A proposal to design expert system for the calculations in the domain of QFT

Andrea Severe S. *

October 7, 2013

Abstract

Main purposes of the paper are followings: 1) To show examples of the calculations in domain of QFT via "derivative rules" of an expert system; 2) To consider advantages and disadvantage that technology of the calculations; 3) To reflect about how one would develop new physical theories, what knowledge would be useful in their investigations and how this problem can be connected with designing an expert system.

Keywords: expert system, quantum field theory, CAS, FeynCalc, knowledge, knowledge representation

1 Introduction

An analytical-calculation technology is a manipulation technique of the symbolic expressions, corresponding some mathematical objects, by digital computers [1]. As examples of the realization of the symbolic calculations systems one can name most popular systems such as: MAXIMA (http://maxima.sourceforge.net/, "5.9.0", 2003), *Mathematica* (http://www.wolfram.com, "5", 2003), Maple (http://www.maplesoft.com , "9.0", 2003), muPAD (http://www.mupad.de/index_uni.shtml, "2.5", 2002). General ideas engrained in these systems are usually referred as Computer Algebra technique (CA) and systems themselves are called CASs.

CAS usually provides general environment for manipulation of the symbolic expressions, and for the calculations in specific knowledge domain special-purpose packages are created. For example, FeynCalc package [2] for *Mathematica* system provides facilities to do calculation upon objects in QFT.

The packages extending field of application of the CAS are usually worked out in procedural-programming paradigm. That is each package usually provides a number of functions which can be used. However it frequently happens that user should transform an answer which the employed function give out, and transformation chain can be nontrivial [3]. Moreover let's noted that a number of the possible transformation functions can be large.

*sys01@narod.ru; Russia, Protvino

By using expert system's [4] "derivation rules" technique one can find desirable answer by calling only one function - a searching function (and a searching time certainly increases to one or more powers in the value in return). Besides the simple convenience this possibility opens new outlook: since no human assistance is needed for the program to find an answer then one could pose a problem of next generation of the complexity. For example, it will be possible to design the systems which could formulate a searching task or could determine equivalence of several searching tasks. "Consideration" section of the paper is devoted to this problem.

There are common prejudices about expert systems and their application in the scientific research. "Appendix A" contains two very simple examples of the calculation via "derivation rules".

2 Consideration[1]

One can propose the following structure of the physical knowledge:

$$< Phys.Problem\{i\} > + < Phys.Theory > \Longrightarrow q_i,$$

where $< Phys.Problem\{i\} >$ – is a formulation of the physical problem, '$+ < Phys.Theory >$' means — in term of the certain physical theory; q_i – are the mathematical expressions that can be probed experimentally.

In order to do computer calculations symbolic representation of the physical and mathematical objects should be given:

$$< Phys.Problem\{i\} > \longrightarrow [\mathrm{Problem\{i\}}]$$

$$< Phys.Theory > \longrightarrow [\mathrm{ES}],$$

$$q_i \longrightarrow [\mathrm{q_i}],$$

where [...] assigns symbolic representation of the object '...'. Let's imagine that $< Phys.Theory >$ can be represented as an expert system. Then calculation can be expressed as:

$$[\mathrm{q_i} == \mathrm{ES}(\mathrm{Problem\{i\}})]$$

These expressions can be understood as current state of out knowledge expressed in term of some computer system (for example, CAS). Any increase or modification of the physical knowledge should be reflected in form of that expressions (opposite statement is not take place).

Let's assume that some computer symbolic transformation can be constructed such as:

$$[\mathrm{ES} \to \mathrm{ES}'],$$

$$[\mathrm{Problem\{i\}} \to \mathrm{Problem\{i\}}'],$$

$$[\mathrm{q_i} \to \mathrm{q_i} + \Delta_\mathrm{i}(\mathrm{ES}, \mathrm{Task})],$$

[1] None of the scientific approaches were used in this section. Nevertheless the author thinks that the considerations could be interesting.

where numeric values of the Δ_i less then uncertainty of the $< q_i >$. And moreover:

$$[\mathrm{ES}'] \longrightarrow < Phys.Theory' >$$

then one could conclude that $< Phys.Theory' >$ can be treated as new physical theory.

In the end of the section author would like to notice a reason why QFT is especial interesting domain for designing the expert system. If ES for the calculation in QFT would be created one could ask a question why only this semantic structure corresponding to the structure of the ES's symbolic productions is marked out in nature. May be realization of the reason will be helpful in better understanding of nature. The more fundamental theory is taken to build an expert system for the more fundamental semantic structure of the knowledge of the ES can be investigated.

3 Conclusions

This paper eventually contains more questions then answers. Sorry for that. Nevertheless let's do some summary in following items:

- Examples of the calculations in QFT domain via "derivative rules" of an expert system are presented in Appendices A and B;
- Advantages of that technology of the calculations are as followings: full automation of the calculation in some specific domain (and since increase labour productivity of the theorists); new programming architecture which will demand further investigations; new outlook in the investigation of the physical theories.
- Disadvantages of that technology of the calculations are as followings: huge increase of the time of the calculations; large cost in term of man-year-intelligence.

