

AN APPLICATION OF COMPUTER ALGEBRA SYSTEM CADABRA TO SCIENTIFIC PROBLEMS OF PHYSICS

*L. A. Sevastianov*¹, *D. S. Kulyabov*², *M. G. Kokotchkova*³

Peoples' Friendship University of Russia, Moscow

In this article we present two examples solved in a new problem-oriented computer algebra system Cadabra. Solution of the same examples in widespread universal computer algebra system Maple turns out to be more difficult.

В статье представлены решения двух задач в новой и узкоспециализированной системе компьютерной алгебры Cadabra. Те же примеры в широко распространенной и универсальной системе Maple оказываются значительно сложнее.

PACS: 03.67.Lx, 01.30.Cc, 03.67.-a

INTRODUCTION

The goal of all computer algebra systems (CAS) is to help researchers in manipulations with formulas in physics, mechanics, algebra, analysis problems. Unhandiness of mathematical operations forces to slow down research process and so automatization of formula manipulation is important. Automatization can help to reduce research time. Furthermore, there are problems which cannot be solved without automatization of formula manipulation.

CAS is a software program that facilitates symbolic mathematics. The core functionality of a CAS is manipulation of mathematical expressions in symbolic form.

Actually, there are about 30 CAS (http://en.wikipedia.org/wiki/Comparison_of_computer_algebra_systems) and from these systems we can choose 5 following systems: **Maple** — <http://www.maplesoft.com/>, **Mathematica** — <http://www.wolfram.com/>, **Axiom** — <http://wiki.axiom-developer.org/>, **Maxima** — <http://maxima.sourceforge.net/> and **Cadabra** — <http://www.aei.mpg.de/~peekas/cadabra/>.

We classify them by area of application. Maple, Mathematica, Axiom, and Maxima are universal systems. Cadabra is a problem-oriented system. It means that in a universal system there are many general methods. And problem-oriented system is made for particular tasks.

Kasper Peeters, a high-energy physicist, is a creator of Cadabra. He works now at the Institute for Theoretical Physics of Utrecht University in The Netherlands, and his area of research is located on the overlap among (quantum) field theory, gravity and string theory. He created Cadabra in 2007 for research in field theory and general relativity which are not available in other systems.

¹E-mail: sevast@sci.pfu.edu.ru

²E-mail: dharmadharma@sci.pfu.edu.ru

³E-mail: mgkokotchkova@gmail.com

1. APPLICATION OF CADABRA

CAS Cadabra has called our attention because its particular tasks are in the following areas:

- Field theory,
- Quantum mechanics,
- Quantum field theory,
- Gravity,
- Supergravity,
- General relativity.

Cadabra is a new CAS designed specifically for the solution of problems encountered in field theory. It has extensive functionality for tensor polynomial simplification taking care of Bianchi and Schouten identities, for fermions and anti-commuting variables, Clifford algebras and Fierz transformations, implicit coordinate dependence, multiple index types and many other field theory related concepts. The input format is a subset of \TeX and thus easy to learn.

The program is completely independent of commercial software and relies only on a few other open source libraries and programs. Versions for Linux, as well as for Mac OS X machines, can be downloaded from the web site (<http://www.aei.mpg.de/~peekas/cadabra/index.html>).

Cadabra can operate with Tensors, Spinors, Lie groups and as all CAS in symbolic view. It means that Cadabra has special commands, special data types from idem areas.

2. PECULIARITIES OF CADABRA

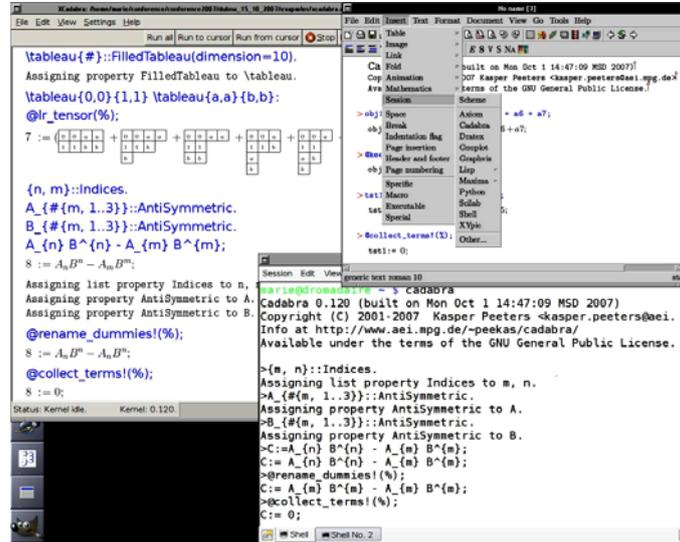
Let us observe several peculiarities of Cadabra [1,2].

