University of Massachusetts Lowell — Comp 3010: Organization of Programming Languages

Assignment 4

Name: **[[*Harishwar Reddy Erri*]]**

UML ID: **[[*02148304*]]**

Collaborators: **[[*NONE*]]**

Make sure that the remaining pages of this assignment do not contain any identifying information.

1 Equivalence of Semantics (40 points)

Recall the grammar for the calculator language used from HW2 and HW3. In this question, you will prove the equivalence of large-step and multi-step semantics.

$$\begin{array}{lll} n & \in & \mathbb{Z} \\ a & ::= & n \mid a_1 + a_2 \mid a_1 \times a_2 \\ b & ::= & \textbf{true} \mid \textbf{false} \mid a = a \mid a \neq a \\ & & \mid a \leq a \mid a > a \mid \neg b \mid b \&\& b \\ v & ::= & n \mid \textbf{true} \mid \textbf{false} \end{array}$$

(a) Prove the following theorem using induction.

Theorem (Equivalence of Semantics for Arithmetic Expressions)**.** *For all expressions a, integers n, we have:*

$$a \Downarrow n \iff a \longrightarrow^* n.$$

Part (a) - Arithmetic Expressions

(i) Set to induct on in the forward direction:
The set of all arithmetic expressions a.

(ii) Property to induct on in the forward direction:
$P(a) := \forall n \in \mathbb{Z}, a \Downarrow n \implies a \to^* n$

(iii) Inductive reasoning principle:
Base case: Show $P(n)$ holds for all integer literals n.
Inductive step: For $\circ \in \{+, \times\}$, if $P(a_1)$ and $P(a_2)$ hold, then show $P(a_1 \circ a_2)$ holds.

(iv) Proof:

Forward direction: $a \Downarrow n \implies a \to^* n$
We proceed by structural induction on a.

Base case: $a = n$ (integer literal)
If $n \Downarrow n$, then trivially $n \to^* n$ in zero steps.

Inductive step: Assume $P(a_1)$ and $P(a_2)$ hold.

Case $a = a_1 + a_2$:
Given $a_1 + a_2 \Downarrow n$, by the evaluation rule, we know:
$a_1 \Downarrow n_1$, $a_2 \Downarrow n_2$, and $n = n_1 + n_2$
By induction hypothesis: $a_1 \to^* n_1$ and $a_2 \to^* n_2$
Therefore: $a_1 + a_2 \to^* n_1 + a_2 \to^* n_1 + n_2 = n$

Case $a = a_1 \times a_2$: Similar to the addition case.

Reverse direction: $a \to^* n \implies a \Downarrow n$
We proceed by induction on the number of steps in $\to^*$.

Base case: Zero steps
If $a \to^* n$ in zero steps, then $a = n$, and trivially $n \Downarrow n$.

Inductive step:
Assume for any k-step reduction $a \to^k n \implies a \Downarrow n$.
Consider a $(k+1)$-step reduction $a \to a' \to^k n$.

Case $a = a_1 + a_2$:
$a_1 + a_2 \to a_1' + a_2 \to^k n$ where $a_1 \to a_1'$
By induction hypothesis, $a_1' + a_2 \Downarrow n$
This implies $a_1' \Downarrow n_1$, $a_2 \Downarrow n_2$, and $n = n_1 + n_2$

Since $a_1 \to a_1'$, by induction hypothesis $a_1 \Downarrow n_1$
Therefore, $a_1 + a_2 \Downarrow n$

Case $a = a_1 \times a_2$: Similar to the addition case.

(i) Provide the actual proof for both forward and reverse directions.

Proof. □

(b) Prove the following theorem using induction.

Theorem (Equivalence of Semantics for Boolean Expressions)**.** *For all expressions a, values $v \in \{$**true**, **false**$\}$, we have:*

$$b \Downarrow v \iff b \longrightarrow^* v.$$

(i) What is the set that you will induct on in the forward direction?

(ii) What is the property that you will induct on in the forward direction? Formally state the property.

(iii) Write the inductive reasoning principle for the forward direction.

(iv) Provide the proof for both forward and reverse directions. Note that you have to handle the proof for all cases. You cannot state "same as above" or "similar to the case".

Hint: When you handle cases that involve comparison between arithmetic expressions, you may need to invoke the lemmas proved in the previous part of the question.

Proof. **Part (b) - Boolean Expressions**

(i) Set to induct on in the forward direction:
The set of all boolean expressions b.
(ii) Property to induct on in the forward direction:
$P(b) := \forall v \in \{\text{true}, \text{false}\}, b \Downarrow v \implies b \to^* v$
(iii) Inductive reasoning principle:
Base cases: Show $P(\text{true})$ and $P(\text{false})$ hold.
Inductive steps:
- For $\circ \in \{=, \neq, \leq, \rangle\}$, if $P(a_1)$ and $P(a_2)$ hold, then show $P(a_1 \circ a_2)$ holds.
- If $P(b)$ holds, then show $P(\neg b)$ holds.
- If $P(b_1)$ and $P(b_2)$ hold, then show $P(b_1 \&\& b_2)$ holds.
(iv) Proof:
Forward direction: $b \Downarrow v \implies b \to^* v$
We proceed by structural induction on b.
Base cases:
- $b =$ true: If true $\Downarrow$ true, then trivially true $\to^*$ true in zero steps.
- $b =$ false: Similar to the true case.
Inductive steps:
Case $b = a_1 = a_2$:
Given $a_1 = a_2 \Downarrow v$, by the evaluation rule:
$a_1 \Downarrow n_1$, $a_2 \Downarrow n_2$, and $v = (n_1 = n_2)$
By the arithmetic expression equivalence theorem:
$a_1 \to^* n_1$ and $a_2 \to^* n_2$
Therefore: $a_1 = a_2 \to^* n_1 = a_2 \to^* n_1 = n_2 \to v$
Cases for $\neq, \leq, \rangle$: Similar to the equality case.

Case $b = \neg b'$:
Given $\neg b' \Downarrow v$, by the evaluation rule:
$b' \Downarrow v'$ and $v = \neg v'$
By induction hypothesis: $b' \to^* v'$
Therefore: $\neg b' \to^* \neg v' = v$

Case $b = b_1 \&\& b_2$:
Given $b_1 \&\& b_2 \Downarrow v$, by the evaluation rule:
$b_1 \Downarrow v_1$, and if $v_1 =$ true then $b_2 \Downarrow v$, else $v =$ false
By induction hypothesis: $b_1 \to^* v_1$
If $v_1 =$ true, then $b_2 \to^* v$ (by induction hypothesis)
Therefore: $b_1 \&\& b_2 \to^*$ true$\&\& b_2 \to^*$ true$\&\& v \to v$
If $v_1 =$ false, then:
$b_1 \&\& b_2 \to^*$ false$\&\& b_2 \to$ false

Reverse direction: $b \to^* v \implies b \Downarrow v$
The proof follows a similar structure to the arithmetic expressions case,
using induction on the number of steps in $\to^*$.
Handle boolean operations and comparisons using case analysis similar to the forward direction.