FMathL

Formal Mathematical Language

and how it relates to the Grammatical Framework (GF)

Kevin Kofler
University of Vienna, Austria
Faculty of Mathematics

August 20, 2009

Outline of the Talk

Introduction

Parsing natural mathematical language

FMathL and GF

Conclusion

Introduction

- research project at the University of Vienna, Austria
- partly sponsored by the Austrian Science Fund (FWF)
- Arnold Neumaier
  - full professor for Computational Mathematics
  - head and ideator of the project
- Peter Schodl
  - working primarily on the Semantic Matrix
  - some work on a grammar for German Mathematics
- Kevin Kofler (me)
  - the “Computer Science guy” in the project
  - working on all the parsing
- Flaviu Mărginean
  - specialist for Logic
  - working on reasoning

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.

Goals:

- modeling language for optimization problems (short term)
- reasoning, e.g. checking the correctness of proofs
- (mostly) automatic translation of mathematical texts
- vision: MathResS – automatic math. research system

Internal representation: Semantic Matrix

- concepts . . . names of rows, columns and entries
  - information in the form `<concept1>.<concept2> = <concept3>`
  - matrix interpretation: `<row>.<column> = <entry>`
  - related to triplet representation (RDF): `(<concept1>,<concept2>,<concept3>)`
- semantic Turing machine
  - minimalist computer operating on the semantic matrix
  - assembly-like programming
    - (but friendlier than Turing machine)
  - basis for higher-level FMathL programming language
- still work in progress
- some encouraging partial results
- details out of the scope of this summer school

Parsing natural mathematical language

- 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* by Arnold Neumaier
    - (*Analysis (Calculus) and Linear Algebra*)
    - contains standard undergraduate mathematics.
- list of about 4000 unique sentence templates
    - created via LaTeXML and automatic prostprocessing
        - formulas replaced by the word `FORMULA` etc.
    - this was the raw material for
        - a lexicon of about 1500 German basic words,
        - a simple morphological grammar (to be replaced by GF),
        - a sentence grammar with about 1000 production rules.

Parsing natural mathematical language

```
"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 (a library of problems from Operations Research described in mathematical English) and the *Analysis und Lineare Algebra* (ALA) textbook:

- 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.

My job in the project: work on the parsing part, especially the natural language parsing.

FMathL and GF

- context-free grammars inadequate to parse natural language
    - even for mathematical texts
    - cannot intuitively represent concepts like NP-VP feature agreement
    - need attributed grammars
- GF provides us:
    - a representation for attributed grammars
        - the GF programming language
    - ready to use syntactic grammars
        - the resource grammars
    - one of few projects successfully parsing natural language
    - support for defining an application lexicon and a semantic grammar

Thus we want to use GF in our project.

Open Issues:

- will need to interface GF with an application
  - semantic matrix / semantic turing machine
  - most likely using C++ (not written yet)
  - need to embed the GF interpreter in some way
  - first experiments with the old (GF 2.9) Haskell API wrapped in the Haskell FFI (allowing use from C/C++)
    - not really satisfying
    - obsolete API (no longer present in GF 3.0)
- GF implementation doesn't have some features we'd like
  - incremental changes in the middle of the text
  - incremental changes to the grammar (w/o recompiling)
    - would be important for learning
    - understand the type of a term only once the def. is read
    - but hard to retrofit to an existing infrastructure
  - error correction
- formal verifiability: GF is a complex program

FMathL and GF

Proposed solution: our own parser/interpreter based on the PGF representation (like the Java API)?

- advantages:
  - can use our programming language of choice
    - avoids programming language binding issues
  - can (maybe) implement some desired features
    - error correction
    - reaction to changes in the text
    - (but can probably be retrofitted to the ref. impl. too)
  - could achieve better formal verifiability
- drawbacks:
  - "reinvents the wheel"
  - significant work
  - compatibility concerns
  - might not be able to solve all our issues
    - e.g. incremental additions to the grammar still need at least the PGF file rebuilt

Conclusion

- FMathL is about understanding natural math. language
- research so far only with messy context-free grammars
  - current grammars mostly useless for practical purposes
  - current implementation using Flex and Bison (GLR)
    - proved completely inadequate
    - so far, failing to parse any nontrivial sentence
    - just for testing purposes
- GF is very interesting to us
  - morphological and syntactic analysis
  - will still need semantic analysis and reasoning layers
    - but not started yet, except for lowest level
- but we are still unsure how to best use / interface with it
  - first step (most likely): convert our grammar to GF
- for more information about our project: `http://www.mat.univie.ac.at/~neum/FMathL.html`