*m^2*n^2*x*x^m*x^n*e^m + 308*B*a*c^3*m*n^3*x*x^m
*x^n*e^m + 308*A*b*c^3*m*n^3*x*x^m*x^n*e^m + 924*A*a*c^2*d*m*n^3*x*x^m*x^n*
e^m + 120*B*a*c^3*n^4*x*x^m*x^n*e^m + 120*A*b*c^3*n^4*x*x^m*x^n*e^m + 360*A
*a*c^2*d*n^4*x*x^m*x^n*e^m + 5*A*a*c^3*m^4*x*x^m*e^m + 60*A*a*c^3*m^3*n*x*x
^m*e^m + 255*A*a*c^3*m^2*n^2*x*x^m*e^m + 450*A*a*c^3*m*n^3*x*x^m*e^m + 274*
A*a*c^3*n^4*x*x^m*e^m + 10*B*b*d^3*m^3*x*x^m*x^(5*n)*e^m + 60*B*b*d^3*m^2*n
*x*x^m*x^(5*n)*e^m + 105*B*b*d^3*m*n^2*x*x^m*x^(5*n)*e^m + 50*B*b*d^3*n^3*x
*x^m*x^(5*n)*e^m + 30*B*b*c*d^2*m^3*x*x^m*x^(4*n)*e^m + 10*B*a*d^3*m^3*x*x^
```

m*x^(4*n)*e^m + 10*A*b*d^3*m^3*x*x^m*x^(4*n)*e^m + 198*B*b*c*d^2*m^2*n*x*x^
m*x^(4*n)*e^m + 66*B*a*d^3*m^2*n*x*x^m*x^(4*n)*e^m + 66*A*b*d^3*m^2*n*x*x^m
*x^(4*n)*e^m + 369*B*b*c*d^2*m*n^2*x*x^m*x^(4*n)*e^m + 123*B*a*d^3*m*n^2*x*
x^m*x^(4*n)*e^m + 123*A*b*d^3*m*n^2*x*x^m*x^(4*n)*e^m + 183*B*b*c*d^2*n^3*x
*x^m*x^(4*n)*e^m + 61*B*a*d^3*n^3*x*x^m*x^(4*n)*e^m + 61*A*b*d^3*n^3*x*x^m*
x^(4*n)*e^m + 30*B*b*c^2*d*m^3*x*x^m*x^(3*n)*e^m + 30*B*a*c*d^2*m^3*x*x^m*x
^(3*n)*e^m + 30*A*b*c*d^2*m^3*x*x^m*x^(3*n)*e^m + 10*A*a*d^3*m^3*x*x^m*x^(3
*n)*e^m + 216*B*b*c^2*d*m^2*n*x*x^m*x^(3*n)*e^m + 216*B*a*c*d^2*m^2*n*x*x^m
*x^(3*n)*e^m + 216*A*b*c*d^2*m^2*n*x*x^m*x^(3*n)*e^m + 72*A*a*d^3*m^2*n*x*x
^m*x^(3*n)*e^m + 441*B*b*c^2*d*m*n^2*x*x^m*x^(3*n)*e^m + 441*B*a*c*d^2*m*n^
2*x*x^m*x^(3*n)*e^m + 441*A*b*c*d^2*m*n^2*x*x^m*x^(3*n)*e^m + 147*A*a*d^3*m
*n^2*x*x^m*x^(3*n)*e^m + 234*B*b*c^2*d*n^3*x*x^m*x^(3*n)*e^m + 234*B*a*c*d^
2*n^3*x*x^m*x^(3*n)*e^m + 234*A*b*c*d^2*n^3*x*x^m*x^(3*n)*e^m + 78*A*a*d^3*
n^3*x*x^m*x^(3*n)*e^m + 10*B*b*c^3*m^3*x*x^m*x^(2*n)*e^m + 30*B*a*c^2*d*m^3
*x*x^m*x^(2*n)*e^m + 30*A*b*c^2*d*m^3*x*x^m*x^(2*n)*e^m + 30*A*a*c*d^2*m^3*
x*x^m*x^(2*n)*e^m + 78*B*b*c^3*m^2*n*x*x^m*x^(2*n)*e^m + 234*B*a*c^2*d*m^2*
n*x*x^m*x^(2*n)*e^m + 234*A*b*c^2*d*m^2*n*x*x^m*x^(2*n)*e^m + 234*A*a*c*d^2
*m^2*n*x*x^m*x^(2*n)*e^m + 177*B*b*c^3*m*n^2*x*x^m*x^(2*n)*e^m + 531*B*a*c^
2*d*m*n^2*x*x^m*x^(2*n)*e^m + 531*A*b*c^2*d*m*n^2*x*x^m*x^(2*n)*e^m + 531*A
*a*c*d^2*m*n^2*x*x^m*x^(2*n)*e^m + 107*B*b*c^3*n^3*x*x^m*x^(2*n)*e^m + 321*
B*a*c^2*d*n^3*x*x^m*x^(2*n)*e^m + 321*A*b*c^2*d*n^3*x*x^m*x^(2*n)*e^m + 321
*A*a*c*d^2*n^3*x*x^m*x^(2*n)*e^m + 10*B*a*c^3*m^3*x*x^m*x^n*e^m + 10*A*b*c^
3*m^3*x*x^m*x^n*e^m + 30*A*a*c^2*d*m^3*x*x^m*x^n*e^m + 84*B*a*c^3*m^2*n*x*x
^m*x^n*e^m + 84*A*b*c^3*m^2*n*x*x^m*x^n*e^m + 252*A*a*c^2*d*m^2*n*x*x^m*x^n
*e^m + 213*B*a*c^3*m*n^2*x*x^m*x^n*e^m + 213*A*b*c^3*m*n^2*x*x^m*x^n*e^m +
639*A*a*c^2*d*m*n^2*x*x^m*x^n*e^m + 154*B*a*c^3*n^3*x*x^m*x^n*e^m + 154*A*b
*c^3*n^3*x*x^m*x^n*e^m + 462*A*a*c^2*d*n^3*x*x^m*x^n*e^m + 10*A*a*c^3*m^3*x
*x^m*e^m + 90*A*a*c^3*m^2*n*x*x^m*e^m + 255*A*a*c^3*m*n^2*x*x^m*e^m + 225*A
*a*c^3*n^3*x*x^m*e^m + 10*B*b*d^3*m^2*x*x^m*x^(5*n)*e^m + 40*B*b*d^3*m*n*x*
x^m*x^(5*n)*e^m + 35*B*b*d^3*n^2*x*x^m*x^(5*n)*e^m + 30*B*b*c*d^2*m^2*x*x^m
*x^(4*n)*e^m + 10*B*a*d^3*m^2*x*x^m*x^(4*n)*e^m + 10*A*b*d^3*m^2*x*x^m*x^(4
*n)*e^m + 132*B*b*c*d^2*m*n*x*x^m*x^(4*n)*e^m + 44*B*a*d^3*m*n*x*x^m*x^(4*n
)*e^m + 44*A*b*d^3*m*n*x*x^m*x^(4*n)*e^m + 123*B*b*c*d^2*n^2*x*x^m*x^(4*n)*
e^m + 41*B*a*d^3*n^2*x*x^m*x^(4*n)*e^m + 41*A*b*d^3*n^2*x*x^m*x^(4*n)*e^m +
30*B*b*c^2*d*m^2*x*x^m*x^(3*n)*e^m + 30*B*a*c*d^2*m^2*x*x^m*x^(3*n)*e^m +
30*A*b*c*d^2*m^2*x*x^m*x^(3*n)*e^m + 10*A*a*d^3*m^2*x*x^m*x^(3*n)*e^m + 144
*B*b*c^2*d*m*n*x*x^m*x^(3*n)*e^m + 144*B*a*c*d^2*m*n*x*x^m*x^(3*n)*e^m + 14
4*A*b*c*d^2*m*n*x*x^m*x^(3*n)*e^m + 48*A*a*d^3*m*n*x*x^m*x^(3*n)*e^m + 147*
B*b*c^2*d*n^2*x*x^m*x^(3*n)*e^m + 147*B*a*c*d^2*n^2*x*x^m*x^(3*n)*e^m + 147
*A*b*c*d^2*n^2*x*x^m*x^(3*n)*e^m + 49*A*a*d^3*n^2*x*x^m*x^(3*n)*e^m + 10*B*
b*c^3*m^2*x*x^m*x^(2*n)*e^m + 30*B*a*c^2*d*m^2*x*x^m*x^(2*n)*e^m + 30*A*b*c
^2*d*m^2*x*x^m*x^(2*n)*e^m + 30*A*a*c*d^2*m^2*x*x^m*x^(2*n)*e^m + 52*B*b*c^
3*m*n*x*x^m*x^(2*n)*e^m + 156*B*a*c^2*d*m*n*x*x^m*x^(2*n)*e^m + 156*A*b*c^2
*d*m*n*x*x^m*x^(2*n)*e^m + 156*A*a*c*d^2*m*n*x*x^m*x^(2*n)*e^m + 59*B*b*c^3
*n^2*x*x^m*x^(2*n)*e^m + 177*B*a*c^2*d*n^2*x*x^m*x^(2*n)*e^m + 177*A*b*c^2*
d*n^2*x*x^m*x^(2*n)*e^m + 177*A*a*c*d^2*n^2*x*x^m*x^(2*n)*e^m + 10*B*a*c^3*
m^2*x*x^m*x^n*e^m + 10*A*b*c^3*m^2*x*x^m*x^n*e^m + 30*A*a*c^2*d*m^2*x*x^m*x
^n*e^m + 56*B*a*c^3*m*n*x*x^m*x^n*e^m + 56*A*b*c^3*m*n*x*x^m*x^n*e^m + 168*
A*a*c^2*d*m*n*x*x^m*x^n*e^m + 71*B*a*c^3*n^2*x*x^m*x^n*e^m + 71*A*b*c^3*n^2
*x*x^m*x^n*e^m + 213*A*a*c^2*d*n^2*x*x^m*x^n*e^m + 10*A*a*c^3*m^2*x*x^m*e^m
+ 60*A*a*c^3*m*n*x*x^m*e^m + 85*A*a*c^3*n^2*x*x^m*e^m + 5*B*b*d^3*m*x*x^m*
x^(5*n)*e^m + 10*B*b*d^3*n*x*x^m*x^(5*n)*e^m + 15*B*b*c*d^2*m*x*x^m*x^(4*n)
*e^m + 5*B*a*d^3*m*x*x^m*x^(4*n)*e^m + 5*A*b*d^3*m*x*x^m*x^(4*n)*e^m + 33*B
*b*c*d^2*n*x*x^m*x^(4*n)*e^m + 11*B*a*d^3*n*x*x^m*x^(4*n)*e^m + 11*A*b*d^3*
n*x*x^m*x^(4*n)*e^m + 15*B*b*c^2*d*m*x*x^m*x^(3*n)*e^m + 15*B*a*c*d^2*m*x*x
^m*x^(3*n)*e^m + 15*A*b*c*d^2*m*x*x^m*x^(3*n)*e^m + 5*A*a*d^3*m*x*x^m*x^(3*
n)*e^m + 36*B*b*c^2*d*n*x*x^m*x^(3*n)*e^m + 36*B*a*c*d^2*n*x*x^m*x^(3*n)*e^
m + 36*A*b*c*d^2*n*x*x^m*x^(3*n)*e^m + 12*A*a*d^3*n*x*x^m*x^(3*n)*e^m + 5*B
*b*c^3*m*x*x^m*x^(2*n)*e^m + 15*B*a*c^2*d*m*x*x^m*x^(2*n)*e^m + 15*A*b*c^2*

