

FMathL

Formal Mathematical Language

Arnold Neumaier, **Peter Schodl**, Kevin Kofler
University of Vienna, Austria

Part of the FMathL project is supported by the FWF

The project

A **modeling language** is an artificial language for the user friendly specification of mathematical problems, with interfaces to the corresponding solvers.

FMathL is intended to be a modeling and documentation language for the working mathematician that

- is based on traditional mathematical syntax,
- allows to express arbitrary mathematics,
- decides automatically which tools to use.

The project

Method: Semantic representation by means of what we call a “semantic matrix” .

We are still far from having a satisfactory solution, but the partial work we have already done is encouraging.

Outline

1. The semantic matrix
2. Representation and testing: the TPTP (Thousands of Problems for Theorem Provers)
3. The OR-Library (a library of optimization problem categories)
4. A semantic Turing machine
5. Parsing natural mathematical language
6. Vision

The semantic matrix

The semantic matrix

Representation via a record-based conception, called the semantic matrix.

Concepts are thought to be the names of rows, columns and entries of a semantic matrix.

The semantic matrix then contain information in the form $\langle \text{concept1} \rangle . \langle \text{concept2} \rangle = \langle \text{concept3} \rangle$.

The semantic matrix: Coding information

For example, if `NN` is the name of the set of positive integers then, in suitable contexts,

```
variable.name=x
```

```
variable.in=NN
```

denotes the assumed knowledge that the variable `x` is a natural number.

In another context,

```
(x.in).NN=true
```

might encode in the semantic matrix the same information.

The semantic matrix: Representation of formulas

record	formula
Ex.nfree=0	
.last=6	
.1.name= x	
.2.op=constant	
.1=\Rz	
.3.op=leq	
.narg=3	
.1=0	
.2=Ex.1	
.3=1	
.4.op=equal	
.1=Ex.1	
.2=2	
.5.op=or	
.1=Ex.3	
.2=Ex.4	
.6.op=set	
.scope=Ex.2	
.formula=Ex.5	
.1=Ex.1	$\{x \in \mathbb{R} \mid 0 \leq x \leq 1 \vee x = 2\}$

The TPTP

To test the universality of the semantic representation, we wrote a translator for the TPTP library (“Thousands of Problems for Theorem Provers”).

A second translator takes the result and produces readable \LaTeX documents.

Example: part of a TPTP file

```
%-----  
% File      : SET016-1 : TPTP v3.5.0. Released v1.0.0.  
% Domain    : Set Theory  
% Problem   : First components of equal ordered pairs are equal  
%-----  
  
cnf(unordered_pair_3,axiom,  
    ( ~ member(X,unordered_pair(Y,Z))  
      | X = Y  
      | X = Z ) ).  
  
cnf(ordered_pair,axiom,  
    ( ordered_pair(X,Y) =  
      unordered_pair( singleton_set(X), unordered_pair(X,Y) ) ) ).
```

Output from the FMathL version of the TPTP

Name: unordered_pair_3
Form: cnf
Domain: SET
Role: axiom
File: SET016-1.p

$$\neg(X \in \{Y, Z\}) \vee X = Y \vee X = Z$$

Name: ordered_pair
Form: cnf
Domain: SET
Role: axiom
File: SET016-1.p

$$(X, Y) = \{\{X\}, \{X, Y\}\}$$

OR-Library:

We just started to take problem classes from the OR-Library (a collection of test data sets for Operations Research problems) and to translate their definitions into a semantic representation.

The OR-Library contains data sets and references to detailed problem descriptions.

Example

Process (at present):

- extract essential parts (by human)
- represent in semantic matrix (by human)
- create readable output (by machine)

Example: The Bin Packing Problem

Description quoted from

E. Falkenauer:

A Hybrid Grouping Genetic Algorithm for Binpacking Networks, Vol. 19 (1989) 379-394

A Bin Packing Problem (BPP) is defined as follows ([Garey and Johnson, 79]): given a finite set O of numbers (the item sizes) and two constants C (the bin's capacity) and N (the number of bins), is it possible to 'pack' all the items into N bins, i.e. does there exist a partition of O into N or less subsets, such that the sum of the elements in any of the subsets doesn't exceed C ?

