

# DSLsofMath 2018: Assignment 1

Patrik Jansson and Daniel Schoepe

## 1 DSLsofMath 2018: Assignment 1

In this assignment the focus is on the following three learning outcomes:

- organize areas of mathematics in DSL terms
- develop adequate notation for mathematical concepts
- discuss and compare different software implementations of mathematical concepts

### 1.1 DSLs, sets and von Neumann

In this assignment you will build up a domain specific language (a DSL) for finite sets. The domain you should model is pure set theory where all members are sets.

Define a datatype *SET*  $v$  for the abstract syntax of set expressions with variables of type  $v$  and a datatype *PRED*  $v$  for predicates over pure set expressions.

---

**Part 1.** *SET* should have constructors for

- the *Empty* set
- the one-element set constructor *Singleton*
- *Union*, and *Intersection*
  - you can also try *Powerset*
- set-valued variables ( $\text{Var} :: v \rightarrow \text{SET } v$ )

*PRED* should have constructors for

- the two predicates *Elem*, *Subset*
  - the logical connectives *And*, *Or*, *Implies*, *Not*
-

**Part 2.** A possible semantic domain for pure sets is

**newtype** *Set* = *S* [*Set*]

Implement the evaluation functions

*eval* :: *Eq v* ⇒ *Env v Set* → *SET v* → *Set*  
*check* :: *Eq v* ⇒ *Env v Set* → *PRED v* → *Bool*

**type** *Env var dom* = [(*var*, *dom*)]

Note that the type parameter *v* to *SET* is for the type of variables in the set expressions, not the type of elements of the sets. (You can think of pure set theory as “untyped” or “untyped”.)

---

**Part 3.** The von Neumann encoding of natural numbers as sets is defined recursively as

*vonNeumann* 0 = *Empty*  
*vonNeumann* (*n* + 1) = *Union* (*vonNeumann* *n*)  
                                  (*Singleton* (*vonNeumann* *n*))

Implement *vonNeumann* and explore, explain and implement the following “pseudocode” claims as functions in Haskell:

*claim1* *n1 n2* = {- if (*n1* ≤ *n2*) then (*n1* ⊆ *n2*) -}  
*claim2* *n* = {- *n* = {0, 1, ..., *n* - 1} -}

You need to insert some embeddings and types and you should use the *eval* and *check* functions. (For debugging it is useful to implement a *show* function for *Set* which uses numerals to show the von Neumann naturals.)

---

Admin:

- *Submission*: Assignments are to be submitted via Fire: <https://dslm-lp3-18.frs.cse.chalmers.se>
- *Deadline*: 2018-01-30
- *Grading*: Discussions with each of the teams during the slot 2018-02-05, 10-12 or 15-17

Note: The examination will be in English.