```
d*m*x*x^m*x^(2*n)*e^m + 15*A*a*c*d^2*m*x*x^m*x^(2*n)*e^m + 13*B*b*c^3*n*x*x
^m*x^(2*n)*e^m + 39*B*a*c^2*d*n*x*x^m*x^(2*n)*e^m + 39*A*b*c^2*d*n*x*x^m*x^
(2*n)*e^m + 39*A*a*c*d^2*n*x*x^m*x^(2*n)*e^m + 5*B*a*c^3*m*x*x^m*x^n*e^m +
5*A*b*c^3*m*x*x^m*x^n*e^m + 15*A*a*c^2*d*m*x*x^m*x^n*e^m + 14*B*a*c^3*n*x*x
^m*x^n*e^m + 14*A*b*c^3*n*x*x^m*x^n*e^m + 42*A*a*c^2*d*n*x*x^m*x^n*e^m + 5*
A*a*c^3*m*x*x^m*e^m + 15*A*a*c^3*n*x*x^m*e^m + B*b*d^3*x*x^m*x^(5*n)*e^m +
3*B*b*c*d^2*x*x^m*x^(4*n)*e^m + B*a*d^3*x*x^m*x^(4*n)*e^m + A*b*d^3*x*x^m*x
^(4*n)*e^m + 3*B*b*c^2*d*x*x^m*x^(3*n)*e^m + 3*B*a*c*d^2*x*x^m*x^(3*n)*e^m
+ 3*A*b*c*d^2*x*x^m*x^(3*n)*e^m + A*a*d^3*x*x^m*x^(3*n)*e^m + B*b*c^3*x*x^m
*x^(2*n)*e^m + 3*B*a*c^2*d*x*x^m*x^(2*n)*e^m + 3*A*b*c^2*d*x*x^m*x^(2*n)*e^
m + 3*A*a*c*d^2*x*x^m*x^(2*n)*e^m + B*a*c^3*x*x^m*x^n*e^m + A*b*c^3*x*x^m*x
^n*e^m + 3*A*a*c^2*d*x*x^m*x^n*e^m + A*a*c^3*x*x^m*e^m)/(m^6 + 15*m^5*n + 8
5*m^4*n^2 + 225*m^3*n^3 + 274*m^2*n^4 + 120*m*n^5 + 6*m^5 + 75*m^4*n + 340*
m^3*n^2 + 675*m^2*n^3 + 548*m*n^4 + 120*n^5 + 15*m^4 + 150*m^3*n + 510*m^2*
n^2 + 675*m*n^3 + 274*n^4 + 20*m^3 + 150*m^2*n + 340*m*n^2 + 225*n^3 + 15*m
^2 + 75*m*n + 85*n^2 + 6*m + 15*n + 1)
```

maple [C] time = 0.18, size = 4972, normalized size = 23.68

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(b*x^n+a)*(B*x^n+A)*(d*x^n+c)^3,x)
```

```
[Out] x*(44*B*a*d^3*m^3*n*(x^n)^4+123*B*a*d^3*m^2*n^2*(x^n)^4+122*B*a*d^3*m*n^3*(
x^n)^4+3*B*b*c^2*d*m^5*(x^n)^3+15*B*b*c*d^2*m^4*(x^n)^4+A*a*d^3*(x^n)^3+B*b
*c^3*(x^n)^2+A*b*c^3*x^n+B*a*c^3*x^n+b*B*d^3*(x^n)^5+A*b*d^3*(x^n)^4+B*a*d^
3*(x^n)^4+10*A*a*c^3*m^2+85*A*a*c^3*n^2+120*A*a*c^3*n^5+A*a*c^3*m^5+5*A*a*c
^3*m^4+274*A*a*c^3*n^4+10*A*a*c^3*m^3+225*A*a*c^3*n^3+a*A*c^3+5*a*A*c^3*m+1
5*a*A*c^3*n+40*A*a*d^3*m*n^4*(x^n)^3+3*A*b*c*d^2*m^5*(x^n)^3+44*A*b*d^3*m^3
*n*(x^n)^4+123*A*b*d^3*m^2*n^2*(x^n)^4+122*A*b*d^3*m*n^3*(x^n)^4+3*B*a*c*d^
2*m^5*(x^n)^3+84*B*a*c^3*m^2*n*x^n+213*B*a*c^3*m*n^2*x^n+30*B*a*c^2*d*m^2*(
x^n)^2+177*B*a*c^2*d*n^2*(x^n)^2+15*B*a*c*d^2*(x^n)^3*m+36*B*a*c*d^2*(x^n)^
3*n+52*B*b*c^3*m*n*(x^n)^2+15*B*b*c^2*d*(x^n)^3*m+36*B*b*c^2*d*(x^n)^3*n+30
*A*a*c^2*d*m^2*x^n+213*A*a*c^2*d*n^2*x^n+15*A*a*c*d^2*(x^n)^2*m+90*A*a*c^3*
m^2*n+66*B*a*d^3*m^2*n*(x^n)^4+123*B*a*d^3*m*n^2*(x^n)^4+13*B*b*c^3*m^4*n*(
x^n)^2+59*B*b*c^3*m^3*n^2*(x^n)^2+107*B*b*c^3*m^2*n^3*(x^n)^2+60*B*b*c^3*m*
n^4*(x^n)^2+15*B*b*c^2*d*m^4*(x^n)^3+120*B*b*c^2*d*n^4*(x^n)^3+30*B*b*c*d^2
*m^3*(x^n)^4+11*B*a*d^3*m^4*n*(x^n)^4+41*B*a*d^3*m^3*n^2*(x^n)^4+39*A*a*c*d
^2*(x^n)^2*n+56*A*b*c^3*m*n*x^n+15*A*b*c^2*d*(x^n)^2*m+39*A*b*c^2*d*(x^n)^2
*n+56*B*a*c^3*m*n*x^n+15*B*a*c^2*d*(x^n)^2*m+39*B*a*c^2*d*(x^n)^2*n+213*B*a
*c^3*m^2*n^2*x^n+308*B*a*c^3*m*n^3*x^n+30*B*a*c^2*d*m^3*(x^n)^2+321*B*a*c^2
*d*n^3*(x^n)^2+30*B*a*c*d^2*m^2*(x^n)^3+44*B*a*d^3*m*n*(x^n)^4+52*B*b*c^3*m
^3*n*(x^n)^2+177*B*b*c^3*m^2*n^2*(x^n)^2+214*B*b*c^3*m*n^3*(x^n)^2+30*B*b*c
^2*d*m^3*(x^n)^3+234*B*b*c^2*d*n^3*(x^n)^3+30*B*b*c*d^2*m^2*(x^n)^4+123*B*b
*c*d^2*n^2*(x^n)^4+15*A*a*c^2*d*m^4*x^n+360*A*a*c^2*d*n^4*x^n+30*A*a*c*d^2*
m^3*(x^n)^2+15*A*a*c^2*d*x^n*m+42*A*a*c^2*d*x^n*n+15*A*b*c*d^2*(x^n)^3*m+36
*A*b*c*d^2*(x^n)^3*n+3*A*b*c^2*d*m^5*(x^n)^2+15*A*b*c*d^2*m^4*(x^n)^3+120*A
*b*c*d^2*n^4*(x^n)^3+66*A*b*d^3*m^2*n*(x^n)^4+123*A*b*d^3*m*n^2*(x^n)^4+3*B
*a*c^2*d*m^5*(x^n)^2+15*B*a*c*d^2*m^4*(x^n)^3+120*B*a*c*d^2*n^4*(x^n)^3+90*
B*b*c*d^2*n^4*(x^n)^4+60*B*b*d^3*m^2*n*(x^n)^5+105*B*b*d^3*m*n^2*(x^n)^5+3*
A*a*c*d^2*m^5*(x^n)^2+48*A*a*d^3*m^3*n*(x^n)^3+10*B*b*d^3*m^4*n*(x^n)^5+35*
B*b*d^3*m^3*n^2*(x^n)^5+147*B*a*c*d^2*n^2*(x^n)^3+78*B*b*c^3*m^2*n*(x^n)^2+
177*B*b*c^3*m*n^2*(x^n)^2+30*B*b*c^2*d*m^2*(x^n)^3+147*B*b*c^2*d*n^2*(x^n)^
3+15*B*b*c*d^2*(x^n)^4*m+33*B*b*c*d^2*(x^n)^4*n+30*A*a*c^2*d*m^3*x^n+462*A*
a*c^2*d*n^3*x^n+321*A*a*c*d^2*n^3*(x^n)^2+48*A*a*d^3*m*n*(x^n)^3+56*A*b*c^3
*m^3*n*x^n+213*A*b*c^3*m^2*n^2*x^n+308*A*b*c^3*m*n^3*x^n+30*A*b*c^2*d*m^3*(
x^n)^2+321*A*b*c^2*d*n^3*(x^n)^2+30*A*b*c*d^2*m^2*(x^n)^3+255*A*a*c^3*m*n^2
+60*A*a*c^3*m*n+61*B*a*d^3*m^2*n^3*(x^n)^4+30*B*a*d^3*m*n^4*(x^n)^4+3*B*b*c
*d^2*m^5*(x^n)^4+40*B*b*d^3*m^3*n*(x^n)^5+105*B*b*d^3*m^2*n^2*(x^n)^5+100*B
```

```
*b*d^3*m*n^3*(x^n)^5+12*A*a*d^3*m^4*n*(x^n)^3+49*A*a*d^3*m^3*n^2*(x^n)^3+78
*A*a*d^3*m^2*n^3*(x^n)^3+50*B*b*d^3*m^2*n^3*(x^n)^5+24*B*b*d^3*m*n^4*(x^n)^
5+11*A*b*d^3*m^4*n*(x^n)^4+41*A*b*d^3*m^3*n^2*(x^n)^4+61*A*b*d^3*m^2*n^3*(x
^n)^4+30*A*a*c*d^2*m^2*(x^n)^2+177*A*a*c*d^2*n^2*(x^n)^2+84*A*b*c^3*m^2*n*x
^n+213*A*b*c^3*m*n^2*x^n+30*A*b*c^2*d*m^2*(x^n)^2+177*A*b*c^2*d*n^2*(x^n)^2
+15*B*a*c^2*d*m^4*(x^n)^2+180*B*a*c^2*d*n^4*(x^n)^2+30*B*a*c*d^2*m^3*(x^n)^
3+234*B*a*c*d^2*n^3*(x^n)^3+24*B*b*d^3*n^4*(x^n)^5+A*b*c^3*m^5*x^n+10*A*b*d
