Supplementary Materials

June 19, 2015

This document describes how to find and use the following artifacts associated with the paper ``System $\mathsf{F}_{\&}$: A Simple Core Language for Extensibility":

- Implementation of the compiler
- Runnable code examples written in our source language
- Mechanized proofs in Coq

Implementation

The implementation of the compiler (in Haskell) is publicly available at: `https://github.com/hkuplg/fcore.git`. To build and install the compiler, simply follow the instructions at `README.md` at the project root. Besides, what may be of special interest to the reader of this paper is the module `Simplify`, which translates $\mathsf{F}_{\&}$ to a variant of System F. It is located (relative to the project root) at `lib/simplify`. Also, the definition of the abstract syntax tree of System $\mathsf{F}_{\&}$ is at `lib/SystemFI.hs`.

Code examples

Two code examples that are used in the paper (Section 3) can be found at: `https://github.com/zhiyuanshi/intersection/tree/master/src`. One is named `ObjectAlgebra.sf`, the other `Visitor.sf`. To run the examples, you need a working installation of the compiler described in the previous section. Here is how you would try out the examples at command line (`f2j` is the name of the compiler; passing the `-r` flag additionally runs the program):

```
$ f2j ObjectAlgebra.sf -r
ObjectAlgebra using [Naive]
Compiling to Java source code ( ./ObjectAlgebra$.java )
7 + 2 = 9
```

Coq proofs

Coq proofs can be found at: `https://github.com/zhiyuanshi/intersection/blob/master/coq/Inter.v`