- *Usage of \TeX notation for both input and output, which eliminates the errors in transcribing problems from paper to computer and back.* The first and most easily visible feature is that all expression input is in the form of a subset of \TeX . Tensor indices, Dirac conjugation, derivative operators, commutators, fermion products and so on are all written just as in \TeX . With a little bit of discipline, one can cut and paste expressions straight from a paper into a Cadabra notebook. The output, similarly, is typeset as \TeX would do it, and Cadabra notebook files are in fact at the same time also valid \TeX files (the program comes with a graphical notebook interface, but can also be used from the command line). Being able to input expressions like this would eliminate a large number of errors in transcribing physics problems to CAS.

- *An optional unlimited undo system. Interactive calculations can be undone to arbitrary level without requiring a full re-evaluation of the entire calculation.* We can infinitely many times return on each line of program, change something and restart it. But only command line interface in terminal does not give this possibility.

- *A simple and documented way to add new algorithms in the form of C++ modules, which directly operate on the internal expression tree.* The language that Cadabra was built on is C++, we can add some algorithms and functions in C++.

- *A command line interface as well as a graphical one, and a \TeX macs front-end.* Cadabra has 3 interfaces: command line interface, Xcadabra and \TeX macs front-end. This picture (see figure) demonstrates it.



Cadabra in command line interface, Xcadabra and Cadabra in \TeX macs

- Built-in understanding of «dummy indices» and dummy symbols, including their automatic relabelling when necessary. Powerful algorithms for canonicalisation of objects with index symmetries, both mono-term and multi-term. Cadabra can operate indices: rename them, change them.
- A new way to deal with products of non-commuting objects, enabling a notation which is identical to standard physicist's notation (i.e., no need for special non-commuting product operators). It is useful for non-commuting objects.
- A flexible way to associate meaning («type information») to tensors by attaching them to «properties». In Cadabra there are many built-in types of tensor (tensor Weyl, tensor Riemann and so on).

3. EXAMPLES IN CADABRA

Let us consider two examples that illustrate Cadabra in action [1, 2].

3.1. Weyl Tensor. The first example is devoted to operations with Weyl tensor. Let us use the fact that Weyl tensor is of built-in type. We want to prove the following identity:

$$\begin{aligned}
 W_{pqrs} W_{ptru} W_{tvqw} W_{uvsu} - W_{pqrs} W_{pqtu} W_{rvtu} W_{svuw} &= \\
 &= W_{mnab} W_{npbc} W_{mscd} W_{spda} - \frac{1}{4} W_{mnab} W_{psba} W_{mpcd} W_{nsdc}. \quad (1)
 \end{aligned}$$

```

1 {m,n,p,q,r,s,t,u,v,w,a,b,c,d,e,f}::Indices(vector).
2 W_{m n p q}::WeylTensor.
3 W_{p q r s} W_{p t r u} W_{t v q w} W_{u v s w}
4 - W_{p q r s} W_{p q t u} W_{r v t w} W_{s v u w}
5 - W_{m n a b} W_{n p b c} W_{m s c d} W_{s p d a}
6 + (1/4) W_{m n a b} W_{p s b a} W_{m p c d} W_{n s d c};
    
```

We set indices of type vector in the first line. In the second line we determine type of Weyl tensor. We transpose all terms of our identity (lines 3–6) at the right side of equality. We get in output

$$1 := W_{pqrs}W_{ptru}W_{tvqw}W_{uvsw} - W_{pqrs}W_{pqtu}W_{rvtw}W_{svuw} - \\ - W_{mnab}W_{npsc}W_{mscd}W_{spda} + \frac{1}{4}W_{mnab}W_{psba}W_{mpcd}W_{nsdc}. \quad (2)$$