^3*m^2*(x^n)^4+41*A*b*d^3*n^2*(x^n)^4+B*a*c^3*m^5*x^n+10*B*a*d^3*m^2*(x^n)^
4+41*B*a*d^3*n^2*(x^n)^4+5*B*b*c^3*m^4*(x^n)^2+60*B*b*c^3*n^4*(x^n)^2+5*m*b
*B*d^3*(x^n)^5+147*A*b*c*d^2*n^2*(x^n)^3+56*B*a*c^3*m^3*n*x^n+183*B*b*c*d^2
*n^3*(x^n)^4+40*B*b*d^3*m*n*(x^n)^5+3*A*a*c^2*d*m^5*x^n+15*A*a*c*d^2*m^4*(x
^n)^2+180*A*a*c*d^2*n^4*(x^n)^2+72*A*a*d^3*m^2*n*(x^n)^3+147*A*a*d^3*m*n^2*
(x^n)^3+14*A*b*c^3*m^4*n*x^n+71*A*b*c^3*m^3*n^2*x^n+154*A*b*c^3*m^2*n^3*x^n
+120*A*b*c^3*m*n^4*x^n+15*A*b*c^2*d*m^4*(x^n)^2+180*A*b*c^2*d*n^4*(x^n)^2+3
0*A*b*c*d^2*m^3*(x^n)^3+234*A*b*c*d^2*n^3*(x^n)^3+44*A*b*d^3*m*n*(x^n)^4+14
*B*a*c^3*m^4*n*x^n+71*B*a*c^3*m^3*n^2*x^n+154*B*a*c^3*m^2*n^3*x^n+120*B*a*c
^3*m*n^4*x^n+147*A*a*d^3*m^2*n^2*(x^n)^3+156*A*a*d^3*m*n^3*(x^n)^3+50*B*b*d
^3*n^3*(x^n)^5+5*A*a*d^3*m^4*(x^n)^3+154*B*a*c^3*n^3*x^n+10*B*b*c^3*m^2*(x^
n)^2+59*B*b*c^3*n^2*(x^n)^2+10*A*b*c^3*m^2*x^n+71*A*b*c^3*n^2*x^n+10*B*a*c^
3*m^2*x^n+30*A*b*d^3*m*n^4*(x^n)^4+A*a*d^3*m^5*(x^n)^3+5*A*b*d^3*m^4*(x^n)^
4+30*A*b*d^3*n^4*(x^n)^4+5*B*a*d^3*m^4*(x^n)^4+30*B*a*d^3*n^4*(x^n)^4+10*B*
b*d^3*m^3*(x^n)^5+B*b*d^3*m^5*(x^n)^5+A*b*d^3*m^5*(x^n)^4+B*a*d^3*m^5*(x^n)
^4+5*B*b*d^3*m^4*(x^n)^5+5*A*a*d^3*(x^n)^3*m+12*A*a*d^3*(x^n)^3*n+10*A*b*c^
3*m^3*x^n+154*A*b*c^3*n^3*x^n+10*B*a*c^3*m^3*x^n+10*B*b*d^3*m^2*(x^n)^5+35*
B*b*d^3*n^2*(x^n)^5+10*A*a*d^3*m^3*(x^n)^3+78*A*a*d^3*n^3*(x^n)^3+40*A*a*d^
3*n^4*(x^n)^3+10*A*b*d^3*m^3*(x^n)^4+61*A*b*d^3*n^3*(x^n)^4+60*A*a*c^3*m^3*
n+255*A*a*c^3*m^2*n^2+450*A*a*c^3*m*n^3+15*A*a*c^3*m^4*n+85*A*a*c^3*m^3*n^2
+225*A*a*c^3*m^2*n^3+274*A*a*c^3*m*n^4+10*B*a*d^3*m^3*(x^n)^4+61*B*a*d^3*n^
3*(x^n)^4+B*b*c^3*m^5*(x^n)^2+3*(x^n)^2*d*c^2*A*b+3*(x^n)^2*d*c^2*B*a+3*(x^
n)^4*b*B*c*d^2+10*b*B*d^3*(x^n)^5*n+10*A*a*d^3*m^2*(x^n)^3+49*A*a*d^3*n^2*(
x^n)^3+5*A*b*c^3*m^4*x^n+120*A*b*c^3*n^4*x^n+5*A*b*d^3*(x^n)^4*m+11*A*b*d^3
*(x^n)^4*n+5*B*a*c^3*m^4*x^n+120*B*a*c^3*n^4*x^n+5*B*a*d^3*(x^n)^4*m+11*B*a
*d^3*(x^n)^4*n+10*B*b*c^3*m^3*(x^n)^2+107*B*b*c^3*n^3*(x^n)^2+3*x^n*a*A*c^2
*d+3*(x^n)^3*A*b*c*d^2+3*(x^n)^3*B*a*c*d^2+3*(x^n)^3*b*B*c^2*d+3*(x^n)^2*a*
A*c*d^2+71*B*a*c^3*n^2*x^n+5*B*b*c^3*(x^n)^2*m+13*B*b*c^3*(x^n)^2*n+5*A*b*c
^3*x^n*m+14*A*b*c^3*x^n*n+5*B*a*c^3*x^n*m+14*B*a*c^3*x^n*n+441*A*b*c*d^2*m^
2*n^2*(x^n)^3+39*A*b*c^2*d*m^4*n*(x^n)^2+177*A*b*c^2*d*m^3*n^2*(x^n)^2+321*
A*b*c^2*d*m^2*n^3*(x^n)^2+180*A*b*c^2*d*m*n^4*(x^n)^2+144*A*b*c*d^2*m^3*n*(
x^n)^3+639*A*a*c^2*d*m^2*n^2*x^n+924*A*a*c^2*d*m*n^3*x^n+234*A*a*c*d^2*m^2*
n*(x^n)^2+531*A*a*c*d^2*m*n^2*(x^n)^2+177*A*a*c*d^2*m^3*n^2*(x^n)^2+321*A*a
*c*d^2*m^2*n^3*(x^n)^2+180*A*a*c*d^2*m*n^4*(x^n)^2+144*B*a*c*d^2*m*n*(x^n)^
3+144*B*b*c^2*d*m*n*(x^n)^3+252*A*a*c^2*d*m^2*n*x^n+639*A*a*c^2*d*m*n^2*x^n
+441*B*b*c^2*d*m*n^2*(x^n)^3+132*B*b*c*d^2*m*n*(x^n)^4+168*A*a*c^2*d*m^3*n*
x^n+468*B*b*c^2*d*m*n^3*(x^n)^3+234*A*b*c^2*d*m^2*n*(x^n)^2+531*A*b*c^2*d*m
*n^2*(x^n)^2+144*A*b*c*d^2*m*n*(x^n)^3+234*B*a*c^2*d*m^2*n*(x^n)^2+531*B*a*
c^2*d*m*n^2*(x^n)^2+180*B*a*c^2*d*m*n^4*(x^n)^2+144*B*a*c*d^2*m^3*n*(x^n)^3
+441*B*a*c*d^2*m^2*n^2*(x^n)^3+468*B*a*c*d^2*m*n^3*(x^n)^3+144*B*b*c^2*d*m^
3*n*(x^n)^3+441*B*b*c^2*d*m^2*n^2*(x^n)^3+360*A*a*c^2*d*m*n^4*x^n+156*A*a*c
*d^2*m^3*n*(x^n)^2+468*A*b*c*d^2*m*n^3*(x^n)^3+39*B*a*c^2*d*m^4*n*(x^n)^2+1
77*B*a*c^2*d*m^3*n^2*(x^n)^2+321*B*a*c^2*d*m^2*n^3*(x^n)^2+366*B*b*c*d^2*m*
n^3*(x^n)^4+39*A*a*c*d^2*m^4*n*(x^n)^2+120*B*a*c*d^2*m*n^4*(x^n)^3+36*B*b*c
^2*d*m^4*n*(x^n)^3+147*B*b*c^2*d*m^3*n^2*(x^n)^3+234*B*b*c^2*d*m^2*n^3*(x^n
)^3+120*B*b*c^2*d*m*n^4*(x^n)^3+132*B*b*c*d^2*m^3*n*(x^n)^4+369*B*b*c*d^2*m
^2*n^2*(x^n)^4+168*A*a*c^2*d*m*n*x^n+147*A*b*c*d^2*m^3*n^2*(x^n)^3+234*A*b*
c*d^2*m^2*n^3*(x^n)^3+120*A*b*c*d^2*m*n^4*(x^n)^3+36*B*a*c*d^2*m^4*n*(x^n)^
3+147*B*a*c*d^2*m^3*n^2*(x^n)^3+234*B*a*c*d^2*m^2*n^3*(x^n)^3+216*B*a*c*d^2
*m^2*n*(x^n)^3+441*B*a*c*d^2*m*n^2*(x^n)^3+216*B*b*c^2*d*m^2*n*(x^n)^3+531*
A*a*c*d^2*m^2*n^2*(x^n)^2+156*A*a*c*d^2*m*n*(x^n)^2+156*A*b*c^2*d*m*n*(x^n)
^2+156*B*a*c^2*d*m*n*(x^n)^2+531*A*b*c^2*d*m^2*n^2*(x^n)^2+642*A*b*c^2*d*m*
```

```
n^3*(x^n)^2+216*A*b*c*d^2*m^2*n*(x^n)^3+441*A*b*c*d^2*m*n^2*(x^n)^3+156*B*a
*c^2*d*m^3*n*(x^n)^2+531*B*a*c^2*d*m^2*n^2*(x^n)^2+642*B*a*c^2*d*m*n^3*(x^n
)^2+198*B*b*c*d^2*m^2*n*(x^n)^4+369*B*b*c*d^2*m*n^2*(x^n)^4+42*A*a*c^2*d*m^
4*n*x^n+213*A*a*c^2*d*m^3*n^2*x^n+462*A*a*c^2*d*m^2*n^3*x^n+642*A*a*c*d^2*m
*n^3*(x^n)^2+156*A*b*c^2*d*m^3*n*(x^n)^2+33*B*b*c*d^2*m^4*n*(x^n)^4+123*B*b
*c*d^2*m^3*n^2*(x^n)^4+183*B*b*c*d^2*m^2*n^3*(x^n)^4+90*B*b*c*d^2*m*n^4*(x^
n)^4+36*A*b*c*d^2*m^4*n*(x^n)^3)/(m+1)/(m+n+1)/(m+2*n+1)/(m+3*n+1)/(m+4*n+1
)/(1+m+5*n)*exp(1/2*(-I*Pi*csgn(I*e)*csgn(I*x)*csgn(I*e*x)+I*Pi*csgn(I*e)*c
sgn(I*e*x)^2+I*Pi*csgn(I*x)*csgn(I*e*x)^2-I*Pi*csgn(I*e*x)^3+2*ln(e)+2*ln(x
))*m)
```

