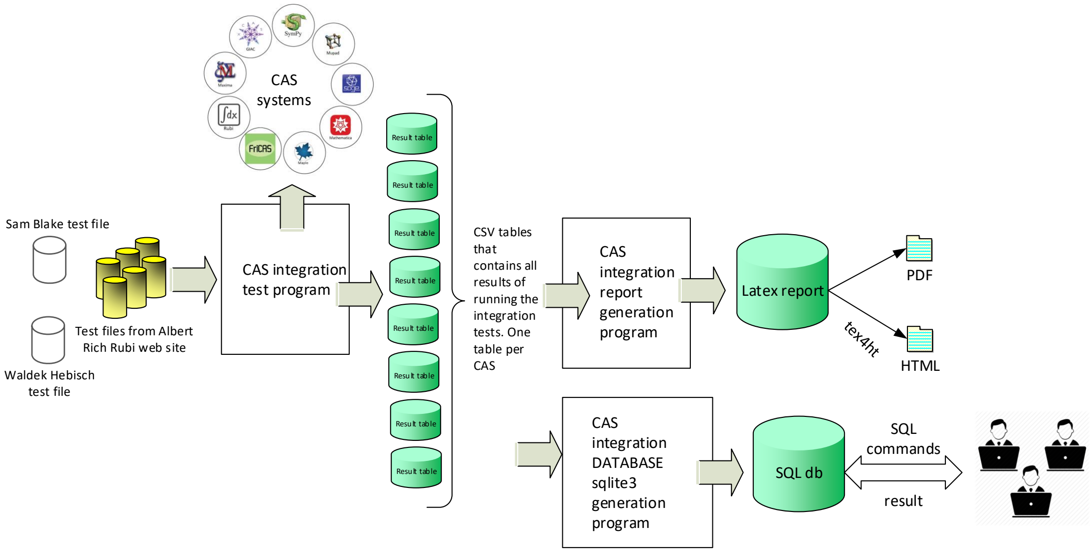

CAS systems
SymPy
Maple
Maxima
Rubi
FriCAS
Mathematica
Sam Blake test file
Test files from Albert Rich Rubi web site
Waldek Hebisch test file
CAS integration test program
Result table
CSV tables that contains all results of running the integration tests. One table per CAS
CAS integration report generation program
Latex report
PDF
tex4ht
HTML
CAS integration DATABASE sqlite3 generation program
SQL db
SQL commands
result

Computer Algebra Independent Integration Tests SQLite3 Database interface

Nasser M. Abbasi

April 14, 2024 Compiled on April 14, 2024 at 12:27 Noon

Contents

1 Introduction 1

2 Examples of SQL commands 7

CHAPTER 1

Introduction

1.1 How to download the database and use it

This is a short introduction on how to download and use the new CAS integration tests SQLite database. The database is an sqlite file.

1. Download cas_integration_tests.zip and extract it. The file inside is called **cas_integration_tests.db** which is a large file (1.3 GB) and is the database file.

2. Install sqlite3 if not already installed on your computer. The web page is sqlite3 database software. This is open source free software which is available to Linux, Mac and Windows. Most linux installation managers can also install this directly.

3. It is recommended to also install DB Browser for SQLite which is a GUI interface to the database that allows one to easily query and view the database.

4. Once sqlite3 is installed then the following command is used to access the database itself

   ```
   sqlite3  cas_integration_tests.db
   >sqlite
   ```

5. Once inside the database. SQL queries are used to find any information needed about the test results. There are many places on the web that show how to use SQL commands.