After that we preform the following operations:

```
1 @young_project_tensor!(%){ModuloMonoterm};
2 @distribute!(%);
3 @canonicalise!(%);
4 @rename_dummies!(%);
5 @collect_terms!(%);
```

Finally, we have got the result where «2:» means command number of system output:

$$2 := 0.$$

Let us discuss the sequence of commands used. In the example we deal with rotation group $SO(d)$. Young tables help us working with group of symmetry (line 1). Command `@distribute!` in line 2 removes brackets using distribution rules. In line 3 we convert the expression to canonical form with the command `@canonicalise!`. After that we rename indices (line 4), collect similar terms (line 5) and as a result we have got 0. The identity is approved.

3.2. Product of Gamma Matrices. Gamma matrices are often used in operations with spinors. Cadabra has built-in algorithms, for example, algorithms for manipulation of gamma matrices in symbolic dimension. In the following example we shall expand the product $\gamma_{sr}\gamma_{rl}\gamma_{km}\gamma_{ms}$, as a result we should get the combination of γ_{kl} and δ_{kl} components:

$$\gamma_{sr}\gamma_{rl}\gamma_{km}\gamma_{ms} = ?$$

```
1 ::PostDefaultRules(@@prodsort!(%),@@eliminate_kr!(%),
2 @@canonicalise!(%),@@collect_terms!(%)).
3 {s,r,l,k,m,n}::Indices(vector).
4 {s,r,l,k,m,n}::Integer(0..d-1).
5 \gamma_{#}::GammaMatrix(metric=\delta).
6 \delta_{m n}::KroneckerDelta.
7 \gamma_{s r} \gamma_{r l} \gamma_{k m} \gamma_{m s};
8 @join!(%){expand};
9 @join!(%){expand};
```

In the first line there are some default simplification rules of the system which the system will do after every stage. Let us write lines 3–6 to declare the built-in types of objects: indices, their range, gamma matrix and Kronecker delta. After that we write initial product (line 7). Using the command `@join!` two times we expand product of two adjacent gamma matrices:

$$1 := (-1)\gamma_{mr}\gamma_{lm}\gamma_{ks}\gamma_{rs}; \\ 2 := (-1)(2\gamma_{lr} - d\gamma_{lr} + \delta_{lr}d - \delta_{lr})\gamma_{ks}\gamma_{rs}; \\ 3 := (-1)(2\gamma_{lr} - \gamma_{lr}d + \delta_{lr}d - \delta_{lr})(2\gamma_{kr} - d\gamma_{kr} + \delta_{kr} - \delta_{kr}d).$$

Further, as in first example, let us use commands of reduction:

```

1 @distribute! (%);
2 @join! (%){expand};
3 @distribute! (%);
4 @factorise! (%){d};
5 @collect_factors! (%);

```

As a result, after each command we have these expressions where the last is the desired one:

$$\begin{aligned}
 4 &:= -4 \gamma_{lr} \gamma_{kr} + 4 \gamma_{lr} \gamma_{kr} d + 4 \gamma_{kl} - 6 \gamma_{kl} d - \gamma_{lr} \gamma_{kr} dd + 2 \gamma_{kl} dd - 2 \delta_{kl} d + \delta_{kl} dd + \delta_{kl}; \\
 5 &:= 12 \gamma_{kl} - 4 d \gamma_{kl} - 3 \delta_{kl} + 2 \delta_{kl} d + 4 (-2 \gamma_{kl} + d \gamma_{kl} + \delta_{kl} - \delta_{kl} d) d - 6 \gamma_{kl} d - \\
 &\quad - (-2 \gamma_{kl} + d \gamma_{kl} + \delta_{kl} - \delta_{kl} d) dd + 2 \gamma_{kl} dd + \delta_{kl} dd; \\
 6 &:= 12 \gamma_{kl} - 18 \gamma_{kl} d - 3 \delta_{kl} + 6 \delta_{kl} d + 8 \gamma_{kl} dd - 4 \delta_{kl} dd - \gamma_{kl} ddd + \delta_{kl} ddd; \\
 7 &:= \gamma_{kl} (12 - 18 d + (8 (dd) - (ddd))) + \delta_{kl} (-3 + 6 d + (-4 (dd) + (ddd))); \\
 8 &:= \gamma_{kl} (12 - 18 d + 8 d^2 - d^3) + \delta_{kl} (-3 + 6 d - 4 d^2 + d^3).
 \end{aligned}$$