maxima [B] time = 1.01, size = 464, normalized size = 2.21

$$\frac{Bbd^3e^mxe^{\left(m\log(x)+5\,n\log(x)\right)}}{m+5\,n+1}+\frac{3\,Bbcd^2e^mxe^{\left(m\log(x)+4\,n\log(x)\right)}}{m+4\,n+1}+\frac{Bad^3e^mxe^{\left(m\log(x)+4\,n\log(x)\right)}}{m+4\,n+1}+\frac{Abd^3e^mxe^{\left(m\log(x)+4\,n\log(x)\right)}}{m+4\,n+1}+\frac{3\,Bbc^2de^mxe^{\left(m\log(x)+3\,n\log(x)\right)}}{m+3\,n+1}+\frac{3\,Bacd^2e^mxe^{\left(m\log(x)+3\,n\log(x)\right)}}{m+3\,n+1}+\frac{3\,Abcd^2e^mxe^{\left(m\log(x)+3\,n\log(x)\right)}}{m+3\,n+1}+\frac{Aad^3e^mxe^{\left(m\log(x)+3\,n\log(x)\right)}}{m+3\,n+1}+\frac{Bbc^3e^mxe^{\left(m\log(x)+2\,n\log(x)\right)}}{m+2\,n+1}+\frac{3\,Bac^2de^mxe^{\left(m\log(x)+2\,n\log(x)\right)}}{m+2\,n+1}+\frac{3\,Abc^2de^mxe^{\left(m\log(x)+2\,n\log(x)\right)}}{m+2\,n+1}+\frac{3\,Aacd^2e^mxe^{\left(m\log(x)+2\,n\log(x)\right)}}{m+2\,n+1}+\frac{Bac^3e^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{Abc^3e^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{3\,Aac^2de^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{(ex)^{m+1}Aac^3}{e(m+1)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(a+b*x^n)*(A+B*x^n)*(c+d*x^n)^3,x, algorithm="maxima")
```