1.2 Description of the schema

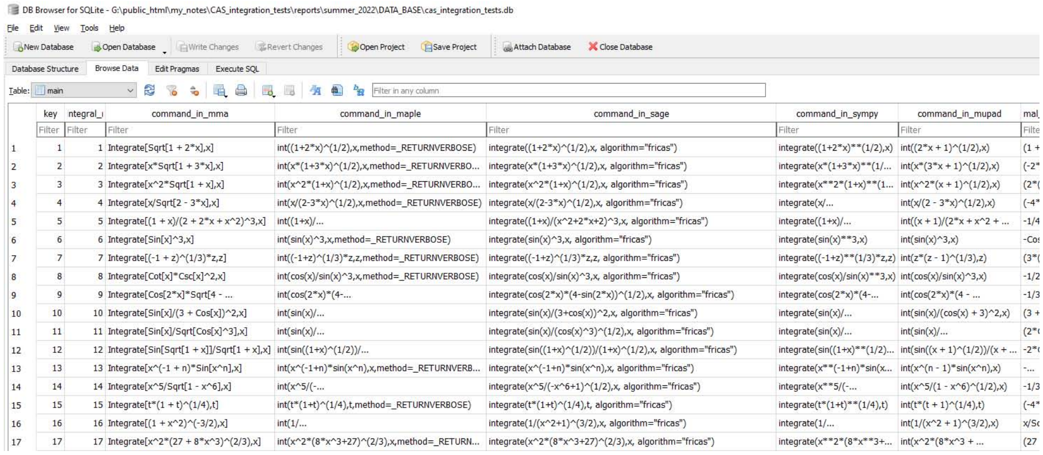
There is only one table called main with currently 72 columns. The following is the result of issuing the command sqlite> PRAGMA table_info(main);. There are 88 columns in one table.

```
sqlite> PRAGMA table_info(main);
0|key|INTEGER|0||1
1|global_integral_number|INTEGER|0||0
2|command_in_mma|TEXT|0||0
3|command_in_rubi|TEXT|0||0
4|command_in_maple|TEXT|0||0
5|command_in_fricas|TEXT|0||0
6|command_in_maxima|TEXT|0||0
7|command_in_giac|TEXT|0||0
8|command_in_sympy|TEXT|0||0
9|command_in_mupad|TEXT|0||0
10|optimal_in_mma|TEXT|0||0
11|optimal_in_maple|TEXT|0||0
12|optimal_leafsize|INTEGER|0||0
13|optimal_anti_in_latex|TEXT|0||0
14|integral_in_latex|TEXT|0||0
15|has_known_anti|INTEGER|0||0
16|input_file_name|TEXT|0||0
17|output_folder|TEXT|0||0
18|input_file_number|INTEGER|0||0
19|integral_number_in_file|INTEGER|0||0
20|rubi_number_of_steps|INTEGER|0||0
21|integrand_size|INTEGER|0||0
22|rubi_pass|INTEGER|0||0
23|rubi_leafsize|INTEGER|0||0
24|rubi_time|TEXT|0||0
25|rubi_anti|TEXT|0||0
26|rubi_anti_in_latex|TEXT|0||0
27|rubi_grade|TEXT|0||0
28|rubi_grade_info|TEXT|0||0
29|rubi_number_of_rules|INTEGER|0||0
30|rubi_ratio|TEXT|0||0
31|rubi_rules_used|TEXT|0||0
32|rubi_verified|INTEGER|0||0
33|mma_pass|INTEGER|0||0
34|mma_leafsize|INTEGER|0||0
35|mma_time|TEXT|0||0
36|mma_anti|TEXT|0||0
37|mma_anti_in_latex|TEXT|0||0
38|mma_grade|TEXT|0||0
```

```
39|mma_grade_info|TEXT|0||0
40|mma_verified|INTEGER|0||0
41|maple_pass|INTEGER|0||0
42|maple_leafsize|INTEGER|0||0
43|maple_time|TEXT|0||0
44|maple_anti|TEXT|0||0
45|maple_anti_in_latex|TEXT|0||0
46|maple_grade|TEXT|0||0
47|maple_grade_info|TEXT|0||0
48|maple_verified|INTEGER|0||0
49|fricas_pass|INTEGER|0||0
50|fricas_leafsize|INTEGER|0||0
51|fricas_time|TEXT|0||0
52|fricas_anti|TEXT|0||0
53|fricas_anti_in_latex|TEXT|0||0
54|fricas_grade|TEXT|0||0
55|fricas_grade_info|TEXT|0||0
56|fricas_verified|INTEGER|0||0
57|maxima_pass|INTEGER|0||0
58|maxima_leafsize|INTEGER|0||0
59|maxima_time|TEXT|0||0
60|maxima_anti|TEXT|0||0
61|maxima_anti_in_latex|TEXT|0||0
62|maxima_grade|TEXT|0||0
63|maxima_grade_info|TEXT|0||0
64|maxima_verified|INTEGER|0||0
65|giac_pass|INTEGER|0||0
66|giac_leafsize|INTEGER|0||0
67|giac_time|TEXT|0||0
68|giac_anti|TEXT|0||0
69|giac_anti_in_latex|TEXT|0||0
70|giac_grade|TEXT|0||0
71|giac_grade_info|TEXT|0||0
72|giac_verified|INTEGER|0||0
73|mupad_pass|INTEGER|0||0
74|mupad_leafsize|INTEGER|0||0
75|mupad_time|TEXT|0||0
76|mupad_anti|TEXT|0||0
77|mupad_anti_in_latex|TEXT|0||0
78|mupad_grade|TEXT|0||0
79|mupad_grade_info|TEXT|0||0
80|mupad_verified|INTEGER|0||0
81|sympy_pass|INTEGER|0||0
82|sympy_leafsize|INTEGER|0||0
83|sympy_time|TEXT|0||0
84|sympy_anti|TEXT|0||0
85|sympy_anti_in_latex|TEXT|0||0
```

```
86|sympy_grade|TEXT|0||0
87|sympy_verified|INTEGER|0||0
88|sympy_grade_info|TEXT|0||0
sqlite>
```

