CS 231 : Types and Programming Languages
Homework #1

Collaboration : Avneet Oberoi & Ronak Sumbaly

October 6, 2015

Question 2

Theorem

For every term $\mathtt{t}$, either $\mathtt{t}$ is a value or there exists $\mathtt{t'}$ such that $\mathtt{t} \rightarrow \mathtt{t'}$.

Proof

Using **structural induction** on the term $\mathtt{t}$, we proceed as follows,

Induction Hypothesis : For term $\mathtt{t}_0$, such that $\mathtt{t}_0$ is a *subterm* of term $\mathtt{t}$, either $\mathtt{t}_0$ is a value or $\mathtt{t}_0 \rightarrow \mathtt{t}_0'$

Performing case analysis on every term of $\mathtt{t}$,

Case 1. $\mathtt{t} = \mathtt{true}$

Since $\mathtt{true}$ is a value itself defined in the language, $\mathtt{t}$ is also a value and hence the theorem holds.

Case 2. $\mathtt{t} = \mathtt{false}$

Since $\mathtt{false}$ is a value itself defined in the language, $\mathtt{t}$ is also a value and hence the theorem holds - *Symmetric to Case* 1.

Case 3. $\mathtt{t} = \text{if } \mathtt{t} \text{ then } \mathtt{t} \text{ else } \mathtt{t}$

Performing case analysis on the last rule in the derivation. We get,

Case 3.1 $\mathtt{t} =$ E-IfTrue

This form of rule tells us that if $\mathtt{t}_1 = \mathtt{true}$ then $\mathtt{t}_1 \rightarrow \mathtt{t}_2$ from which we can conclude that $\mathtt{t} \rightarrow \mathtt{t}_2$. Hence for term $\mathtt{t}$ there exists $\mathtt{t'}$ where $\mathtt{t'} = \mathtt{t}_2$ where $\mathtt{t} \rightarrow \mathtt{t'}$ for $\mathtt{t}_1 = \mathtt{true}$. Hence the theorem holds.

Case 3.2 $\mathtt{t} =$ E-IfFalse

Symmetric to Case 3.2. This form of rule tells us that if $\mathtt{t}_1 = \mathtt{false}$ then $\mathtt{t}_1 \rightarrow \mathtt{t}_3$ from which we can conclude that $\mathtt{t} \rightarrow \mathtt{t}_3$. Hence for term $\mathtt{t}$ there exists $\mathtt{t'}$ where $\mathtt{t'} = \mathtt{t}_3$ where $\mathtt{t} \rightarrow \mathtt{t'}$ for $\mathtt{t}_1 = \mathtt{false}$. Hence the theorem holds.

Case 3.3 $\mathtt{t} =$ E-If

This form of rule tells us $\mathtt{t}_1 \rightarrow \mathtt{t}_1'$ for some $\mathtt{t}_1'$. Applying the **induction hypothesis** on $\mathtt{t}_1$ we can say that $\mathtt{t}_1'$ is a subterm of $\mathtt{t}$, hence it is either a value or there exist $\mathtt{t}_1''$ where $\mathtt{t}_1' \rightarrow \mathtt{t}_1''$. Hence the theorem holds.

$\therefore$ By structural induction we can conclude that for every term $\mathtt{t}$, either $\mathtt{t}$ is a value or there exists $\mathtt{t'}$ such that $\mathtt{t} \rightarrow \mathtt{t'}$.

Question 3

Evaluation Strategy

For the language of booleans given suppose the then and else branches of an if expression are evaluated (in that order) before the guard is evaluated.

Modified Small Operational Semantics

$$\frac{t_2 \to t_2'}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \ \to \text{if } t_1 \text{ then } t_2' \text{ else } t_3} \qquad \text{(E-THEN)}$$

$$\frac{t_3 \to t_3'}{\text{if } t_1 \text{ then } v_2 \text{ else } t_3 \ \to \text{if } t_1 \text{ then } v_2 \text{ else } t_3'} \qquad \text{(E-ELSE)}$$

$$\frac{t_1 \to t_1'}{\text{if } t_1 \text{ then } v_2 \text{ else } v_3 \ \to \text{if } t_1' \text{ then } v_2 \text{ else } v_3} \qquad \text{(E-GUARD)}$$

$$\frac{}{\text{if } true \text{ then } v_2 \text{ else } v_3 \ \to v_2} \qquad \text{(E-IFTRUE)}$$

$$\frac{}{\text{if } false \text{ then } v_2 \text{ else } v_3 \ \to v_3} \qquad \text{(E-IFFALSE)}$$

Question 4

Theorem

If $\mathtt{t} \to \mathtt{t}'$ and $\mathtt{t} \to \mathtt{t}''$ then $\mathtt{t}' = \mathtt{t}''$.

Proof

Using **induction on derivation** on the term $\mathtt{t}$, we proceed as follows,

Induction Hypothesis : For $\mathtt{t}_0 \to \mathtt{t}_0'$ and $\mathtt{t}_0 \to \mathtt{t}_0''$, such that $\mathtt{t}_0 \to \mathtt{t}_0'$ is a $sub-derivation$ of $\mathtt{t} \to \mathtt{t}'$, then $\mathtt{t}_0' = \mathtt{t}_0''$.