```
[Out] B*b*d^3*e^m*x*e^(m*log(x) + 5*n*log(x))/(m + 5*n + 1) + 3*B*b*c*d^2*e^m*x*e
^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) + B*a*d^3*e^m*x*e^(m*log(x) + 4*n*lo
g(x))/(m + 4*n + 1) + A*b*d^3*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1)
 + 3*B*b*c^2*d*e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + 3*B*a*c*d^2*
e^m*x*e^(m*log(x) + 3*n*log(x))/(m + 3*n + 1) + 3*A*b*c*d^2*e^m*x*e^(m*log(
x) + 3*n*log(x))/(m + 3*n + 1) + A*a*d^3*e^m*x*e^(m*log(x) + 3*n*log(x))/(m
 + 3*n + 1) + B*b*c^3*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 3*B*a
*c^2*d*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 3*A*b*c^2*d*e^m*x*e^
(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 3*A*a*c*d^2*e^m*x*e^(m*log(x) + 2*n
*log(x))/(m + 2*n + 1) + B*a*c^3*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1)
+ A*b*c^3*e^m*x*e^(m*log(x) + n*log(x))/(m + n + 1) + 3*A*a*c^2*d*e^m*x*e^(
m*log(x) + n*log(x))/(m + n + 1) + (e*x)^(m + 1)*A*a*c^3/(e*(m + 1))
```

mupad [B] time = 5.65, size = 1089, normalized size = 5.19

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(A + B*x^n)*(a + b*x^n)*(c + d*x^n)^3,x)
```

```
[Out] (A*a*c^3*x*(e*x)^m)/(m + 1) + (d^2*x*x^(4*n)*(e*x)^m*(A*b*d + B*a*d + 3*B*b
*c)*(4*m + 11*n + 33*m*n + 82*m*n^2 + 33*m^2*n + 61*m*n^3 + 11*m^3*n + 6*m^
2 + 4*m^3 + m^4 + 41*n^2 + 61*n^3 + 30*n^4 + 41*m^2*n^2 + 1))/(5*m + 15*n +
 60*m*n + 255*m*n^2 + 90*m^2*n + 450*m*n^3 + 60*m^3*n + 274*m*n^4 + 15*m^4*
n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 85*n^2 + 225*n^3 + 274*n^4 + 120*n^5 +
255*m^2*n^2 + 225*m^2*n^3 + 85*m^3*n^2 + 1) + (c*x*x^(2*n)*(e*x)^m*(3*A*a*d
^2 + B*b*c^2 + 3*A*b*c*d + 3*B*a*c*d)*(4*m + 13*n + 39*m*n + 118*m*n^2 + 39
*m^2*n + 107*m*n^3 + 13*m^3*n + 6*m^2 + 4*m^3 + m^4 + 59*n^2 + 107*n^3 + 60
*n^4 + 59*m^2*n^2 + 1))/(5*m + 15*n + 60*m*n + 255*m*n^2 + 90*m^2*n + 450*m
*n^3 + 60*m^3*n + 274*m*n^4 + 15*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 85
*n^2 + 225*n^3 + 274*n^4 + 120*n^5 + 255*m^2*n^2 + 225*m^2*n^3 + 85*m^3*n^2
 + 1) + (d*x*x^(3*n)*(e*x)^m*(A*a*d^2 + 3*B*b*c^2 + 3*A*b*c*d + 3*B*a*c*d)*
(4*m + 12*n + 36*m*n + 98*m*n^2 + 36*m^2*n + 78*m*n^3 + 12*m^3*n + 6*m^2 +
4*m^3 + m^4 + 49*n^2 + 78*n^3 + 40*n^4 + 49*m^2*n^2 + 1))/(5*m + 15*n + 60*
m*n + 255*m*n^2 + 90*m^2*n + 450*m*n^3 + 60*m^3*n + 274*m*n^4 + 15*m^4*n +
10*m^2 + 10*m^3 + 5*m^4 + m^5 + 85*n^2 + 225*n^3 + 274*n^4 + 120*n^5 + 255*
m^2*n^2 + 225*m^2*n^3 + 85*m^3*n^2 + 1) + (c^2*x*x^n*(e*x)^m*(3*A*a*d + A*b
*c + B*a*c)*(4*m + 14*n + 42*m*n + 142*m*n^2 + 42*m^2*n + 154*m*n^3 + 14*m^
3*n + 6*m^2 + 4*m^3 + m^4 + 71*n^2 + 154*n^3 + 120*n^4 + 71*m^2*n^2 + 1))/(
5*m + 15*n + 60*m*n + 255*m*n^2 + 90*m^2*n + 450*m*n^3 + 60*m^3*n + 274*m*n
^4 + 15*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 85*n^2 + 225*n^3 + 274*n^4
```

```
+ 120*n^5 + 255*m^2*n^2 + 225*m^2*n^3 + 85*m^3*n^2 + 1) + (B*b*d^3*x*x^(5*n
)*(e*x)^m*(4*m + 10*n + 30*m*n + 70*m*n^2 + 30*m^2*n + 50*m*n^3 + 10*m^3*n
+ 6*m^2 + 4*m^3 + m^4 + 35*n^2 + 50*n^3 + 24*n^4 + 35*m^2*n^2 + 1))/(5*m +
15*n + 60*m*n + 255*m*n^2 + 90*m^2*n + 450*m*n^3 + 60*m^3*n + 274*m*n^4 + 1
5*m^4*n + 10*m^2 + 10*m^3 + 5*m^4 + m^5 + 85*n^2 + 225*n^3 + 274*n^4 + 120*
n^5 + 255*m^2*n^2 + 225*m^2*n^3 + 85*m^3*n^2 + 1)
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)**m*(a+b*x**n)*(A+B*x**n)*(c+d*x**n)**3,x)
```

```
[Out] Timed out
```

3.12 $\int (ex)^m (A+Bx^n) (c+dx^n)^3 \, dx$

Optimal. Leaf size=137

$$\frac{c^2x^{n+1}(ex)^m(3Ad+Bc)}{m+n+1}+\frac{d^2x^{3n+1}(ex)^m(Ad+3Bc)}{m+3n+1}+\frac{3cdx^{2n+1}(ex)^m(Ad+Bc)}{m+2n+1}+\frac{Ac^3(ex)^{m+1}}{e(m+1)}+\frac{Bd^3x^{4n+1}(ex)^m}{m+4n+1}$$

Rubi [A] time = 0.11, antiderivative size = 137, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 3, integrand size = 22, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.136, Rules used = {448, 20, 30}

$$\frac{c^2x^{n+1}(ex)^m(3Ad+Bc)}{m+n+1}+\frac{d^2x^{3n+1}(ex)^m(Ad+3Bc)}{m+3n+1}+\frac{3cdx^{2n+1}(ex)^m(Ad+Bc)}{m+2n+1}+\frac{Ac^3(ex)^{m+1}}{e(m+1)}+\frac{Bd^3x^{4n+1}(ex)^m}{m+4n+1}$$

Antiderivative was successfully verified.

```
[In] Int[(e*x)^m*(A + B*x^n)*(c + d*x^n)^3,x]
```

```
[Out] (c^2*(B*c + 3*A*d)*x^(1 + n)*(e*x)^m)/(1 + m + n) + (3*c*d*(B*c + A*d)*x^(1
 + 2*n)*(e*x)^m)/(1 + m + 2*n) + (d^2*(3*B*c + A*d)*x^(1 + 3*n)*(e*x)^m)/(1
 + m + 3*n) + (B*d^3*x^(1 + 4*n)*(e*x)^m)/(1 + m + 4*n) + (A*c^3*(e*x)^(1 +
 m))/(e*(1 + m))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 30

```
Int[(x_)^(m_.), x_Symbol] :> Simp[x^(m + 1)/(m + 1), x] /; FreeQ[m, x] && N
eQ[m, -1]
```

Rule 448

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

Rubi steps

$$\begin{aligned}\int (ex)^m (A+Bx^n)(c+dx^n)^3 \, dx &= \int \left(Ac^3(ex)^m + c^2(Bc+3Ad)x^n(ex)^m + 3cd(Bc+Ad)x^{2n}(ex)^m + d^2(3Bc+\right.\\ &= \frac{Ac^3(ex)^{1+m}}{e(1+m)} + \left(Bd^3\right)\int x^{4n}(ex)^m \, dx + (3cd(Bc+Ad))\int x^{2n}(ex)^m \, dx + \left(d^2\right.\\ &= \frac{Ac^3(ex)^{1+m}}{e(1+m)} + \left(Bd^3x^{-m}(ex)^m\right)\int x^{m+4n} \, dx + \left(3cd(Bc+Ad)x^{-m}(ex)^m\right)\int x^{\cdots}\\ &= \frac{c^2(Bc+3Ad)x^{1+n}(ex)^m}{1+m+n} + \frac{3cd(Bc+Ad)x^{1+2n}(ex)^m}{1+m+2n} + \frac{d^2(3Bc+Ad)x^{1+3n}(\cdots}{1+m+3n}\end{aligned}$$

Mathematica [A] time = 0.18, size = 106, normalized size = 0.77

$$x(ex)^m\left(\frac{c^2x^n(3Ad+Bc)}{m+n+1}+\frac{d^2x^{3n}(Ad+3Bc)}{m+3n+1}+\frac{3cdx^{2n}(Ad+Bc)}{m+2n+1}+\frac{Ac^3}{m+1}+\frac{Bd^3x^{4n}}{m+4n+1}\right)$$

Antiderivative was successfully verified.

```
[In] Integrate[(e*x)^m*(A + B*x^n)*(c + d*x^n)^3,x]
```

```
[Out] x*(e*x)^m*((A*c^3)/(1 + m) + (c^2*(B*c + 3*A*d)*x^n)/(1 + m + n) + (3*c*d*(
B*c + A*d)*x^(2*n))/(1 + m + 2*n) + (d^2*(3*B*c + A*d)*x^(3*n))/(1 + m + 3*
n) + (B*d^3*x^(4*n))/(1 + m + 4*n))
```

IntegrateAlgebraic [F] time = 0.11, size = 0, normalized size = 0.00

$$\int (ex)^m (A + Bx^n) (c + dx^n)^3 \, dx$$

Verification is not applicable to the result.

