

December 3, 2009

## Programming Languages and Types

### Homework Assignment 8

Please hand in your homework by email to <mailto:plecture@informatik.uni-marburg.de> until December 10. Please submit your solutions in appropriate file formats.

Consider the substitution-based FAE interpreter in Sec. 6.2 of the textbook. It is available at the course website.

#### H8.1 Single-Step Reduction

Add a function `step` : FAE  $\rightarrow$  FAE to the interpreter, which performs a single reduction step according to the operational semantics of the lambda calculus as defined on slide 8 of the lecture notes. FAE is of course a bit richer than the pure lambda calculus, hence think about an appropriate extension of the operational semantics for the FAE constructs.

#### H8.2 Multi-Step Reduction

Add a function `multistep` : FAE  $\rightarrow$  FAE, which performs computation steps until the expression has been reduced to a value. Confirm with a few tests that `multistep` agrees with `interp`.

#### H8.3 Type Checking

Extend the language with a type checker in the style of the simply-typed lambda calculus. Lambda abstractions should be annotated with types (cf. slides). Think about appropriate typing rules and base types for the arithmetic parts of FAE. Implement the type checker as a function `typecheck` : Context FAE  $\rightarrow$  Type (for an appropriate definition of Context).

#### H8.4 Extensions

Extend the syntax, interpreter, step-relation, and type-checker with the following features (cf. slides):

- pairs
- fixed point operator