A Framework for Representing and Processing Arbitrary Mathematics

Arnold Neumaier, Peter Schodl
Fakultät für Mathematik, University of Vienna, Nordbergstr. 15, A-1090 Wien, Austria

Introduction

Mathematicians usually have to formulate the same idea multiple times, depending on the recipient:

- in great detail for a student,
- in less detail but a common language for a colleague,
- in a document markup language for a publication,
- in an algebraic modeling language for a numerical solver,
- in a special language and at a high level of detail for a proof checker.

Our vision is that, when represented in an adequate way, the same mathematical content only needs to be communicated to the computer once, and the machine can then extract the information in different formats, depending on the addressee.

We have achieved some promising results representing and reformulating optimization problems into different formal and (controlled) natural languages, and we designed our framework such that it is likely to be able to represent arbitrary mathematics.

The semantic memory

The semantic memory

- is adequate to represent arbitrary mathematics,
- is not a fixed file format, but rather an abstract concept of a data structure,
- stores information in a way akin to the semantic web.

The semantic memory contains relations of the form

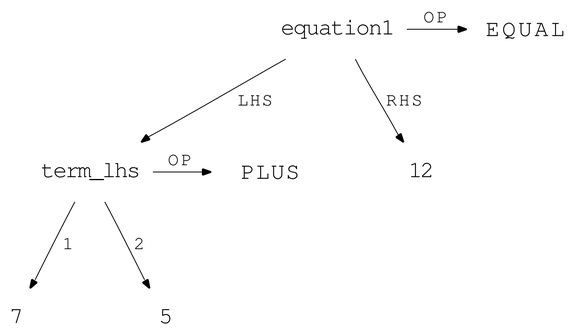
$$\texttt{object1.object2} = \texttt{object3} \tag{1}$$

with the only restriction that

$$\text{if} \quad \texttt{a.b} = \texttt{c} \quad \text{and} \quad \texttt{a.b} = \texttt{d} \quad \text{then} \quad \texttt{c} = \texttt{d}.$$

Such relations can be stored and visualized in a labeled, directed graph, where the relation (1) is represented as an arc from vertex `object1` to vertex `object3`, labeled with `object2`. In this form, the semantic memory can be regarded as a concept map [2].

For example, the information $7 + 5 = 12$ may be represented by the following graph (left) and the corresponding set of relations (right):




```
equation1.RHS = 12
equation1.LHS = term_lhs
equation1.OP = EQUAL
term_lhs.1 = 7
term_lhs.2 = 5
term_lhs.OP = PLUS
```

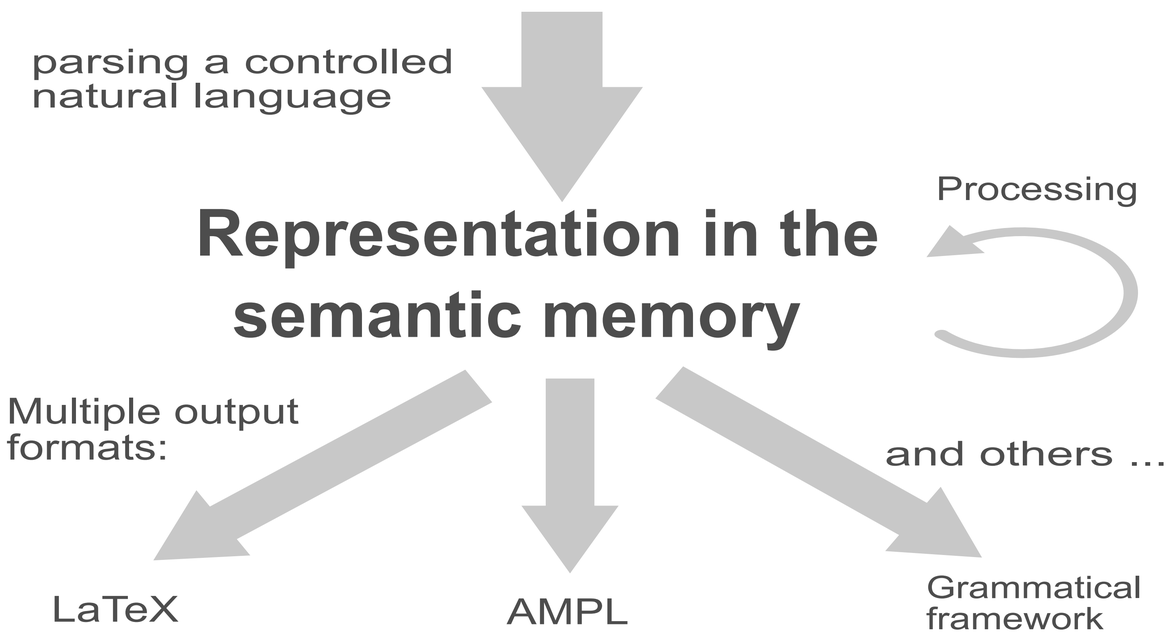
Example: Multidimensional knapsack problem

