Progress and Preservation Proofs for the Expressions "`iszero`" and "`pred`" in the Arith Language

Zayd Hammoudeh
(zayd.hammoudeh@sjsu.edu)

April 13, 2016

Contents

List of Figures

1 Arith Language

Arith is a basic language; its expressions, values, and types are enumerated in figure 1. Arith's small-step, evaluation order semantics are defined in figure 1 while Arith's type rules are enumerated in figure 1.

$e ::=$		*Expressions*
	`true`	Boolean True
	`false`	Boolean False
	`0`	Integer Value 0
	`succ` (e)	Successor Expressions
	`pred` (e)	Predecessor Expressions
	`iszero` (e)	Zero Value Check Expressions
	`if` (e) `then` (e) `else` (e)	Conditional Expressions
$v ::=$		*Values*
	i	integer values
	b	boolean values
$T ::=$		*Types*
	`Bool`	Boolean Type
	`Int`	Integer Type

Figure 1: The Arith language

Evaluation Rules: $\boxed{e \to e'}$

[E-Succ-Ctxt] $$\frac{e \to e'}{\texttt{succ}\ (e) \to \texttt{succ}\ (e)'}$$

[E-Succ] $$\frac{i' = i + 1}{\texttt{succ}\ (i) \to i'}$$

[E-Pred-Ctxt] $$\frac{e \to e'}{\texttt{pred}\ (e) \to \texttt{pred}\ (e)'}$$

[E-Pred] $$\frac{i' = i - 1}{\texttt{pred}\ (i) \to i'}$$

[E-IsZero-Ctxt] $$\frac{e \to e'}{\texttt{iszero}\ (e) \to \texttt{iszero}\ (e')}$$

[E-IsZero-Z] $$\texttt{iszero}\ (\texttt{0}) \to \texttt{true}$$

[E-IsZero-NZ] $$\frac{i \neq 0}{\texttt{iszero}\ (i) \to \texttt{false}}$$

[E-If-Ctxt] $$\frac{e_1 \to e_1'}{\texttt{if}\ (e_1)\ \texttt{then}\ (e_2)\ \texttt{else}\ (e_3) \to \texttt{if}\ (e_1')\ \texttt{then}\ (e_2)\ \texttt{else}\ (e_3)}$$

[E-If-True] $$\texttt{if}\ (\texttt{true})\ \texttt{then}\ (e_2)\ \texttt{else}\ (e_3) \to e_2$$

[E-If-False] $$\texttt{if}\ (\texttt{false})\ \texttt{then}\ (e_2)\ \texttt{else}\ (e_3) \to e_3$$

Figure 2: Small-Step, Evaluation Order Semantics Semantics for the Arith Language

Type Rules: $\boxed{e : T}$

$$[\text{T-True}] \quad \texttt{true} : \texttt{Bool}$$

$$[\text{T-False}] \quad \texttt{false} : \texttt{Bool}$$

$$[\text{T-Int}] \quad i : \texttt{Int}$$

$$[\text{T-Succ}] \quad \frac{e : \texttt{Int}}{\texttt{succ}\ (e) : \texttt{Int}}$$

$$[\text{T-Pred}] \quad \frac{e : \texttt{Int}}{\texttt{pred}\ (e) : \texttt{Int}}$$

$$[\text{T-IsZero}] \quad \frac{e : \texttt{Int}}{\texttt{iszero}\ (e) : \texttt{Bool}}$$

$$[\text{T-If}] \quad \frac{e_1 : \texttt{Bool}, \quad e_2 : T, \quad e_3 : T}{\texttt{if}\ (e_1)\ \texttt{then}\ (e_2)\ \texttt{else}\ (e_3) : T}$$

Figure 3: Type Rules for the Arith Language

2 Progress

In semantics context, "progress" entails that a well-type expression will not "get stuck." Figure 2 shows the formal, theoretical definition of progress.

Given $e : T$, then either:

1. e is a value.
2. There exists an e' such that: $e \rightarrow e'$.

Figure 4: Formal Definition of the Progress Theorem

The following subsections are the formal proofs of progress for the type rules in figure 1.

2.1 Progress Proof for *iszero*