```
[In] IntegrateAlgebraic[(e*x)^m*(A + B*x^n)*(c + d*x^n)^3,x]
```

```
[Out] Defer[IntegrateAlgebraic][(e*x)^m*(A + B*x^n)*(c + d*x^n)^3, x]
```

fricas [B] time = 0.47, size = 1104, normalized size = 8.06

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(A+B*x^n)*(c+d*x^n)^3,x, algorithm="fricas")
```

```
[Out] ((B*d^3*m^4 + 4*B*d^3*m^3 + 6*B*d^3*m^2 + 4*B*d^3*m + B*d^3 + 6*(B*d^3*m +
B*d^3)*n^3 + 11*(B*d^3*m^2 + 2*B*d^3*m + B*d^3)*n^2 + 6*(B*d^3*m^3 + 3*B*d^
3*m^2 + 3*B*d^3*m + B*d^3)*n)*x*x^(4*n)*e^(m*log(e) + m*log(x)) + ((3*B*c*d
^2 + A*d^3)*m^4 + 3*B*c*d^2 + A*d^3 + 4*(3*B*c*d^2 + A*d^3)*m^3 + 8*(3*B*c*
d^2 + A*d^3 + (3*B*c*d^2 + A*d^3)*m)*n^3 + 6*(3*B*c*d^2 + A*d^3)*m^2 + 14*(
3*B*c*d^2 + A*d^3 + (3*B*c*d^2 + A*d^3)*m^2 + 2*(3*B*c*d^2 + A*d^3)*m)*n^2
+ 4*(3*B*c*d^2 + A*d^3)*m + 7*(3*B*c*d^2 + A*d^3 + (3*B*c*d^2 + A*d^3)*m^3
+ 3*(3*B*c*d^2 + A*d^3)*m^2 + 3*(3*B*c*d^2 + A*d^3)*m)*n)*x*x^(3*n)*e^(m*lo
g(e) + m*log(x)) + 3*((B*c^2*d + A*c*d^2)*m^4 + B*c^2*d + A*c*d^2 + 4*(B*c^
2*d + A*c*d^2)*m^3 + 12*(B*c^2*d + A*c*d^2 + (B*c^2*d + A*c*d^2)*m)*n^3 + 6
*(B*c^2*d + A*c*d^2)*m^2 + 19*(B*c^2*d + A*c*d^2 + (B*c^2*d + A*c*d^2)*m^2
+ 2*(B*c^2*d + A*c*d^2)*m)*n^2 + 4*(B*c^2*d + A*c*d^2)*m + 8*(B*c^2*d + A*c
*d^2 + (B*c^2*d + A*c*d^2)*m^3 + 3*(B*c^2*d + A*c*d^2)*m^2 + 3*(B*c^2*d + A
*c*d^2)*m)*n)*x*x^(2*n)*e^(m*log(e) + m*log(x)) + ((B*c^3 + 3*A*c^2*d)*m^4
+ B*c^3 + 3*A*c^2*d + 4*(B*c^3 + 3*A*c^2*d)*m^3 + 24*(B*c^3 + 3*A*c^2*d + (
B*c^3 + 3*A*c^2*d)*m)*n^3 + 6*(B*c^3 + 3*A*c^2*d)*m^2 + 26*(B*c^3 + 3*A*c^2
*d + (B*c^3 + 3*A*c^2*d)*m^2 + 2*(B*c^3 + 3*A*c^2*d)*m)*n^2 + 4*(B*c^3 + 3*
A*c^2*d)*m + 9*(B*c^3 + 3*A*c^2*d + (B*c^3 + 3*A*c^2*d)*m^3 + 3*(B*c^3 + 3*
A*c^2*d)*m^2 + 3*(B*c^3 + 3*A*c^2*d)*m)*n)*x*x^n*e^(m*log(e) + m*log(x)) +
(A*c^3*m^4 + 24*A*c^3*n^4 + 4*A*c^3*m^3 + 6*A*c^3*m^2 + 4*A*c^3*m + A*c^3 +
 50*(A*c^3*m + A*c^3)*n^3 + 35*(A*c^3*m^2 + 2*A*c^3*m + A*c^3)*n^2 + 10*(A*
c^3*m^3 + 3*A*c^3*m^2 + 3*A*c^3*m + A*c^3)*n)*x*e^(m*log(e) + m*log(x)))/(m
^5 + 24*(m + 1)*n^4 + 5*m^4 + 50*(m^2 + 2*m + 1)*n^3 + 10*m^3 + 35*(m^3 + 3
*m^2 + 3*m + 1)*n^2 + 10*m^2 + 10*(m^4 + 4*m^3 + 6*m^2 + 4*m + 1)*n + 5*m +
 1)
```

giac [B] time = 0.75, size = 2278, normalized size = 16.63

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(A+B*x^n)*(c+d*x^n)^3,x, algorithm="giac")
```

```
[Out] (B*d^3*m^4*x*x^m*x^(4*n)*e^m + 6*B*d^3*m^3*n*x*x^m*x^(4*n)*e^m + 11*B*d^3*m
^2*n^2*x*x^m*x^(4*n)*e^m + 6*B*d^3*m*n^3*x*x^m*x^(4*n)*e^m + 3*B*c*d^2*m^4*
x*x^m*x^(3*n)*e^m + A*d^3*m^4*x*x^m*x^(3*n)*e^m + 21*B*c*d^2*m^3*n*x*x^m*x^
(3*n)*e^m + 7*A*d^3*m^3*n*x*x^m*x^(3*n)*e^m + 42*B*c*d^2*m^2*n^2*x*x^m*x^(3
```