Let the integer N be the number of contracts, let the integer M be number the of budgets. Let c_j be the contract volume of project j for $j = 1,\ldots,N$, let $A_{i,j}$ be the estimated cost of budget i for project j for $i = 1,\ldots,M$ and $j = 1,\ldots,N$, and let B_i be the available amount of budget i for $i = 1,\ldots,M$. For $j = 1,\ldots,N$, let $x_j = 1$ if project j is selected, and let $x_j = 0$ otherwise.

Problem : Given integers N and M, vector c, matrix A and vector B, find the binary vector x such that

$$\sum_{j=1}^{N} c_j x_j$$

is maximal under the constraint $\sum_{j=1}^{N} A_{i,j} x_j \leq B_i$ for $i = 1,\ldots,M$.



Let integer N be number of contract , let integer M be number of budget, let c_j be contract volume of project j for $j = 1,\ldots,N$, let $A_{i,j}$ be estimated cost of budget i for project j for $i = 1,\ldots,M$ and $j = 1,\ldots,N$, let B_i be available amount of budget i for $i = 1,\ldots,M$ and let $x_j = 1$ if project j is selected, and let $x_j = 0$ otherwise for $j = 1,\ldots,N$.

Problem : Given integer N, integer M , vector c, matrix A and vector B find binary vector x such that

$$\sum_{j=1}^{N} c_j x_j$$

is maximal under the constraint $\sum_{j=1}^{N} A_{i,j} x_j \leq B_i$ for $i = 1,\ldots,M$.

```
param N ;
param M ;
param c{j in 1..N} ;
param A{i in 1..M , j in
1..N} ;
param B{i in 1..M} ;
var x{j in 1..N} binary ;

maximize target :  sum{j
in 1..N}(c[j] * x[j]);
subject to
constraint_3014{i in 1..M}
:  sum{j in 1..N}(A[i , j]
* x[j]) <= B[i];
```

Input: Controlled natural language

There is a consensus among mathematicians and linguists that the communication of mathematics to a computer is much easier than the communication of arbitrary content, because:

- Mathematical discourse has a well-defined domain, is highly structured, and has a relatively small set of discourse relations.
- Mathematicians tend to use a relatively small set of phrases to express their ideas, and there is a standard interpretation for these phrases.
- Mathematicians use words and phrases in a very rigid way, very little variety in time, person, etc.
- A faithful representation of the exact content of an arbitrary (but meaningful) mathematical statement is possible.

Currently, we even simplify further, and define a controlled natural language that is expressive enough to fulfill the needs of mathematicians, while still sounding like natural language.

