Programming Languages and Types

Homework 10

Yi Dai

January 16, 2013

1 Abstract Syntax vs. Concrete Syntax

1. Give the abstract syntax for the language ABE (Arithmetic-Boolean Expressions) presented in the lecture, in BNF and in Scala.
2. Give a concrete syntax other than infix notation for ABE. That is, choose either prefix notation or postfix notation.

2 Inductive Definitions and Rule Induction

1. Give an inductive definition for the ternary relation *Rem*:

   (m, n) *Rem* r if r is the remainder when n is divided by m, where $m \neq 0$.

   You can assume the availability of other arithmetic operations or relations.
2. Use the inductive definition for Rem, prove that

   $$(m, n) \; \mathit{Rem} \; r_1 \text{ and } (m, n) \; \mathit{Rem} \; r_2 \text{ implies } r1 = r2.$$

3 Evaluation Semantics vs. Reduction Semantics

1. Give the evaluation semantics for ABE.

2. Implement both the evaluation semantics and the multi-step reduction semantics for ABE. Show that the two implementations give the same results for the same expressions by a reasonable number of tests.

3. Prove that the evaluation semantics coincides with the multi-step reduction semantics, that is,

$$\forall e \in \mathit{Exp}, e \Longrightarrow v \text{ if and only if } e \longrightarrow^{*} v.$$