Comparison of Cadabra and Maple

	Cadabra	Maple
1. Creator	Kasper Peeters	MapleSoft
2. First public release	2007	1985
3. Latest stable version	0.128 (November 28, 2007)	11.02 (November 26, 2007)
4. Open source	Yes, with source code	No
5. License	GPL	Proprietary
6. Operating system support	Mac OS X, Linux	Windows, Mac OS X, Linux
7. Language	C++	C
8. On-line help	No	Yes
9. Graphics	No	Yes
10. Modules support	Yes	Yes
11. Input	T _E X notation and commands of system	Commands of system
12. Output	T _E X subset	T _E X, HTML
13. Application	Quantum field theory, gravity, supergravity	Universal

4. COMPARISON OF CADABRA AND MAPLE

In the end let us compare two CAS: Cadabra and Maple. Results of comparison are shown in the table where documentation and particular experience were used. Besides, information from the source http://en.wikipedia.org/wiki/Comparison_of_computer_algebra_systems was considered. Maple is a well-known and widespread system working since 1985.

Cadabra was created in 2006 and was developed by Kasper Peeters. It is made for some scientific problems only, but is made profoundly.

Latest stable versions of both systems were released in November 2007, the systems are burning and need upgradings. The advantage of Cadabra is that it is distributed under GPL and its source code is open. This fact is very important in Russia nowadays. Particularly, in Peoples' Friendship University of Russia free operation systems become more popular. From this point of view Cadabra can be supported by Linux, Mac OS X and can be ported on Windows. Maple supports all operation systems, but even the student's version of Maple costs about \$100, and the full version costs 20 times more.

Cadabra, like Maple, has possibility to add modules, but has not graph support. Another Cadabra's disadvantage is absence of interactive help.

One of Cadabra's advantages is that input and output are in \TeX notation. That can be useful for writing scientific articles and books. To insert a formula or result of calculations you simply need to cut it from Cadabra window and paste into the text.

The next point of comparison is an area of applications. Maple is a universal system which contains many algorithms of calculation methods. Cadabra's area of applications is limited to problems in field theory.

We have seen some examples of Cadabra in Sec.3. It turned out that it is impossible to solve these examples in Maple in symbolic form because Maple has the following problems:

- It is impossible to operate with Weyl tensors in symbolic form.
- It is possible to operate with gamma matrices of one index only.
- The dimension of space must be defined numerically and cannot be symbolic, for example, d .
- Maple can operate only with matrices and their components, but not with tensors in general form, whereas Cadabra can work with abstract component tensors as well as with individual components.

CONCLUSION

Having compared the two systems, we can see that they have different paradigms. Maple can be successfully used in *general engineering problems*. It was built to solve equation systems and hence can work with matrices. To manipulate with tensors we need to represent all its components in matrices, too. To represent a symmetric or antisymmetric tensor, we need to symmetrize or antisymmetrize the corresponding matrix. Cadabra was designed for *scientists* dealing with *physical problems*. It can work with objects of field theory in the form of abstract components and in the numerical form.

We recommend Cadabra CAS as a useful tool for scientists in their physical research.

REFERENCES

1. *Peeters K.* Introducing Cadabra: A Symbolic Computer Algebra System for Field Theory Problems. hep-th/0701238 25. 2007; http://www.aei.mpg.de/~peekas/cadabra/cadabra_hep.ps
2. *Peeters K.* Cadabra: Reference Guide and Tutorial. Preprint AEI-2006-038. 2001–2007; <http://www.aei.mpg.de/~peekas/cadabra/cadabra.ps>