4 Acknowledgements

I would like to thank my mother and friends.
This project **was not** supported by RFBF by the project 04-07-90165.

References

[1] K.S. Tan, W.-H. Steeb, Y. Hardy, Symbolic C++ / Edited by G.M. Kobelkova // M. Mir, 2001, 622 p.

[2] FeynCalc, www.feyncalc.org

[3] Wester, (Editor)(1999), Computer Algebra Systems: A Practical Guide,UK: John Wiley & Sons; Chapter "A Critique of the Mathematical Abilities of CA Systems"; http://www.math.unm.edu/~wester/cas_review.html

[4] AAAI, http://aaai.org/AITopics/html/expert.html;
CLIPS, http://www.ghg.net/clips/CLIPS.html

[5] M. Peskin, D. Schroeder, "An Introduction to Quantum Field Theory", Trans. from English edited by A. A. Belavin, A. V. Berkov, M.-Izhevsk, NIC "R&C Dynamics", 2001; Task 2.1(a,b)

5 Appendix A

This appendix contains very simple examples of the calculation via "derivation rules" and exhibits calculation of the Momentum-Energy tensor in the theory of free electro-magnetic field. First we calculate non-symmetry TEM ($T[\mu,\nu]$), check it's property and then symmetrize it ($T1[\mu,\nu]$). This standard academic task is taken from [5].

Searching procedure has name 'Search'.

The examples can be downloaded from http://sys01.narod.ru/fc-1-10.nb

```
<< HighEnergyPhysics`FeynCalc`

A[m_] = QuantumField[A, {m}]
```

A_m

```
t[m_, n_] = ExpandPartialD[A[n].LeftPartialD[m]]
```

$\partial_m A_n$

```
F[m_, n_] = 2 * AntiSymmetrize[t[m, n], {m, n}]
```

$\partial_m A_n - \partial_n A_m$

```
L = 1 / 4 * F[m, n] * F[m, n]
```

$\frac{1}{4} (\partial_m A_n - \partial_n A_m)^2$

```
MetricTensor[l__] :=
 MetricTensor[Sequence @@ Sort[{l}]] /; Not[OrderedQ[{l}]]


(* --- Building Energy-Momentum Tensor --- *)

q[m_, n_] = QuantumField[B, {m, n}]
```

QuantumField (B, m, n)

```
t0001 = Expand[L] /. t[m_, n_] → q[m, n];
```

```
t0002 = (-FunctionalD[t0001, q[k, l]] /. q[k_, l_] → t[k, l])
```

$\partial_l A_k - \partial_k A_l$

```
T[l_, r_] = t0002 * t[r, k] - MetricTensor[l, r] * L;

(* *** Checking property of T... *** *)
t01=ExpandPartialD[RightPartialD[l].T[l,r]] /.Dot->Times
```

$$-\partial_r A_k \ (\partial_k \partial_l \ A_l) + \partial_r A_k \ (\partial_l \partial_l \ A_k) - \frac{1}{2} g^{lr} \partial_m A_n \ (\partial_l \partial_m \ A_n) + \frac{1}{2} g^{lr} \partial_n A_m \ (\partial_l \partial_m \ A_n) + \frac{1}{2} g^{lr} \partial_m A_n \ (\partial_l \partial_n \ A_m) - \frac{1}{2} g^{lr} \partial_n A_m \ (\partial_l \partial_n \ A_m) - \partial_k A_l \ (\partial_l \partial_r \ A_k) + \partial_l A_k \ (\partial_l \partial_r \ A_k)$$

```
(* --- Using `equation of a motion'... --- *)

s02=QuantumField[Q___,PartialD[LorentzIndex[m_]],Partial
D[LorentzIndex[m_]],A,{n_}]:>QuantumField[Q,PartialD[Lor
entzIndex[m]],PartialD[LorentzIndex[n]],A,{m}]
```

QuantumField (Q___, $\partial_{m\_}$, $\partial_{m\_}$, A, n_) :→QuantumField (Q, ∂_m, ∂_n, A, $\{m\}$)

```
t001=Evaluate[t01/.s02]
```

$$-\partial_r A_k \ (\partial_k \partial_l \ A_l) + \partial_r A_k \ (\partial_l \partial_k \ A_l) - \frac{1}{2} g^{lr} \partial_m A_n \ (\partial_l \partial_m \ A_n) + \frac{1}{2} g^{lr} \partial_n A_m \ (\partial_l \partial_m \ A_n) + \frac{1}{2} g^{lr} \partial_m A_n \ (\partial_l \partial_n \ A_m) - \frac{1}{2} g^{lr} \partial_n A_m \ (\partial_l \partial_n \ A_m) - \partial_k A_l \ (\partial_l \partial_r \ A_k) + \partial_l A_k \ (\partial_l \partial_r \ A_k)$$

```
(* --- Load `derivation rules' and searhing
   procedure --- *)
```

```
<<"/mnt/win_e/My/progs/fc-m-2.m"
```

$$\begin{pmatrix} \text{Simplify001Q} & \text{Simplify001} \\ \text{Rule2Q} & \text{Contract} \\ \text{Rule3Q} & \text{SortingPartials} \end{pmatrix}$$

```
SortingPartials[Simplify001[t001,Contract]]
```