Results

- Optimization problems can be represented and a description of the problem can then be automatically generated in the algebraic modeling language AMPL and in almost natural language.
- For the management of the semantic memory we have three implementations (Matlab, C/Soprano and JAVA).
- We created an interface that enables proof checking via the Naproche project [1].
- Grammatically correct text-output is generated via an interface to the Grammatical Framework [3].

```
( mkS ( mkAdv for_Prep (mkNP ( mkPN "$ j = 1,
    \\ldots , N $")) ) (impP3 (mkNP ( mkPN "$
    {}{c}_{j} $")) (mkVP (mkNP the_Det ( mkCN
    (mkN2 (mkCN (CompoundCN (mkN "contract")
    (mkN "volume") ) ) ) (mkNP the_Det ( mkCN
    (mkN "project") (mkNP ( mkPN "$ j $")) ) )
    ) ) ) )

( mkS (mkAdv for_Prep (mkS "$ j \\in \\left
    \\{ 1, \\ldots , N \\right\\} $") ) (impP3
    (mkNP ( mkPN "$ {}{c}_{j} $")) (mkVP (
    mkNP the_Det ( mkCN (mkN2 (mkCN (
    CompoundCN (mkN "contract") (mkN "volume")
    ) ) ) (mkNP the_Det ( mkCN (mkN "project
    ") (mkNP ( mkPN "$ j $")) ) ) ) ) ) ) )

( mkS (mkAdv (mkNP and_Conj (mkListNP (mkS "$
    i \\in \\left\\{ 1, \\ldots , M \\right
    \\} $") (mkS "$ j \\in \\left\\{ 1, \\
    ldots , N \\right\\} $") )) ) (impP3 (mkNP
    ( mkPN "$ {}{A}_{i,j} $")) (mkVP (mkNP
    the_Det ( mkCN (mkN2 (mkCN (CompoundCN (
    mkN "estimated") (mkN "cost") ) ) ) (mkNP
    and_Conj (mkListNP (mkNP the_Det ( mkCN (
    mkN "project") (mkNP ( mkPN "$ j $")) ) )
    (mkNP the_Det ( mkCN (mkN "budget") (mkNP
    ( mkPN "$ i $")) ) ) )) ) ) ) ) )

( mkS (mkAdv (mkS "$ i \\in \\left\\{ 1, \\
    ldots , M \\right\\} $") ) (impP3 (mkNP (
    mkPN "$ {}{B}_{i} $")) (mkVP (mkNP the_Det
    ( mkCN (mkN2 (mkCN (mkA "available") (mkN
    "amount") ) ) (mkNP the_Det ( mkCN (mkN "
    budget") (mkNP ( mkPN "$ i $")) ) ) ) ) )
    ) )

( mkS (mkAdv (mkS "$ j \\in \\left\\{ 1, \\
    ldots , N \\right\\} $") ) (mkS (mkConj
    ",") (mkS (mkAdv if_Subj (mkS (mkCl (mkNP
    the_Det ( mkCN (mkN "project") (mkNP (
    mkPN "$ j $")) ) ) (mkVP (mkN "selected")
    ) ) ) ) (impP3 (mkS "$ {}{x}_{j} = 1 $") )
    ) (mkS (mkAdv "otherwise")(impP3 (mkS "$
    {}{x}_{j} = 0 $") ) ) ) )
```

Acknowledgements

Support by the Austrian Science Fund (FWF) under contract number P20631 is gratefully acknowledged.

References

[1] D. Kühlwein, M. Cramer, P. Koepke, and B. Schröder.
The Naproche system.
In *Intelligent Computer Mathematics*. Springer, 2009.

[2] J.D. Novak and A.J. Cañas.
The theory underlying concept maps and how to construct and use them.
In *Technical report IHMC CmapTools 2006-01*. Florida Institute for Human and Machine Cognition, Pensacola Fl, 2006.

[3] A. Ranta.
Grammatical framework.
In *Journal of Functional Programming*, ***14*** *Nr. 2*. Cambridge University Press, 2004.

For more information visit the FMathL homepage: `http://www.mat.univie.ac.at/~neum/FMathL.html`

universität wien