AN APPLICATION OF COMPUTER ALGEBRA SYSTEM CADABRA TO QUANTUM FIELD

M. G. Kokotchikova, D. S. Kulyabov, L. A. Sevastianov

Peoples' Friendship University of Russia

QPC-2007, Dubna, Russia
15–19 October, 2007

Table of Contents

Bibliography

- Cadabra. A field-theory motivated approach to computer algebra, 2001-2007, Kasper Peeters, `http://www.aei.mpg.de/~peekas/cadabra/`
- Kasper Peeters. Introducing Cadabra: a symbolic computer algebra system for field theory problems. — http://www.aei.mpg.de/p̃eekas/cadabra/cadabra_hep.ps

CAS — Computer Algebra System

A computer algebra system (CAS) is a software program that facilitates symbolic mathematics. The core functionality of a CAS is manipulation of mathematical expressions in symbolic form.

- Maple — `http://www.maplesoft.com/`
- Mathematica — `http://www.wolfram.com/`
- Axiom — `http://wiki.axiom-developer.org/`
- Maxima — `http://maxima.sourceforge.net/`
- Cadabra — `http://www.aei.mpg.de/~peekas/cadabra/`

Classification of CAS

- Universal System
  - Maple
  - Mathematica
  - Axiom
  - Maxima
- Problem-oriented System
  - Cadabra

Applicable to

- Field Theory
- Quantum Mechanics
- Quantum Field Theory

Mathematical Apparatus

- Tensor
- Spinor
- Lie group

Peculiarities of Cadabra

User Interface

- Usage of TEX notation for both input and output, which eliminates many errors in transcribing problems from paper to computer and back.
- An optional unlimited undo system. Interactive calculations can be undone to arbitrary level without requiring a full re-evaluation of the entire calculation.
- A simple and documented way to add new algorithms in the form of $C++$ modules, which directly operate on the internal expression tree.
- A command line interface as well as a graphical one, and a TEXmacs frontend.

Peculiarities of Cadabra

Cadabra In Command Line Interface

marie@dromadaire:~ - Shell - Konsole

Session Edit View Bookmarks Settings Help

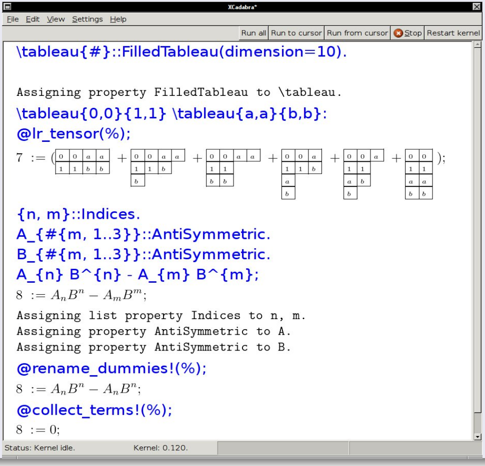
```
marie@dromadaire ~ $ cadabra
Cadabra 0.120 (built on Mon Oct 1 14:47:09 MSD 2007)
Copyright (C) 2001-2007  Kasper Peeters <kasper.peeters
@aei.mpg.de>
Info at http://www.aei.mpg.de/~peekas/cadabra/
Available under the terms of the GNU General Public Lic
ense.

>{n, m}::Indices.
Assigning list property Indices to n, m.
>A_{#{m, 1..3}}::AntiSymmetric.
Assigning property AntiSymmetric to A.
>B_{#{m, 1..3}}::AntiSymmetric.
Assigning property AntiSymmetric to B.
>A_{n} B^{n} - A_{m} B^{m};
1:= A_{n} B^{n} - A_{m} B^{m};
>@rename_dummies!(%);
1:= A_{n} B^{n} - A_{n} B^{n};
>@collect_terms!(%);
1:= 0;
>
```

Shell

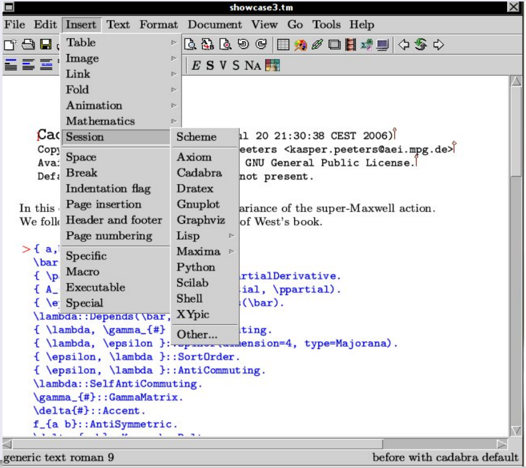
Peculiarities of Cadabra

XCadabra



Peculiarities of Cadabra

Cadabra In TEXmacs



Peculiarities of Cadabra

Field Theory Features

- 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.
- 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).
- A flexible way to associate meaning («type information») to tensors by attaching them to «properties».