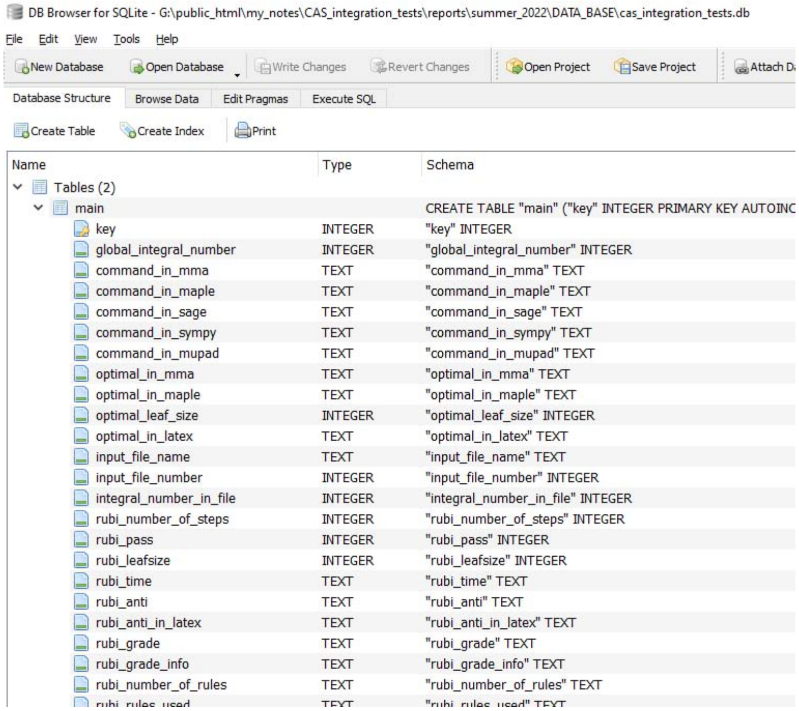
1.3 Screen shots using the database via DB Browser for SQLite