```
*n)*e^m + 14*A*d^3*m^2*n^2*x*x^m*x^(3*n)*e^m + 24*B*c*d^2*m*n^3*x*x^m*x^(3*
n)*e^m + 8*A*d^3*m*n^3*x*x^m*x^(3*n)*e^m + 3*B*c^2*d*m^4*x*x^m*x^(2*n)*e^m
+ 3*A*c*d^2*m^4*x*x^m*x^(2*n)*e^m + 24*B*c^2*d*m^3*n*x*x^m*x^(2*n)*e^m + 24
*A*c*d^2*m^3*n*x*x^m*x^(2*n)*e^m + 57*B*c^2*d*m^2*n^2*x*x^m*x^(2*n)*e^m + 5
7*A*c*d^2*m^2*n^2*x*x^m*x^(2*n)*e^m + 36*B*c^2*d*m*n^3*x*x^m*x^(2*n)*e^m +
36*A*c*d^2*m*n^3*x*x^m*x^(2*n)*e^m + B*c^3*m^4*x*x^m*x^n*e^m + 3*A*c^2*d*m^
4*x*x^m*x^n*e^m + 9*B*c^3*m^3*n*x*x^m*x^n*e^m + 27*A*c^2*d*m^3*n*x*x^m*x^n*
e^m + 26*B*c^3*m^2*n^2*x*x^m*x^n*e^m + 78*A*c^2*d*m^2*n^2*x*x^m*x^n*e^m + 2
4*B*c^3*m*n^3*x*x^m*x^n*e^m + 72*A*c^2*d*m*n^3*x*x^m*x^n*e^m + A*c^3*m^4*x*
x^m*e^m + 10*A*c^3*m^3*n*x*x^m*e^m + 35*A*c^3*m^2*n^2*x*x^m*e^m + 50*A*c^3*
m*n^3*x*x^m*e^m + 24*A*c^3*n^4*x*x^m*e^m + 4*B*d^3*m^3*x*x^m*x^(4*n)*e^m +
18*B*d^3*m^2*n*x*x^m*x^(4*n)*e^m + 22*B*d^3*m*n^2*x*x^m*x^(4*n)*e^m + 6*B*d
^3*n^3*x*x^m*x^(4*n)*e^m + 12*B*c*d^2*m^3*x*x^m*x^(3*n)*e^m + 4*A*d^3*m^3*x
*x^m*x^(3*n)*e^m + 63*B*c*d^2*m^2*n*x*x^m*x^(3*n)*e^m + 21*A*d^3*m^2*n*x*x^
m*x^(3*n)*e^m + 84*B*c*d^2*m*n^2*x*x^m*x^(3*n)*e^m + 28*A*d^3*m*n^2*x*x^m*x
^(3*n)*e^m + 24*B*c*d^2*n^3*x*x^m*x^(3*n)*e^m + 8*A*d^3*n^3*x*x^m*x^(3*n)*e
^m + 12*B*c^2*d*m^3*x*x^m*x^(2*n)*e^m + 12*A*c*d^2*m^3*x*x^m*x^(2*n)*e^m +
72*B*c^2*d*m^2*n*x*x^m*x^(2*n)*e^m + 72*A*c*d^2*m^2*n*x*x^m*x^(2*n)*e^m + 1
14*B*c^2*d*m*n^2*x*x^m*x^(2*n)*e^m + 114*A*c*d^2*m*n^2*x*x^m*x^(2*n)*e^m +
36*B*c^2*d*n^3*x*x^m*x^(2*n)*e^m + 36*A*c*d^2*n^3*x*x^m*x^(2*n)*e^m + 4*B*c
^3*m^3*x*x^m*x^n*e^m + 12*A*c^2*d*m^3*x*x^m*x^n*e^m + 27*B*c^3*m^2*n*x*x^m*
x^n*e^m + 81*A*c^2*d*m^2*n*x*x^m*x^n*e^m + 52*B*c^3*m*n^2*x*x^m*x^n*e^m + 1
56*A*c^2*d*m*n^2*x*x^m*x^n*e^m + 24*B*c^3*n^3*x*x^m*x^n*e^m + 72*A*c^2*d*n^
3*x*x^m*x^n*e^m + 4*A*c^3*m^3*x*x^m*e^m + 30*A*c^3*m^2*n*x*x^m*e^m + 70*A*c
^3*m*n^2*x*x^m*e^m + 50*A*c^3*n^3*x*x^m*e^m + 6*B*d^3*m^2*x*x^m*x^(4*n)*e^m
 + 18*B*d^3*m*n*x*x^m*x^(4*n)*e^m + 11*B*d^3*n^2*x*x^m*x^(4*n)*e^m + 18*B*c
*d^2*m^2*x*x^m*x^(3*n)*e^m + 6*A*d^3*m^2*x*x^m*x^(3*n)*e^m + 63*B*c*d^2*m*n
*x*x^m*x^(3*n)*e^m + 21*A*d^3*m*n*x*x^m*x^(3*n)*e^m + 42*B*c*d^2*n^2*x*x^m*
x^(3*n)*e^m + 14*A*d^3*n^2*x*x^m*x^(3*n)*e^m + 18*B*c^2*d*m^2*x*x^m*x^(2*n)
*e^m + 18*A*c*d^2*m^2*x*x^m*x^(2*n)*e^m + 72*B*c^2*d*m*n*x*x^m*x^(2*n)*e^m
+ 72*A*c*d^2*m*n*x*x^m*x^(2*n)*e^m + 57*B*c^2*d*n^2*x*x^m*x^(2*n)*e^m + 57*
A*c*d^2*n^2*x*x^m*x^(2*n)*e^m + 6*B*c^3*m^2*x*x^m*x^n*e^m + 18*A*c^2*d*m^2*
x*x^m*x^n*e^m + 27*B*c^3*m*n*x*x^m*x^n*e^m + 81*A*c^2*d*m*n*x*x^m*x^n*e^m +
 26*B*c^3*n^2*x*x^m*x^n*e^m + 78*A*c^2*d*n^2*x*x^m*x^n*e^m + 6*A*c^3*m^2*x*
x^m*e^m + 30*A*c^3*m*n*x*x^m*e^m + 35*A*c^3*n^2*x*x^m*e^m + 4*B*d^3*m*x*x^m
*x^(4*n)*e^m + 6*B*d^3*n*x*x^m*x^(4*n)*e^m + 12*B*c*d^2*m*x*x^m*x^(3*n)*e^m
 + 4*A*d^3*m*x*x^m*x^(3*n)*e^m + 21*B*c*d^2*n*x*x^m*x^(3*n)*e^m + 7*A*d^3*n
*x*x^m*x^(3*n)*e^m + 12*B*c^2*d*m*x*x^m*x^(2*n)*e^m + 12*A*c*d^2*m*x*x^m*x^
(2*n)*e^m + 24*B*c^2*d*n*x*x^m*x^(2*n)*e^m + 24*A*c*d^2*n*x*x^m*x^(2*n)*e^m
 + 4*B*c^3*m*x*x^m*x^n*e^m + 12*A*c^2*d*m*x*x^m*x^n*e^m + 9*B*c^3*n*x*x^m*x
^n*e^m + 27*A*c^2*d*n*x*x^m*x^n*e^m + 4*A*c^3*m*x*x^m*e^m + 10*A*c^3*n*x*x^
m*e^m + B*d^3*x*x^m*x^(4*n)*e^m + 3*B*c*d^2*x*x^m*x^(3*n)*e^m + A*d^3*x*x^m
*x^(3*n)*e^m + 3*B*c^2*d*x*x^m*x^(2*n)*e^m + 3*A*c*d^2*x*x^m*x^(2*n)*e^m +
B*c^3*x*x^m*x^n*e^m + 3*A*c^2*d*x*x^m*x^n*e^m + A*c^3*x*x^m*e^m)/(m^5 + 10*
m^4*n + 35*m^3*n^2 + 50*m^2*n^3 + 24*m*n^4 + 5*m^4 + 40*m^3*n + 105*m^2*n^2
 + 100*m*n^3 + 24*n^4 + 10*m^3 + 60*m^2*n + 105*m*n^2 + 50*n^3 + 10*m^2 + 4
0*m*n + 35*n^2 + 5*m + 10*n + 1)
```

maple [C] time = 0.13, size = 1609, normalized size = 11.74

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(B*x^n+A)*(d*x^n+c)^3,x)
```

```
[Out] x*(57*B*c^2*d*m^2*n^2*(x^n)^2+36*B*c^2*d*m*n^3*(x^n)^2+24*B*c^2*d*m^3*n*(x^
n)^2+11*B*d^3*n^2*(x^n)^4+B*d^3*m^4*(x^n)^4+8*A*d^3*n^3*(x^n)^3+6*B*d^3*m^2
*(x^n)^4+24*B*c^3*n^3*x^n+A*d^3*m^4*(x^n)^3+4*B*d^3*m^3*(x^n)^4+6*B*d^3*n^3
*(x^n)^4+4*A*d^3*m^3*(x^n)^3+4*B*c^3*m^3*x^n+6*B*d^3*(x^n)^4*n+4*A*d^3*(x^n
)^3*m+7*A*d^3*(x^n)^3*n+B*c^3*m^4*x^n+4*m*B*d^3*(x^n)^4+26*B*c^3*n^2*x^n+6*
```

```
A*d^3*m^2*(x^n)^3+14*A*d^3*n^2*(x^n)^3+3*A*c^2*d*x^n+3*B*c*d^2*(x^n)^3+3*A*
c*d^2*(x^n)^2+4*B*c^3*x^n*m+9*B*c^3*x^n*n+6*B*c^3*m^2*x^n+3*B*c^2*d*(x^n)^2
+10*A*c^3*m^3*n+35*A*c^3*m^2*n^2+50*A*c^3*m*n^3+30*A*c^3*m^2*n+70*A*c^3*m*n
^2+30*A*c^3*m*n+36*A*c*d^2*m*n^3*(x^n)^2+24*A*c*d^2*m^3*n*(x^n)^2+57*A*c*d^
2*m^2*n^2*(x^n)^2+72*B*c^2*d*m^2*n*(x^n)^2+114*B*c^2*d*m*n^2*(x^n)^2+63*B*c
*d^2*m*n*(x^n)^3+21*B*c*d^2*m^3*n*(x^n)^3+42*B*c*d^2*m^2*n^2*(x^n)^3+24*B*c
*d^2*m*n^3*(x^n)^3+84*B*c*d^2*m*n^2*(x^n)^3+27*A*c^2*d*m^3*n*x^n+78*A*c^2*d
*m^2*n^2*x^n+72*A*c^2*d*m*n^3*x^n+72*A*c*d^2*m^2*n*(x^n)^2+114*A*c*d^2*m*n^
2*(x^n)^2+81*A*c^2*d*m^2*n*x^n+156*A*c^2*d*m*n^2*x^n+72*A*c*d^2*m*n*(x^n)^2
+72*B*c^2*d*m*n*(x^n)^2+63*B*c*d^2*m^2*n*(x^n)^3+81*A*c^2*d*m*n*x^n+A*c^3+(
x^n)^4*B*d^3+(x^n)^3*A*d^3+12*B*c^2*d*m^3*(x^n)^2+36*B*c^2*d*n^3*(x^n)^2+4*
A*c^3*m+10*A*c^3*n+24*A*c^3*n^4+4*A*c^3*m^3+50*A*c^3*n^3+6*A*c^3*m^2+35*A*c
^3*n^2+x^n*B*c^3+A*c^3*m^4+72*A*c^2*d*n^3*x^n+18*A*c*d^2*m^2*(x^n)^2+57*A*c
*d^2*n^2*(x^n)^2+27*B*c^3*m^2*n*x^n+18*B*c*d^2*m^2*(x^n)^3+3*A*c*d^2*m^4*(x
^n)^2+21*A*d^3*m^2*n*(x^n)^3+28*A*d^3*m*n^2*(x^n)^3+3*B*c^2*d*m^4*(x^n)^2+2
6*B*c^3*m^2*n^2*x^n+7*A*d^3*m^3*n*(x^n)^3+14*A*d^3*m^2*n^2*(x^n)^3+8*A*d^3*
m*n^3*(x^n)^3+3*B*c*d^2*m^4*(x^n)^3+18*B*d^3*m^2*n*(x^n)^4+22*B*d^3*m*n^2*(
x^n)^4+78*A*c^2*d*n^2*x^n+12*A*c*d^2*(x^n)^2*m+24*A*c*d^2*(x^n)^2*n+27*B*c^
3*m*n*x^n+12*B*c^2*d*(x^n)^2*m+24*B*c^2*d*(x^n)^2*n+12*A*c^2*d*x^n*m+27*A*c
^2*d*x^n*n+42*B*c*d^2*n^2*(x^n)^3+12*A*c^2*d*m^3*x^n+18*B*d^3*m*n*(x^n)^4+3
*A*c^2*d*m^4*x^n+12*A*c*d^2*m^3*(x^n)^2+36*A*c*d^2*n^3*(x^n)^2+21*A*d^3*m*n
*(x^n)^3+9*B*c^3*m^3*n*x^n+24*B*c^3*m*n^3*x^n+12*B*c*d^2*m^3*(x^n)^3+24*B*c
*d^2*n^3*(x^n)^3+6*B*d^3*m^3*n*(x^n)^4+11*B*d^3*m^2*n^2*(x^n)^4+6*B*d^3*m*n
^3*(x^n)^4+52*B*c^3*m*n^2*x^n+18*B*c^2*d*m^2*(x^n)^2+57*B*c^2*d*n^2*(x^n)^2
+12*B*c*d^2*(x^n)^3*m+21*B*c*d^2*(x^n)^3*n+18*A*c^2*d*m^2*x^n)/(m+1)/(m+n+1
)/(m+2*n+1)/(m+3*n+1)/(m+4*n+1)*exp(1/2*(-I*Pi*csgn(I*e)*csgn(I*x)*csgn(I*e
*x)+I*Pi*csgn(I*e)*csgn(I*e*x)^2+I*Pi*csgn(I*x)*csgn(I*e*x)^2-I*Pi*csgn(I*e
*x)^3+2*ln(e)+2*ln(x))*m)
```

maxima [A] time = 0.84, size = 219, normalized size = 1.60

$$\frac{Bd^3e^mxe^{\left(m\log(x)+4\,n\log(x)\right)}}{m+4\,n+1}+\frac{3\,Bcd^2e^mxe^{\left(m\log(x)+3\,n\log(x)\right)}}{m+3\,n+1}+\frac{Ad^3e^mxe^{\left(m\log(x)+3\,n\log(x)\right)}}{m+3\,n+1}+\frac{3\,Bc^2de^mxe^{\left(m\log(x)+2\,n\log(x)\right)}}{m+2\,n+1}+\frac{3\,Acd^2e^mxe^{\left(m\log(x)+2\,n\log(x)\right)}}{m+2\,n+1}+\frac{Bc^3e^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{3\,Ac^2de^mxe^{\left(m\log(x)+n\log(x)\right)}}{m+n+1}+\frac{(ex)^{m+1}\,Ac^3}{e(m+1)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)^m*(A+B*x^n)*(c+d*x^n)^3,x, algorithm="maxima")
```