Example (1. Weyl Tensor 1/2)

Consider the identity:

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

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

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

Example (1. Weyl Tensor 2/2)

XCadabra: /home/marie/conference/conference2007/dubna_15_10_2007/exapmles/weil.cdb*

File Edit View Settings Help

Run all | Run to cursor | Run from cursor | Stop | Restart kernel

```
@young_project_tensor!(%){ModuloMonoterm};
```

$$
\begin{aligned}
22 := {} & (\tfrac{2}{3} W_{pqrs} - \tfrac{1}{3} W_{psqr} + \tfrac{1}{3} W_{prqs})(\tfrac{2}{3} W_{ptru} + \tfrac{1}{3} W_{purt} + \tfrac{1}{3} W_{prtu})(\tfrac{2}{3} W_{qwtv} \\
& + \tfrac{1}{3} W_{qvtw} - \tfrac{1}{3} W_{qtvw})(\tfrac{2}{3} W_{swuv} + \tfrac{1}{3} W_{svuw} - \tfrac{1}{3} W_{suvw}) \\
& - (\tfrac{2}{3} W_{pqrs} - \tfrac{1}{3} W_{psqr} + \tfrac{1}{3} W_{prqs})(\tfrac{2}{3} W_{pqtu} - \tfrac{1}{3} W_{puqt} + \tfrac{1}{3} W_{ptqu})(\tfrac{2}{3} W_{rvtw} \\
& + \tfrac{1}{3} W_{rwtv} + \tfrac{1}{3} W_{rtvw})(\tfrac{2}{3} W_{svuw} + \tfrac{1}{3} W_{swuv} + \tfrac{1}{3} W_{suvw}) \\
& - (\tfrac{2}{3} W_{abmn} - \tfrac{1}{3} W_{anbm} + \tfrac{1}{3} W_{ambn})(\tfrac{2}{3} W_{bcnp} - \tfrac{1}{3} W_{bpcn} + \tfrac{1}{3} W_{bncp})(\tfrac{2}{3} W_{cdms} \\
& - \tfrac{1}{3} W_{csdm} + \tfrac{1}{3} W_{cmds})(\tfrac{2}{3} W_{adps} - \tfrac{1}{3} W_{asdp} + \tfrac{1}{3} W_{apds}) \\
& + \tfrac{1}{4} (\tfrac{2}{3} W_{abmn} - \tfrac{1}{3} W_{anbm} + \tfrac{1}{3} W_{ambn})(-\tfrac{2}{3} W_{abps} - \tfrac{1}{3} W_{apbs} + \tfrac{1}{3} W_{asbp})(\tfrac{2}{3} W_{cdmp} \\
& - \tfrac{1}{3} W_{cpdm} + \tfrac{1}{3} W_{cmdp})(-\tfrac{2}{3} W_{cdns} - \tfrac{1}{3} W_{cnds} + \tfrac{1}{3} W_{csdn});
\end{aligned}
$$

```
@distribute!(%):
@canonicalise!(%):
@rename_dummies!(%):
@collect_terms!(%);
```

$$22 := 0;$$

Status: Kernel idle. Kernel: 0.120.

Example (2. Product Of Gamma Matrix (1/2))

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

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

$$1 := (-1)\gamma_{mr}\gamma_{lm}\gamma_{ks}\gamma_{rs};$$

$$2 := (-1)(2\gamma_{lr} - d\gamma_{lr} + \delta_{lr}d - \delta_{lr})\gamma_{ks}\gamma_{rs};$$

$$3 := (-1)(2\gamma_{lr} - \gamma_{lr}d + \delta_{lr}d - \delta_{lr})(2\gamma_{kr} - d\gamma_{kr} + \delta_{kr} - \delta_{kr}d);$$

Example (2. Product Of Gamma Matrix (2/2))

XCadabra: /home/marie/conference/conference2007/dubna_15_10_2007/exapmles/gamma.cdb*

File Edit View Settings Help

Run all | Run to cursor | Run from cursor | Stop | Restart kernel

```
@distribute!(%);
@join!(%){expand};
@distribute!(%);
@factorise!(%){d};
@collect_factors!(%);
```

$$4 \;:= -4\,\gamma_{lr}\gamma_{kr} + 4\,\gamma_{lr}\gamma_{kr}d + 4\,\gamma_{kl} - 6\,\gamma_{kl}d - \gamma_{lr}\gamma_{kr}dd + 2\,\gamma_{kl}dd - 2\,\delta_{kl}d + \delta_{kl}dd + \delta_{kl};$$

$$4 \;:= 12\,\gamma_{kl} - 4\,d\gamma_{kl} - 3\,\delta_{kl} + 2\,\delta_{kl}d + 4\,(-2\,\gamma_{kl} + d\gamma_{kl} + \delta_{kl} - \delta_{kl}d)d - 6\,\gamma_{kl}d - (-2\,\gamma_{kl} + d\gamma_{kl} + \delta_{kl} - \delta_{kl}d)dd + 2\,\gamma_{kl}dd + \delta_{kl}dd;$$

$$4 \;:= 12\,\gamma_{kl} - 18\,\gamma_{kl}d - 3\,\delta_{kl} + 6\,\delta_{kl}d + 8\,\gamma_{kl}dd - 4\,\delta_{kl}dd - \gamma_{kl}ddd + \delta_{kl}ddd;$$

$$4 \;:= \gamma_{kl}(12 - 18\,d + (8\,(dd) - (ddd))) + \delta_{kl}(-3 + 6\,d + (-4\,(dd) + (ddd)));$$

$$4 \;:= \gamma_{kl}(12 - 18\,d + 8\,d^2 - d^3) + \delta_{kl}(-3 + 6\,d - 4\,d^2 + d^3);$$

Status: Kernel idle. Kernel: 0.120.