DB Browser for SQLite - G:\public_html\my_notes\CAS_integration_tests\reports\summer_2022\DATA_BASE\cas_integration_tests.db

	key	ntegral_ı	command_in_mma	command_in_maple	command_in_sage	command_in_sympy	command_in_mupad	mal_
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filte
1	1	1	Integrate[Sqrt[1 + 2*x],x]	int((1+2*x)^(1/2),x,method=_RETURNVERBOSE)	integrate((1+2*x)^(1/2),x, algorithm="fricas")	integrate((1+2*x)**(1/2),x)	int((2*x + 1)^(1/2),x)	(1 +
2	2	2	Integrate[x*Sqrt[1 + 3*x],x]	int(x*(1+3*x)^(1/2),x,method=_RETURNVERBO...	integrate(x*(1+3*x)^(1/2),x, algorithm="fricas")	integrate(x*(1+3*x)**(1/...	int(x*(3*x + 1)^(1/2),x)	(-2*
3	3	3	Integrate[x^2*Sqrt[1 + x],x]	int(x^2*(1+x)^(1/2),x,method=_RETURNVERBO...	integrate(x^2*(1+x)^(1/2),x, algorithm="fricas")	integrate(x**2*(1+x)**(1...	int(x^2*(x + 1)^(1/2),x)	(2*(
4	4	4	Integrate[x/Sqrt[2 - 3*x],x]	int(x/(2-3*x)^(1/2),x,method=_RETURNVERBOSE)	integrate(x/(2-3*x)^(1/2),x, algorithm="fricas")	integrate(x/...	int(x/(2 - 3*x)^(1/2),x)	(-4*
5	5	5	Integrate[(1 + x)/(2 + 2*x + x^2)^3,x]	int((1+x)/...	integrate((1+x)/(x^2+2*x+2)^3,x, algorithm="fricas")	integrate((1+x)/...	int((x + 1)/(2*x + x^2 + ...	-1/4
6	6	6	Integrate[Sin[x]^3,x]	int(sin(x)^3,x,method=_RETURNVERBOSE)	integrate(sin(x)^3,x, algorithm="fricas")	integrate(sin(x)**3,x)	int(sin(x)^3,x)	-Cos
7	7	7	Integrate[(-1 + z)^(1/3)*z,z]	int((-1+z)^(1/3)*z,z,method=_RETURNVERBOSE)	integrate((-1+z)^(1/3)*z,z, algorithm="fricas")	integrate((-1+z)**(1/3)*z,z)	int(z*(z - 1)^(1/3),z)	(3*(
8	8	8	Integrate[Cot[x]*Csc[x]^2,x]	int(cos(x)/sin(x)^3,x,method=_RETURNVERBOSE)	integrate(cos(x)/sin(x)^3,x, algorithm="fricas")	integrate(cos(x)/sin(x)**3,x)	int(cos(x)/sin(x)^3,x)	-1/2
9	9	9	Integrate[Cos[2*x]*Sqrt[4 - ...	int(cos(2*x)*(4-...	integrate(cos(2*x)*(4-sin(2*x))^(1/2),x, algorithm="fricas")	integrate(cos(2*x)*(4-...	int(cos(2*x)*(4 - ...	-1/3
10	10	10	Integrate[Sin[x]/(3 + Cos[x])^2,x]	int(sin(x)/...	integrate(sin(x)/(3+cos(x))^2,x, algorithm="fricas")	integrate(sin(x)/...	int(sin(x)/(cos(x) + 3)^2,x)	(3 +
11	11	11	Integrate[Sin[x]/Sqrt[Cos[x]^3],x]	int(sin(x)/...	integrate(sin(x)/(cos(x)^3)^(1/2),x, algorithm="fricas")	integrate(sin(x)/...	int(sin(x)/...	(2*
12	12	12	Integrate[Sin[Sqrt[1 + x]]/Sqrt[1 + x],x]	int(sin((1+x)^(1/2))/...	integrate(sin((1+x)^(1/2))/(1+x)^(1/2),x, algorithm="fricas")	integrate(sin((1+x)**(1/2)...	int(sin((x + 1)^(1/2))/(x + ...	-2*
13	13	13	Integrate[x^(-1 + n)*Sin[x^n],x]	int(x^(-1+n)*sin(x^n),x,method=_RETURNVERB...	integrate(x^(-1+n)*sin(x^n),x, algorithm="fricas")	integrate(x**(-1+n)*sin(x...	int(x^(n - 1)*sin(x^n),x)	-...
14	14	14	Integrate[x^5/Sqrt[1 - x^6],x]	int(x^5/(-...	integrate(x^5/(-x^6+1)^(1/2),x, algorithm="fricas")	integrate(x**5/(-...	int(x^5/(1 - x^6)^(1/2),x)	-1/3
15	15	15	Integrate[t*(1 + t)^(1/4),t]	int(t*(1+t)^(1/4),t,method=_RETURNVERBOSE)	integrate(t*(1+t)^(1/4),t, algorithm="fricas")	integrate(t*(1+t)**(1/4),t)	int(t*(t + 1)^(1/4),t)	(-4*
16	16	16	Integrate[(1 + x^2)^(-3/2),x]	int(1/...	integrate(1/(x^2+1)^(3/2),x, algorithm="fricas")	integrate(1/...	int(1/(x^2 + 1)^(3/2),x)	x/Sc
17	17	17	Integrate[x^2*(27 + 8*x^3)^(2/3),x]	int(x^2*(8*x^3+27)^(2/3),x,method=_RETURN...	integrate(x^2*(8*x^3+27)^(2/3),x, algorithm="fricas")	integrate(x**2*(8*x**3+...	int(x^2*(8*x^3 + ...	(27