Performing case analysis on the last rule of derivation of $\mathtt{t} \to \mathtt{t}'$,

Case 1. E-IFTRUE

This form of rule tells us if $\mathtt{t}_1 = \mathtt{true}$, then $\mathtt{t} \to \mathtt{t}_2$ i.e $\mathtt{t}' = \mathtt{t}_2$. Applying the last rule of derivation on $\mathtt{t} \to \mathtt{t}''$,

Case 1.1 E-IFTRUE: Again since $\mathtt{t}_1 = \mathtt{true}$ then $\mathtt{t} \to \mathtt{t}_2$ i.e $\mathtt{t}'' = \mathtt{t}_2$. $\therefore \mathtt{t}' = \mathtt{t}''$. Theorem holds.

Case 1.2 E-IFFALSE: For this form $\mathtt{t}_1 = \mathtt{false}$, which is a contradiction to Case 1. where $\mathtt{t}_1 = \mathtt{true}$

Case 1.3 E-IF: From Case 1. we know that $\mathtt{t}_1 = \mathtt{true}$ but for E-IF $\mathtt{t}_1 \to \mathtt{t}_1'$ and since $\mathtt{true}$ is a value and it cannot step. This case is an contradiction.

Case 2. E-IFFALSE

Analogous to Case 1.

Case 3. E-IF

This form of rule tells us that $\texttt{t}_1 \to \texttt{t}_1'$ for some $\texttt{t}_1$. Applying case analysis on the last rule of the derivation for $\texttt{t} \to \texttt{t}''$,

Case 3.1 E-IFTRUE: This form give $\texttt{t}_1 = \texttt{true}$ but it is a contradiction to fact stated in Case 3. that $\texttt{t}_1 \to \texttt{t}_1'$ as $\texttt{true}$ is a value which cannot step. Hence this case is a contradiction.

Case 3.2 E-IFFALSE: Similar to the argument stated in Case 3.1. This case is also a contradiction.

Case 3.3 E-IF: This form states that $\texttt{t}_1 \to \texttt{t}_1''$. Applying **induction hypothesis** we can conclude that since $\texttt{t}_1 \to \texttt{t}_1'$ and $\texttt{t}_1 \to \texttt{t}_1''$ is a subderivation of $\texttt{t} \to \texttt{t}'$ and $\texttt{t} \to \texttt{t}''$, $\therefore \texttt{t}' = \texttt{t}''$.

$\therefore$ By induction on derivation we can conclude that if $\texttt{t} \to \texttt{t}'$ and $\texttt{t} \to \texttt{t}''$ then $\texttt{t}' = \texttt{t}''$.

Question 5

BNF Grammar

BNF grammar for a new metavariable `s` that characterizes the stuck expressions for **Question 1**

$\langle t \rangle$:= `true` | `false` | $\langle n \rangle$ | $\langle t \rangle$ `+` $\langle t \rangle$ | $\langle t \rangle$ `>` $\langle t \rangle$ | `if` $\langle t \rangle$ `then` $\langle t \rangle$ `else` $\langle t \rangle$ |

$\langle badbool \rangle$:= `true` | `false`

$\langle n \rangle$:= `integers`

$\langle s \rangle$:= $\langle badbool \rangle$ `+` $\langle badbool \rangle$ | $\langle badbool \rangle$ `+` $\langle n \rangle$ | $\langle n \rangle$ `+` $\langle badbool \rangle$
| $\langle badbool \rangle$ `>` $\langle badbool \rangle$ | $\langle badbool \rangle$ `>` $\langle n \rangle$ | $\langle n \rangle$ `>` $\langle badbool \rangle$
| `if` $\langle n \rangle$ `then` $\langle t \rangle$ `else` $\langle t \rangle$

Question 6

(a.) *Stuck in Original Semantics but not in Modified Semantics* :

For the term {if $\texttt{t}_1$ then $\texttt{t}_2$ else $\texttt{t}_3$}, in the original semantics if $\texttt{t}_1 = n$ then the term would be stuck, whereas in the modified semantics since the guard is evaluated after the *then* and *else* condition, hence it would not be stuck but eventually stuck. For example,

$$\text{if } \texttt{7} \text{ then } (7 > 9) \text{ else } \texttt{true}$$

(b.) *Stuck in Modified Semantics but not in Original Semantics* :

For the term {if $\texttt{t}_1$ then $\texttt{t}_2$ else $\texttt{t}_3$}, in the modified semantics since the guard is evaluated at the end if either of $\texttt{t}_2$ or $\texttt{t}_3$ have an invalid output then it is stuck whereas in the original semantics since the guard is evaluated first the term would be eventually stuck. For example,

$$\text{if } \texttt{false} \text{ then } \texttt{false} \text{ else } \texttt{false} < 7$$

Question 7

(a.) *Eventually Stuck in Original Semantics but not in Modified Semantics* : No terms

(b.) *Eventually Stuck in Modified Semantics but not in Original Semantics* :

$$\text{if } \texttt{true} \text{ then } 7 > 9 \text{ else } \texttt{false} < 10$$