Example (3. Maxwell (1/3))

```
{ a,b,c,d,e }::Indices(vector).
{\partial{#}, \nabla{#}}::PartialDerivative.
{ A_{a}, F_{a b} }::Depends(\partial).
{ a,b,c,d,e }::Indices(vector).
\delta{#}::Accent.
F_{a b}::AntiSymmetric.
\delta_{a b}::KroneckerDelta.
S:= -(1/4) F_{a b} F_{a b};
```

$$S := (-1/4)F_{ab}F_{ab};$$

```
@substitute!(%)(F_{a b}->\partial_{a}{A_{b}}
                         -\partial_{b}{A_{a}});
```

$$S := (-1/4)(\partial_a A_b - \partial_b A_a)(\partial_a A_b - \partial_b A_a);$$

Example (3. Maxwell (2/3))

XCadabra: /home/marie/conference/conference2007/dubna_15_10_2007/soft/source/cadabra-0.120/examples/maxwell.cnb*

File Edit View Settings Help

Run all | Run to cursor | Run from cursor | Stop | Restart kernel

```
@distribute!(%);
```

$$S := -\frac{1}{4}\,\partial_{a}A_{b}\partial_{a}A_{b} + \frac{1}{4}\,\partial_{a}A_{b}\partial_{b}A_{a} + \frac{1}{4}\,\partial_{b}A_{a}\partial_{a}A_{b} - \frac{1}{4}\,\partial_{b}A_{a}\partial_{b}A_{a};$$

```
@vary!(%)( A_{b} -> \delta{A_{b}});
```

$$S := -\frac{1}{4}\,\partial_{a}\delta A_{b}\partial_{a}A_{b} - \frac{1}{4}\,\partial_{a}A_{b}\partial_{a}\delta A_{b} + \frac{1}{4}\,\partial_{a}\delta A_{b}\partial_{b}A_{a} + \frac{1}{4}\,\partial_{a}A_{b}\partial_{b}\delta A_{a} + \frac{1}{4}\,\partial_{b}\delta A_{a}\partial_{a}A_{b} + \frac{1}{4}\,\partial_{b}A_{a}\partial_{a}\delta A_{b} - \frac{1}{4}\,\partial_{b}\delta A_{a}\partial_{b}A_{a} - \frac{1}{4}\,\partial_{b}A_{a}\partial_{b}\delta A_{a};$$

```
@prodsort!(%);
@rename_dummies!(%);
@collect_terms!(%);
```

$$S := -\frac{1}{4}\,\partial_{a}A_{b}\partial_{a}\delta A_{b} - \frac{1}{4}\,\partial_{a}A_{b}\partial_{a}\delta A_{b} + \frac{1}{4}\,\partial_{b}A_{a}\partial_{a}\delta A_{b} + \frac{1}{4}\,\partial_{a}A_{b}\partial_{b}\delta A_{a} + \frac{1}{4}\,\partial_{a}A_{b}\partial_{b}\delta A_{a} + \frac{1}{4}\,\partial_{b}A_{a}\partial_{a}\delta A_{b} - \frac{1}{4}\,\partial_{b}A_{a}\partial_{b}\delta A_{a} - \frac{1}{4}\,\partial_{b}A_{a}\partial_{b}\delta A_{a};$$

$$S := -\frac{1}{4}\,\partial_{a}A_{b}\partial_{a}\delta A_{b} - \frac{1}{4}\,\partial_{a}A_{b}\partial_{a}\delta A_{b} + \frac{1}{4}\,\partial_{a}A_{b}\partial_{b}\delta A_{a} + \frac{1}{4}\,\partial_{a}A_{b}\partial_{b}\delta A_{a} + \frac{1}{4}\,\partial_{a}A_{b}\partial_{b}\delta A_{a} + \frac{1}{4}\,\partial_{a}A_{b}\partial_{b}\delta A_{a} - \frac{1}{4}\,\partial_{a}A_{b}\partial_{a}\delta A_{b} - \frac{1}{4}\,\partial_{a}A_{b}\partial_{a}\delta A_{b};$$

$$S := -\partial_{a}A_{b}\partial_{a}\delta A_{b} + \partial_{a}A_{b}\partial_{b}\delta A_{a};$$

Status: Kernel idle   Kernel: 0.120

Example (3. Maxwell (3/3))