Internal representation (Extracted by hand)

```
binp.type = optimization problem

binp.problem.generic = bin packing problem
  .attribute = 1-dimensional
  .ref = [GareyJohnson]

binp.instance.nconc = 4
  .1.concept = bin capacity $
    .name = C
    .in = "NN"
  .2.concept = number $ of items
    .name = L
    .in = "NN"
  .3.concept = item size $
    .name = m
    .in = "NN"
  .4.name = M
    .is_a = set
    .elements.in = "NN"
      .concept = item size
    .property.nprop = 1
      .1 = "|M| = L"

binp.feasible.nconc = 1
  .1.concept = packing $
    .name = P
    .property.nprop = 2
      .1 = "P is a partition of M"
      .2 = "forall B in P: sum_{m in B} m <= C"
    .elements.concept = bin $
      .name = B

binp.optimal.nopt = 1
  .1.name = Formulation 1
    .mode = min
    .variables = P
    .objective = N
    .property.nprop = 1
      .1 = "N = |P|"
```

1-dimensional bin packing problem

An instance of a 1-dimensional bin packing problem [GareyJohnson] is defined by two positive integers, the bin capacity C , the number L of items, together with a set M with $|M| = L$. The elements of M are called the item sizes.

Given an instance, a packing P satisfies: P is a partition of M and for all $B \in P$: $\sum_{m \in B} m \leq C$. An element B of

P is called a bin.

Problem:

Find a P that minimizes N under the constraint $N = |P|$.

A semantic Turing machine

For performing algorithms directly on the semantic matrix, we defined a semantic Turing machine (STM). It has a programming language that is

- universal
- semantics-friendly
- very small and transparent
- assembler-like but luxurious
- the semantic matrix acts as memory

An interpreter is implemented in Matlab (1500 lines of code).

A semantic Turing machine

```
program test
process setup
  ^core.x=const 1
  ^core.five=const 5
  goto loop
process loop
  ^x ++
  ^test=(^x==^five)
  if ^test goto end
  goto loop
process end
  stop
start setup
```

A semantic Turing machine

Universal semantic Turing machine (USTM) A single program that simulates the action of an arbitrary STM program on an arbitrary input.

The USTM needs less than 300 lines of STM program code.

Thus only a small amount of code must be checked by hand for correctness before the system can be trusted.

Parsing mathematical text

Much simpler than parsing general natural language:

- very restricted domain
- small set of frequently repeated phrases
- usually exact meaning

Test case: 450 page German lecture notes

“Analysis und Lineare Algebra” (ALA),

containing standard undergraduate mathematics.

ALA lecture notes

Via LaTeXML and automatic postprocessing (formulas replaced by the word FORMULA, etc.), we created a list of about 4000 unique sentence templates.

This was the raw material for

- a lexicon of about 1500 German basic words,
- a simple morphological grammar (later to be replaced by the Grammatical Framework?),
- a sentence grammar with about 1000 production rules.

ALA lecture notes

"defsentence = "v heisst "o "o.
"defsentence = "o "v heissen "o.
"defsentence = "v heisst dann "o.
"defsentence = "v heisst dann "v.
"defsentence = "o ist "o mit "f
"defsentence = "o "o ist "o "o.
"defsentence = "o von "v heissen "o.
"defsentence = fuer "f heisst "v "o.
"defsentence = "v heisst "p, "if "f.
"defsentence = "qt solche "o heisst "o.
"defsentence = wir schreiben "f falls "f.
"defsentence = "v bezeichnet "o aller "v.
"defsentence = "o sind "o der form "f.
"defsentence = man nennt "r "o von "v.
"defsentence = "if "f heisst "v ein "o.
"defsentence = "o der form "f heisst "o.
"defsentence = "v heisst "o 'art "o "v.
"defsentence = ein "o heisst "p, "if "f.
"defsentence = statt "v schreibt man auch "v.
"defsentence = "o wird kurz als "o bezeichnet.
"defsentence = "qt "o mit "o "v ist "o.

After more experience with the OR-Library and ALA:

- We will define a formal subset of mathematical language (FMathL) that can be easily used and parsed automatically.
- All output of our system will be automatically readable.
- An (almost) automatic translation of ALA into english.

Vision: MathResS – An automatic mathematical research student

In a long term perspective we also have far more utopian dreams about an system that could aid mathematicians in doing tedious repetitive work (including checking mathematical text for completeness and correctness) that currently must be done with little computer support.

The amount of additional work is horrible: Freek Wiedijk estimates 40 human hours to produce a verified version of one page of normal \LaTeX .

We shall see how far one can go towards this vision.

Thanks for your attention

More information on FMathL can be found at

<http://www.mat.univie.ac.at/~neum/FMathL.html>