The IDP system reference manual

Johan Wittocx

November 21, 2011

Chapter 1

Comments

Chapter 2

Vocabularies

A vocabulary with name `MyVoc` is declared by

```
vocabulary MyVoc {
   // contents of the vocabulary
}
```

Chapter 3

Theories

Chapter 4

Structures

Chapter 5

Options

Chapter 6

Procedures

6.1 Declaring a procedure

A procedure with name `MyProc` and arguments `A1`, ..., `An` is declared by

```
procedure MyProc(A1,...,An) {
   // contents of the procedure
}
```

Inside a procedure, any chunk of Lua code can be written. For Lua's reference manual, see `http://www.lua.org/manual/5.1/`. In the following, we assume that the reader is familiar with the basic concepts of Lua.

6.2 IDP types

Besides the standard types of variables available in Lua, the following extra types are available in IDP procedures.

sort A set of sorts with the same name. Can be used as a single sort if the set is a singleton.

predicate_symbol A set of predicates with the same name, but possibly with different arities. Can be used as a single predicate if the set is a singleton. If `P` is a predicate_symbol and `n` an integer, then `P/n` returns a predicate_symbol containing all predicates in `P` with arity `n`. If `s1`, ..., `sn` are sorts, then `P[s1,...,sn]` returns a predicate_symbol containing all predicates Q/n in `P`, such that the i'th sort of Q belongs to the set `si`, for $1 \leq i \leq n$.

function_symbol A set of first-order functions with the same name, but possibly with different arities. Can be used as a single first-order function if the set is a singleton. If `F` is a function_symbol and `n` an integer, then `F/n:1` returns a function_symbol containing all function in `F` with arity `n`.

If `s1`, ..., `sn`, `t` are sorts, then `F[s1,...,sn:t]` returns a function_symbol containing all functions G/n in `F`, such that the i'th sort of F belongs to the set `si`, for $1 \leq i \leq n$, and the output sort of G belongs to `t`.

symbol A set of symbols of a vocabulary with the same name. Can be used as if it were a sort, predicate_symbol, or function_symbol.

vocabulary A vocabulary. If `V` is a vocabulary and `s` a string, `V[s]` returns the symbols in `V` with name `s`.

compound A domainelement of the form $F(d_1, \ldots, d_n)$, where F is a first-order function and d_1, ..., d_n are domain elements.

tuple A tuple of domain elements. `T[n]` returns the `n`'th element in tuple `T`.

predicate_table A table of tuples of domain elements.

predicate_interpretation An interpretation for a predicate. If `T` is a predicate_interpreation, then `T.ct`, `T.pt`, `T.cf`, `T.pf` return a predicate_table containing, respectively, the certainly true, possibly true, certainly false, and possibly false tuples in `T`.

function_interpretation An interpretation for a function. `F.graph` returns the predicate_interpretation of the graph associated to the function_interpreation `F`.

structure A first-order structure. To obtain the interpretation of a sort, singleton predicate_symbol, or singleton function_symbol `symb` in structure `S`, write `S[symb]`.

theory A logic theory.

options A set of options.

namespace A namespace.

overloaded An overloaded object.