Semantic Search for Quantity Expressions

Bachelor Thesis PROPOSAL / FIRST DRAFT[*]

EdN:1

Tom Wiesing
Supervisor: Michael Kohlhase
Jacobs University, Bremen, Germany

May 2, 2015

[2]

EdN:2

Abstract

In this proposal we describe how to introduce units to MathWebSearch. The aim of the project is build an extensible semantics-aware system that searches a corpus of documents for quantity expressions. The project will be based on the existing MathWebSearch system and related technologies. [3][4]

EdN:3
EdN:4

Contents

[*]EdNote: Remove draft status
[2]EdNote: Find a Math Co-supervisor
[3]EdNote: Physics search
[4]EdNote: Re-write abstract

1 Introduction

2 The structure of Mathmatics: Theories, Views and Imports

Before we continue, we want to give an introduction to mathematical theories[5]. EdN:5

2.1 Modeling Mathematics with the help of theories

Theories, in this sense, are a set of symbols. Each of the symbols optionally can optionally have a type and a definition. Within each theory, we can then use these symbols to write down terms (or expressions) within this theory. Types and definitions of these symbols are terms themselves[1]. As a simple example of this, let us consider the theory of semigroups:

Semigroup		
G	:	type
$\circ$	:	$G \to G \to G$
assoc	:	$\mathrm{ded}\,(\forall x \in G.\forall y \in G.\forall z \in G.(x \circ y) \circ z = x \circ (y \circ z))$

In this theory, we define 3 symbols: G, $\circ$ and assoc. In the first line we define a type G. Next we define a function $\circ$ that takes 2 arguments of type G and returns another term of type G. In the last line, we make the statement that associativity holds[6]. EdN:6

2.2 Extending theories using imports

Sometimes we want to extend theories without having to define everything again. For example, we want to say that a Monoid is a semi-group along with an identity element. In the semi-group example above, we have also used terms from other theories to define G as a type.

We can model this concept by using imports. An import from one theory into another makes symbols of the imported theory available in the target theory. We can thus define a Monoid as follows[7]: EdN:7

Monoid		
import Semigroup		
e	:	G
id	:	$\mathrm{ded}\,(\forall x \in G.x \circ e = e \circ x = x)$

2.3 Views as mappings between theories

However imports are not the only way theories can be related. If we have 2 theories, we sometimes want to have a map between them. In addition to the theory of monoids above, we could for example declare the following theory of non-negative integers:

[5] EdNote: Better introduction?

[1] They are not terms over the same theory however.

[6] EdNote: possibly explain / mention Curry–Howard isomorphism

[7] EdNote: Inline import syntax?

Non-negative integers		
$\mathbb{Z}_0^+$	:	type
0	:	$\mathbb{Z}_0^+$
$+$	:	$\mathbb{Z}_0^+ \to \mathbb{Z}_0^+ \to \mathbb{Z}_0^+$
assoc	:	$\text{ded}\left(\forall x \in \mathbb{Z}_0^+.\forall y \in \mathbb{Z}_0^+.\forall z \in \mathbb{Z}_0^+.(x \circ y) \circ z = x \circ (y \circ z)\right)$
id	:	$\text{ded}\left(\forall x \in \mathbb{Z}_0^+.x + 0 = 0 + x = x\right)$

A map from the theory of monoids to the theory of positive integers should map all symbols from the theory of monoids to symbols from the theory of positive integers. Furthermore, such a map should be truth preserving, i. e. if I write down a true statement as a term over the theory of monoids and translate this term, it should still be true in the theory of positive integers. Such a mapping is called a *View* from the theory of monoids to the theory of Positive integers. Such a view ϕ could look as follows:

$$\phi = \left\{ \begin{array}{l} G \mapsto \mathbb{Z}_0^+ \\ e \mapsto 0 \\ \circ \mapsto + \\ \mathsf{assoc} \mapsto \mathsf{assoc} \\ \mathsf{id} \mapsto \mathsf{id} \end{array} \right\}$$

If we take a closer look at this view, we notice that we also have to map the imported symbols. This is needed so that we can translate any term or statement in theory of monoids to a term or statment into the theory of non-negative integers.

