Formal Semantics Review Problems

CSE505

November 7, 2001

In this exercise, you are going to add lists to the call-by-value simply-typed lambda calculus (with booleans). A list is defined recursively as either the value nil or the value (cons v_1 v_2), where v_2 is itself a list. For example, the list consisting of the booleans true, false, and true (in that order) would be represented by the following list value:

(cons true (cons false (cons true nil[Bool])))

The above list value has type Bool List. The value nil is annotated with the element type expected for this occurrence of the empty list. This annotation is necessary so that the type system doesn't have to "guess" the correct type of an empty list. The new primitives head and tail access the first and second components of a cons cell, respectively.

The new cases of the language's syntax are as follows:

e ::= . . .
wrong: T
nil[T]
cons e_1 e_2
head e
tail e

T ::= . . .
T List

The expression denoted wrong: T is a new value that is used to represent errors that will not be prevented by the static type system and can therefore not be considered stuck. In particular, the type system will not ensure that head and tail are always invoked on a non-empty list. The expression wrong is annotated with a type so that the type system does not need to "guess" it (see the typing rule T-Wrong below).

1 Values

Show the new cases of the grammar for values. I've provided one of them.

v ::= . . .
wrong: T

2 Operational Semantics

Show the new rules of the operational semantics. I've provided the rules for head; you should add the other rules.

$$\frac{e \longrightarrow e'}{\text{head } e \longrightarrow \text{head } e'} \text{ (E-Head)}$$

$$\frac{}{\text{head nil}[T] \longrightarrow \text{wrong} : T} \text{ (E-HeadNil)}$$

$$\frac{}{\text{head } (\text{cons } v_1 v_2) \longrightarrow v_1} \text{ (E-HeadCons)}$$

3 Typing Rules

Show the new typing rules. I've provided the rules for wrong and head; you should add the rules for nil, cons and tail.

$$\frac{}{\Gamma \vdash (\text{wrong} : T) : T} \text{ (T-Wrong)}$$

$$\frac{\Gamma \vdash e : T \text{ List}}{\Gamma \vdash \text{head } e : T} \text{ (T-Head)}$$

4 Type Soundness

a. State (no proof necessary) any new parts of the Canonical Forms lemma that are necessary in the proofs below.

b. Add the relevant cases to the proof of the Progress theorem. I've provided the cases for wrong and head; you should add the cases for nil, cons and tail.

   Theorem (Progress): If $\vdash e : T$, then either e is a value or there exists e' such that $e \longrightarrow e'$.

   - Case T-Wrong: Then $e =$ wrong: T, so e is a value.
   - Case T-Head: Then $e =$ (head e_1) and $\vdash e_1 : T$ List. By the inductive hypothesis, e_1 is either a value or there exists e_1' such that $e_1 \longrightarrow e_1'$. In the latter case, by E-Head we have (head e_1) $\longrightarrow$ (head e_1'). In the former case, by the Canonical Forms Lemma (assuming you wrote it properly) we have that e_1 is either nil$[T]$ or has the form (cons v_1 v_2). If e_1 is nil$[T]$, then by E-HeadNil we have $e \longrightarrow$ wrong : T. If e_1 has the form (cons v_1 v_2), then by E-HeadCons we have $e \longrightarrow v_1$.

c. Add the relevant cases to the proof of the Type Preservation theorem. I've provided the cases for wrong and head; you should add the cases for nil, cons and tail.

   Theorem (Type Preservation): If $\Gamma \vdash e : T$ and $e \longrightarrow e'$, then $\Gamma \vdash e' : T$.

   - Case T-Wrong: Then $e =$ wrong: T. By inspection, there is no e' such that wrong: $T \longrightarrow e'$ (assuming you didn't add such a rule), so this case is satisfied trivially.
   - Case T-Head: Then $e =$ (head e_1) and $\Gamma \vdash e_1 : T$ List. We're given that $e \longrightarrow e'$. Case analysis of the last rule used in the derivation of this reduction step:
     - Case E-Head: Then $e' =$ head e_1' and $e_1 \longrightarrow e_1'$. By the inductive hypothesis we have that $\Gamma \vdash e_1' : T$ List. Therefore, by T-Head we have $\Gamma \vdash e' : T$.
     - Case E-HeadNil: Then $e_1 =$ nil$[T']$ and $e' =$ wrong : T'. Since $\Gamma \vdash e_1 : T$ List, by T-Nil (assuming you wrote it properly) we have that $T' = T$. Then by T-Wrong we have $\Gamma \vdash (\text{wrong} : T) : T$.
     - Case E-HeadCons: Then $e_1 = (\text{cons } v_1 v_2)$ and $e' = v_1$. Since $\Gamma \vdash e_1 : T$ List, by T-Cons (assuming you wrote it properly) we have that $\Gamma \vdash v_1 : T$.