Figure 1.1: Database GUI



DB Browser for SQLite - G:\public_html\my_notes\CAS_integration_tests\reports\summer_2022\DATA_BASE\cas_integration_tests.db

Name	Type	Schema
Tables (2)		
main		CREATE TABLE "main" ("key" INTEGER PRIMARY KEY AUTOINC
key	INTEGER	"key" INTEGER
global_integral_number	INTEGER	"global_integral_number" INTEGER
command_in_mma	TEXT	"command_in_mma" TEXT
command_in_maple	TEXT	"command_in_maple" TEXT
command_in_sage	TEXT	"command_in_sage" TEXT
command_in_sympy	TEXT	"command_in_sympy" TEXT
command_in_mupad	TEXT	"command_in_mupad" TEXT
optimal_in_mma	TEXT	"optimal_in_mma" TEXT
optimal_in_maple	TEXT	"optimal_in_maple" TEXT
optimal_leaf_size	INTEGER	"optimal_leaf_size" INTEGER
optimal_in_latex	TEXT	"optimal_in_latex" TEXT
input_file_name	TEXT	"input_file_name" TEXT
input_file_number	INTEGER	"input_file_number" INTEGER
integral_number_in_file	INTEGER	"integral_number_in_file" INTEGER
rubi_number_of_steps	INTEGER	"rubi_number_of_steps" INTEGER
rubi_pass	INTEGER	"rubi_pass" INTEGER
rubi_leafsize	INTEGER	"rubi_leafsize" INTEGER
rubi_time	TEXT	"rubi_time" TEXT
rubi_anti	TEXT	"rubi_anti" TEXT
rubi_anti_in_latex	TEXT	"rubi_anti_in_latex" TEXT
rubi_grade	TEXT	"rubi_grade" TEXT
rubi_grade_info	TEXT	"rubi_grade_info" TEXT
rubi_number_of_rules	TEXT	"rubi_number_of_rules" TEXT
rubi_rules_used	TEXT	"rubi_rules_used" TEXT

Figure 1.2: database structure

CHAPTER 2

Examples of SQL commands

The following section gives examples of SQL commands on the CAS integration tests database. In these examples `rowid` is the main unique key for each row in the table. There is one row per integral. So total number of rows is **85963**. Basically any valid SQL command can be used to query the test results.