0

```
Search[t001,#==0&]
```

Searching level 1

Searching level 2

Searching level 3

Searching level 4

Searching level 6

Searching level 7

Searching level 8

*** Get answer? (True) ***

0

```
(* --- Symmetrize T[l,r] --- *)

T1[l_,r_]=T[l,r] +
ExpandPartialD[RightPartialD[k].(F[k,l].A[r])]/.Dot->Tim
es
```

$$-\frac{1}{4} g^{\ell r} (\partial_m A_n - \partial_n A_m)^2 + \partial_k A_\ell \, \partial_k A_r - \partial_k A_r \, \partial_\ell A_k + (\partial_\ell A_k - \partial_k A_\ell) \, \partial_r A_k + A_r (\partial_k \partial_k A_\ell) - A_r (\partial_k \partial_\ell A_k)$$

```
(* --- Checking property of T1... --- *)

t02=ExpandPartialD[RightPartialD[l].T1[l,r]]
/.Dot->Times
```

$$\partial_\ell A_r (\partial_k \partial_k A_\ell) - \partial_\ell A_r (\partial_k \partial_\ell A_k) + \partial_k A_r (\partial_k \partial_\ell A_\ell) - \partial_r A_k (\partial_k \partial_\ell A_\ell) + \partial_k A_\ell (\partial_k \partial_\ell A_r) - \partial_\ell A_k (\partial_k \partial_\ell A_r) - \partial_k A_r (\partial_\ell \partial_\ell A_k) + \partial_r A_k (\partial_\ell \partial_\ell A_k) - \frac{1}{2} g^{\ell r} \partial_m A_n (\partial_\ell \partial_m A_n) + \frac{1}{2} g^{\ell r} \partial_n A_m (\partial_\ell \partial_m A_n) + \frac{1}{2} g^{\ell r} \partial_m A_n (\partial_\ell \partial_n A_m) - \frac{1}{2} g^{\ell r} \partial_n A_m (\partial_\ell \partial_n A_m) - \partial_k A_\ell (\partial_\ell \partial_r A_k) + \partial_\ell A_k (\partial_\ell \partial_r A_k) + A_r (\partial_k \partial_k \partial_\ell A_\ell) - A_r (\partial_k \partial_\ell \partial_\ell A_k)$$

```
t002=Evaluate[t02/.s02]
```

$$\partial_k A_r\ (\partial_k \partial_\ell\ A_\ell) - \partial_r A_k\ (\partial_k \partial_\ell\ A_\ell) + \partial_k A_\ell\ (\partial_k \partial_\ell\ A_r) - \partial_\ell A_k\ (\partial_k \partial_\ell\ A_r) - \partial_k A_r\ (\partial_\ell \partial_k\ A_\ell) + \partial_r A_k\ (\partial_\ell \partial_k\ A_\ell) - \frac{1}{2} g^{\ell r}\, \partial_m A_n\ (\partial_\ell \partial_m\ A_n) + \frac{1}{2} g^{\ell r}\, \partial_n A_m\ (\partial_\ell \partial_m\ A_n) + \frac{1}{2} g^{\ell r}\, \partial_m A_n\ (\partial_\ell \partial_n\ A_m) - \frac{1}{2} g^{\ell r}\, \partial_n A_m\ (\partial_\ell \partial_n\ A_m) - \partial_k A_\ell\ (\partial_\ell \partial_r\ A_k) + \partial_\ell A_k\ (\partial_\ell \partial_r\ A_k) + A_r\ (\partial_k \partial_k \partial_\ell\ A_\ell) - A_r\ (\partial_k \partial_\ell \partial_k\ A_\ell)$$

```
SortingPartials[Simplify001[t002,Contract]]
```

0

```
Search[t002,#==0&]
```

Searching level 1

Searching level 2

Searching level 3

Searching level 4

Searching level 6

Searching level 7

Searching level 8

*** Get answer? (True) ***

0

6 Appendix B

This appendix should contain the module which is used in examples from Appendix A. It's available at http://sys01.narod.ru/fc-m-2.m

Searching procedure has name 'Search'. Breadth first search algorithm was used combined with cycling control.

Only three "derivation rules" were created and stored in the variable 'Rules'. User's rules can be added to it simply.