```
[Out] B*d^3*e^m*x*e^(m*log(x) + 4*n*log(x))/(m + 4*n + 1) + 3*B*c*d^2*e^m*x*e^(m*
log(x) + 3*n*log(x))/(m + 3*n + 1) + A*d^3*e^m*x*e^(m*log(x) + 3*n*log(x))/
(m + 3*n + 1) + 3*B*c^2*d*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + 3
*A*c*d^2*e^m*x*e^(m*log(x) + 2*n*log(x))/(m + 2*n + 1) + B*c^3*e^m*x*e^(m*l
og(x) + n*log(x))/(m + n + 1) + 3*A*c^2*d*e^m*x*e^(m*log(x) + n*log(x))/(m
+ n + 1) + (e*x)^(m + 1)*A*c^3/(e*(m + 1))
```

mupad [B] time = 5.31, size = 563, normalized size = 4.11

$$\frac{Ac^3x\,(ex)^m}{m+1}+\frac{d^2x\,x^{3n}\,(ex)^m\,(A\,d+3\,B\,c)\,\left(m^3+7\,m^2\,n+3\,m^2+14\,m\,n^2+14\,m\,n+3\,m+8\,n^3+14\,n^2+7\,n+1\right)}{m^4+10\,m^3\,n+4\,m^3+35\,m^2\,n^2+30\,m^2\,n+6\,m^2+50\,m\,n^3+70\,m\,n^2+30\,m\,n+4\,m+24\,n^4+50\,n^3+35\,n^2+10\,n+1}+\frac{c^2x\,x^n\,(ex)^m\,(3\,A\,d+B\,c)\,\left(m^3+9\,m^2\,n+3\,m^2+26\,m\,n^2+18\,m\,n+3\,m+24\,n^3+26\,n^2+9\,n+1\right)}{m^4+10\,m^3\,n+4\,m^3+35\,m^2\,n^2+30\,m^2\,n+6\,m^2+50\,m\,n^3+70\,m\,n^2+30\,m\,n+4\,m+24\,n^4+50\,n^3+35\,n^2+10\,n+1}+\frac{B\,d^3x\,x^{4n}\,(ex)^m\,\left(m^3+6\,m^2\,n+3\,m^2+11\,m\,n^2+12\,m\,n+3\,m+6\,n^3+11\,n^2+6\,n+1\right)}{m^4+10\,m^3\,n+4\,m^3+35\,m^2\,n^2+30\,m^2\,n+6\,m^2+50\,m\,n^3+70\,m\,n^2+30\,m\,n+4\,m+24\,n^4+50\,n^3+35\,n^2+10\,n+1}+\frac{3\,c\,d\,x\,x^{2n}\,(ex)^m\,(A\,d+B\,c)\,\left(m^3+8\,m^2\,n+3\,m^2+19\,m\,n^2+16\,m\,n+3\,m+12\,n^3+19\,n^2+8\,n+1\right)}{m^4+10\,m^3\,n+4\,m^3+35\,m^2\,n^2+30\,m^2\,n+6\,m^2+50\,m\,n^3+70\,m\,n^2+30\,m\,n+4\,m+24\,n^4+50\,n^3+35\,n^2+10\,n+1}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((e*x)^m*(A + B*x^n)*(c + d*x^n)^3,x)
```

```
[Out] (A*c^3*x*(e*x)^m)/(m + 1) + (d^2*x*x^(3*n)*(e*x)^m*(A*d + 3*B*c)*(3*m + 7*n
 + 14*m*n + 14*m*n^2 + 7*m^2*n + 3*m^2 + m^3 + 14*n^2 + 8*n^3 + 1))/(4*m +
10*n + 30*m*n + 70*m*n^2 + 30*m^2*n + 50*m*n^3 + 10*m^3*n + 6*m^2 + 4*m^3 +
 m^4 + 35*n^2 + 50*n^3 + 24*n^4 + 35*m^2*n^2 + 1) + (c^2*x*x^n*(e*x)^m*(3*A
*d + B*c)*(3*m + 9*n + 18*m*n + 26*m*n^2 + 9*m^2*n + 3*m^2 + m^3 + 26*n^2 +
 24*n^3 + 1))/(4*m + 10*n + 30*m*n + 70*m*n^2 + 30*m^2*n + 50*m*n^3 + 10*m^
3*n + 6*m^2 + 4*m^3 + m^4 + 35*n^2 + 50*n^3 + 24*n^4 + 35*m^2*n^2 + 1) + (B
*d^3*x*x^(4*n)*(e*x)^m*(3*m + 6*n + 12*m*n + 11*m*n^2 + 6*m^2*n + 3*m^2 + m
^3 + 11*n^2 + 6*n^3 + 1))/(4*m + 10*n + 30*m*n + 70*m*n^2 + 30*m^2*n + 50*m
```

```
*n^3 + 10*m^3*n + 6*m^2 + 4*m^3 + m^4 + 35*n^2 + 50*n^3 + 24*n^4 + 35*m^2*n
^2 + 1) + (3*c*d*x*x^(2*n)*(e*x)^m*(A*d + B*c)*(3*m + 8*n + 16*m*n + 19*m*n
^2 + 8*m^2*n + 3*m^2 + m^3 + 19*n^2 + 12*n^3 + 1))/(4*m + 10*n + 30*m*n + 7
0*m*n^2 + 30*m^2*n + 50*m*n^3 + 10*m^3*n + 6*m^2 + 4*m^3 + m^4 + 35*n^2 + 5
0*n^3 + 24*n^4 + 35*m^2*n^2 + 1)
```

sympy [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((e*x)**m*(A+B*x**n)*(c+d*x**n)**3,x)
[Out] Timed out
```

Chapter 4

Appendix

Local contents

4.1 Download section

The following zip files contain the raw integrals used in this test.

This is a subset of Rubi test suite thanks to Albert Rich, which includes only the algebraic integrals with elementray optimal antiderivatives. It also includes a subset of a test file provided thanks to Sam Blake.

Mathematica format Mathematica_syntax_CAS_integration_elementary_version.zip

Maple and Mupad format Maple_syntax_CAS_integration_elementary_version.zip

Sympy format SYMPY_syntax_CAS_integration_elementary_version.zip

Sage math format SAGE_syntax_CAS_integration_elementary_version.zip

4.2 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.

4.2.1 Mathematica and Rubi grading function

```
(* Original version thanks to Albert Rich emailed on 03/21/2017 *)
(* ::Package:: *)

(* ::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_] :=
  If[ExpnType[result]<=ExpnType[optimal],
    If[FreeQ[result,Complex] || Not[FreeQ[optimal,Complex]],
      If[LeafCount[result]<=2*LeafCount[optimal],
        "A",
      "B"],
    "C"],
  If[FreeQ[result,Integrate] && FreeQ[result,Int],
    "C",
  "F"]]


(* ::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]
```

4.2.2 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

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";
      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";
  end if;


  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;
            #both result and optimal complex
            if leaf_count_result<=2*leaf_count_optimal then
               return "A";
            else
               return "B";
            end if
         else #result contains complex but optimal is not
            if debug then
               print("result contains complex but optimal is not");
            fi;
            return "C";
         end if
      else # result do not contain complex
           # this assumes optimal do not as well
           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";
           end if
      end if
  else #ExpnType(result) > ExpnType(optimal)
      if debug then
         print("ExpnType(result) > ExpnType(optimal)");
      fi;
      return "C";
  end if

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:
```

4.2.3 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)),ExpnType(op(2,expn)))
    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):

    leaf_count_result  = leaf_count(result)
    leaf_count_optimal = leaf_count(optimal)

    expnType_result  = expnType(result)
    expnType_optimal = expnType(optimal)

    if str(result).find("Integral") != -1:
        return "F"

    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:
                    return "A"
                else:
                    return "B"
            else: #result contains complex but optimal is not
                return "C"
        else: # result do not contain complex, this assumes optimal do not as
   well
            if leaf_count_result <= 2*leaf_count_optimal:
                return "A"
            else:
                return "B"
    else:
        return "C"
```

4.2.4 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'

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.
    """
    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',
           '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")


    #thanks to answer at https://ask.sagemath.org/question/49179/what-is-
    sagemath-equivalent-to-atomic-type-in-maple/
    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:
        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.operands()[0]),expnType(expn.operands()[1]))
    elif expn.operator() == add_vararg or expn.operator() == mul_vararg: #
    isinstance(expn,Add) or isinstance(expn,Mul)
        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")

    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:
                    return "A"
                else:
                    return "B"
            else: #result contains complex but optimal is not
                return "C"
        else: # result do not contain complex, this assumes optimal do not as
    well
            if leaf_count_result <= 2*leaf_count_optimal:
                return "A"
            else:
                return "B"
    else:
        return "C"
```