2.1 Find rowid of all intergals that rubi scored grade A and Mathematica scored B

```
sqlite> select rowid from main where rubi_grade="A" and mma_grade="C";
41
98
113
175
....
```

2.2 Obtain the leaf size of the integrals in above example

```
sqlite> select mma_leafsize,rubi_leafsize from main where
           rubi_grade="A" and mma_grade="C";
34|172
37|22
24|21
17|103
....
```

2.3 Find the optimal and its leaf size in the first test file only

```
sqlite> select optimal_leafsize,optimal_in_maple from main
          where input_file_number=1;
13|1/3*(1+2*x)^(3/2)
18|-2/27*(1+3*x)^(3/2)+2/45*(1+3*x)^(5/2)
21|2/3*(1+x)^(3/2)-4/5*(1+x)^(5/2)+2/7*(1+x)^(7/2)
18|2/27*(2-3*x)^(3/2)-4/9*(2-3*x)^(1/2)
14|-1/4/(x^2+2*x+2)^2
....
```

2.4 Find all integrals with optimal antiderivative that has the function polylog

Find all integrals with optimal antiderivative that has the function polylog in them, and show the file number and the integral number in that file.

```
sqlite> select input_file_number,integral_number_in_file,
           command_in_fricas,optimal_in_maple from main where
           optimal_in_maple LIKE '%polylog%';
|156|integrate(log(t)/(1+t),t, algorithm="fricas")|ln(t)*ln(1+t)+polylog(2,-t)
2|7|....
```

2.5 Find all integrals with optimal antiderivative that has FresnelS or FresnelC

Find all integrals with optimal antiderivative that has FresnelS or FresnelC in the optimal, and show the file number and the integral number in that file.

```
sqlite> select input_file_number,integral_number_in_file,
              command_in_fricas,optimal_in_maple from main where
              optimal_in_maple LIKE '%FresnelS%' OR'%FresnelC%';
2|4|integrate(sin(x)/(1+x)^(1/2),x, algorithm="fricas")|cos(1)*FresnelS(2^(1/2)/Pi^(1/2)*(1+x)^(1/2))
...
```

2.6 Find the integral with the most used Rubi steps used to solve it.

```
sqlite> select input_file_number,integral_number_in_file,
               rubi_number_of_steps from main where
               rubi_number_of_steps =(SELECT max(rubi_number_of_steps) from main);
210|5294|1980
```

2.7 Find the integrals with the most used Rubi rules to solve it

```
sqlite> select input_file_number,integral_number_in_file,
           rubi_number_of_rules from main where
           rubi_number_of_rules =(SELECT max(rubi_number_of_rules) from main);
210|6214|43
```

2.8 Find the integral with the largest optimal leaf size

```
sqlite> select rowid,input_file_name,input_file_number,
               integral_number_in_file,optimal_leafsize from main where
               optimal_leafsize =(SELECT max(optimal_leafsize) from main);
66679|7_Inverse_hyperbolic_functions/7.3_Inverse_hyperbolic_tangent/7.3.2-d_x-^m-a+b_arctanh-c_x^n-^p

sqlite>  select command_in_mma from main where rowid=66679;
Integrate[(a + b*ArcTanh[c*x^2])^2/(d*x)^(5/2),x]

sqlite> select rubi_time,rubi_grade,rubi_number_of_rules,
           rubi_rules_used from main where rowid=1509;

8.0726703|A|33|{6051, 6043, 6041, 2507, 2521, 212, 2520, 12, 266, 6857, 6131, 6055,
    2449, 2352, 6139, 6057, 2497, 209, 5048, 4966, 5040, 4964, 2505, 218,
    6874, 217, 1179, 642, 1176, 631, 210, 30, 2637}
```

2.9 Find the integrals which used one Rubi step to solve

```
sqlite>select command_in_mma from main where rubi_number_of_steps=1;
Integrate[(-1 + 2*x)^(-1) - (1 + 2*x)^(-1),x]
Integrate[x/(1 - x^2)^5,x]
...
```

2.10 Find the integrals which used two Rubi step to solve

```
sqlite>select command_in_mma from main where rubi_number_of_steps=2;
Integrate[(-3 + x)*(-7 + 4*x^2),x]
Integrate[(-7 + 4*x^2)/(3 + 2*x),x]
...
```