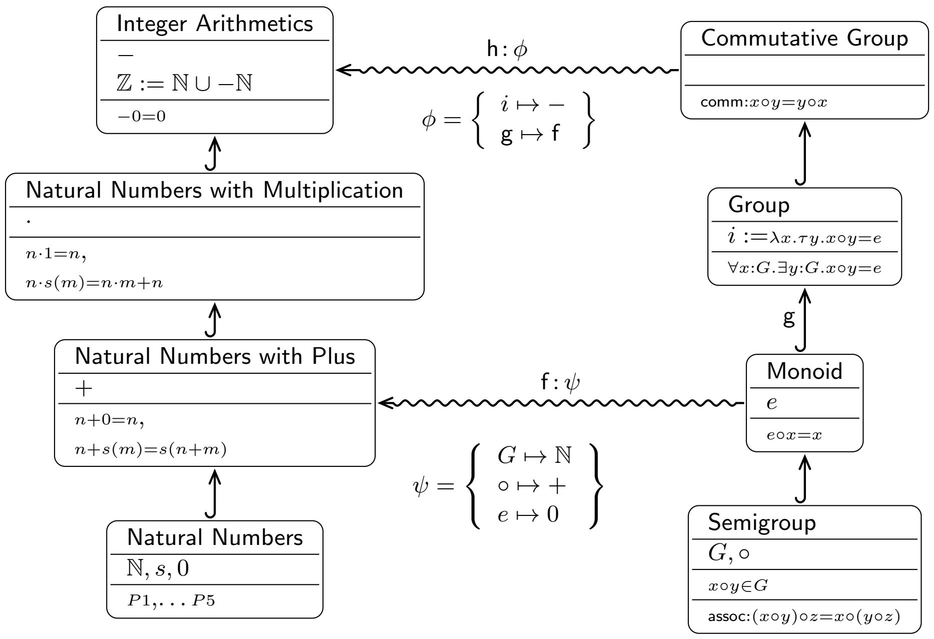
2.4 Building theory graphs

[8] We have seen in the examples above that we can model mathematics with the help of theories, views and imports. To make this structure even more obvious, we can represent it in a graph, a so-called theory graph. We consider the theories as vertices of such a graph and the views and imports as edges. An example can be found in figure 1. EdN:8

2.5 MMT as a concrete implementation

MMT is a **M**odule system for **M**athematical **T**heories [RK13]. In MMT we can represent mathematical theories, views and imports. [9] EdN:9

[8]EdNote: Remove section heading or extend this section.
[9]EdNote: Continue this section



Integer Arithmetics
$-$
$\mathbb{Z} := \mathbb{N} \cup -\mathbb{N}$
$-0=0$
Natural Numbers with Multiplication
$\cdot$
$n\cdot 1=n$,
$n\cdot s(m)=n\cdot m+n$
Natural Numbers with Plus
$+$
$n+0=n$,
$n+s(m)=s(n+m)$
Natural Numbers
$\mathbb{N}, s, 0$
$P1, \ldots P5$
Commutative Group
comm: $x\circ y=y\circ x$
Group
$i := \lambda x.\tau y.x\circ y=e$
$\forall x{:}G.\exists y{:}G.x\circ y=e$
Monoid
e
$e\circ x=x$
Semigroup
$G, \circ$
$x\circ y\in G$
assoc: $(x\circ y)\circ z=x\circ(y\circ z)$
h : ϕ
f : ψ
g
$\phi = \left\{ \begin{array}{c} i \mapsto - \\ \mathsf{g} \mapsto \mathsf{f} \end{array} \right\}$
$\psi = \left\{ \begin{array}{c} G \mapsto \mathbb{N} \\ \circ \mapsto + \\ e \mapsto 0 \end{array} \right\}$

Figure 1: A simple theory graph. Imports are represented as solid edges and views as squiged edges.

3 Modeling Quantity Expressions

3.1 Structure of Quantity Expressions

3.2 Applying theory modeling to Quantity Expressions

4 Making Quantity Expressions searchable

5 Caveats of the current implementation

5.1 Type Equalities

5.2 Unification Queries

6 Future work

6.1 Extension of the theory graph of units

6.2 Integration with MathWebSearch

7 Conclusion

References

[RK13] Florian Rabe and Michael Kohlhase. A scalable module system. *Information & Computation*, 0(230):1–54, 2013.