2.11 Find the integrals which used four Rubi step to solve and shows the rules used

```
sqlite>select command_in_mma,rubi_rules_used from main where rubi_number_of_steps=4;
Integrate[(a*x^2 + b*x^3)/(c*x^2 + d*x^3),x]|{1607, 1598, 45}
Integrate[x^4/(4 + 5*x^2 + x^4),x]|{1136, 1180, 209}
...
```

2.12 Find the integral with most uses rubi number of steps

```
sqlite> select rowid,input_file_number,integral_number_in_file,
               rubi_number_of_steps,rubi_number_of_rules,
               command_in_mma  from main where
               rubi_number_of_steps =(SELECT max(rubi_number_of_steps) from main);

80438|210|5294|1980|17|Integrate[(-24*x^3*Log[20] + E^x*(x^3 - 2*x^4)*Log[20])/(-2641807540224*E^2 +
```

2.13 Find how all CAS system scored on the above problem

Find how all CAS system scored on the above problem (1 is pass, 0 is failed)

```
sqlite> select mma_pass,rubi_pass,maple_pass,fricas_pass,maxima_pass,mupad_pass,giac_pass,sympy_pass

      1|1|1|1|1|1|1|1
```

So all solved it

2.14 Find the maximum number of rules Rubi used on a problem

```
sqlite> SELECT DISTINCT rubi_number_of_rules from main where rubi_number_of_rules =(SELECT max(rubi_n

   43
```

2.15 Find how many problems used this maximum number of rules

```
SELECT COUNT(*) from main where rubi_number_of_rules =9;

   1
```

2.16 Find the problem with max leaf size of the optimal solution

```
select rowid,optimal_leafsize from main where optimal_leafsize =(SELECT max(optimal_leafsize) from ma

  66679|6520
```

The above says it is problem #66679 in the database, with leaf size for optimal of 6520

2.17 Find the problem with max leaf size of the optimal solution

```
select rowid,optimal_leafsize from main where optimal_leafsize =(SELECT max(optimal_leafsize) from ma

  66679|6520
```

The above says it is problem #66679 in the database, with leaf size for optimal of 6520

2.18 Find the leaf size of solutions of all other CAS system for the above problem

```
select rubi_leafsize,maple_leafsize,fricas_leafsize,mma_leafsize,maxima_leafsize,giac_leafsize,sympy_

6520|0|0|0|0|0|0
```

zero for leaf size means it was not solved. The above means only Rubi solved it.

2.19 Display the above problem

```
sqlite> select command_in_maple from main where rowid=66679;

      int((a+b*arctanh(c*x^2))^2/(d*x)^(5/2),x)
```

2.20 Find how many problem Fricas uses special functions Frensel

Find how many problem Fricas uses special functions Frensel in its solution

```
sqlite> select COUNT(*) from main where fricas_anti LIKE '%fresnel_sin%' OR'%fresnel_cos%';

      395
```

2.21 Find how many problem Fricas uses Weierstrass special functions

Find how many problem Fricas uses Weierstrass special functions in its solution

```
sqlite>select COUNT(*) from main where fricas_anti LIKE '%weierstrass%';

     4501
```

2.22 Find how many problem Maple uses Frensel

```
sqlite> select COUNT(*) from main where maple_anti LIKE '%fresnel%';

    644
```

2.23 Find number of integrals solved by a CAS

```
sqlite> select count(*) from main where fricas_pass=1;
       69536
```

2.24 Find number of integrals failed by CAS with exception

```
>sqlite3 cas_integration_tests.db
sqlite> select count(*) from main where giac_grade="F(-2)";

     6314
```

2.25 Find percentage of integrals failed by CAS with exception

```
>sqlite3 cas_integration_tests.db
sqlite> select count(*) from main;

   106812

sqlite> select count(*) from main where giac_grade="F(-2)";

     6314
```

So the percentage is 5.9113