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

Assignment 4

Name: **[[*Ashish Kosana*]]**

UML ID: **[[*02148256*]]**

Collaborators: [[*NONE*]]

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.

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

(a) Prove the following theorem using induction.

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

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

1.1 Arithmetic Expressions

First, let's recall the inference rules for arithmetic expressions:

$$\frac{}{n \Downarrow n} \quad \text{(Const)}$$

$$\frac{a_1 \Downarrow n_1 \quad a_2 \Downarrow n_2}{a_1 + a_2 \Downarrow n_1 + n_2} \quad \text{(Add)}$$

$$\frac{a_1 \Downarrow n_1 \quad a_2 \Downarrow n_2}{a_1 \times a_2 \Downarrow n_1 \times n_2} \quad \text{(Mult)}$$

For small-step semantics:

$$\frac{a_1 \to a_1'}{a_1 + a_2 \to a_1' + a_2} \quad \text{(Add1)}$$

$$\frac{a_2 \to a_2'}{n_1 + a_2 \to n_1 + a_2'} \quad \text{(Add2)}$$

$$\frac{}{n_1 + n_2 \to n} \quad \text{where } n = n_1 + n_2 \quad \text{(Add3)}$$

$$\frac{a_1 \to a_1'}{a_1 \times a_2 \to a_1' \times a_2} \quad \text{(Mult1)}$$

$$\frac{a_2 \to a_2'}{n_1 \times a_2 \to n_1 \times a_2'} \quad \text{(Mult2)}$$

$$\frac{}{n_1 \times n_2 \to n} \quad \text{where } n = n_1 \times n_2 \quad \text{(Mult3)}$$

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

$a \Downarrow n \Leftrightarrow a \to^* n$

1.1.1 Forward Direction

(i) We will induct on the structure of the arithmetic expression a.

(ii) The property $P(a)$ we will induct on is:

For all n, if $a \Downarrow n$, then $a \to^* n$.

(iii) Inductive reasoning principle:

To prove $\forall a.P(a)$, it suffices to show:

- Base case: $P(n)$ holds for all integer constants n.
- Inductive cases:
  - If $P(a_1)$ and $P(a_2)$ hold, then $P(a_1 + a_2)$ holds.
  - If $P(a_1)$ and $P(a_2)$ hold, then $P(a_1 \times a_2)$ holds.

(iv) Proof:

Proof. Base case: For any integer constant n, we have $n \Downarrow n$ and $n \to^* n$ (in zero steps), so $P(n)$ holds.

Inductive case for addition: Assume $P(a_1)$ and $P(a_2)$ hold. We need to prove $P(a_1 + a_2)$. Suppose $a_1 + a_2 \Downarrow n$. By the (Add) rule, we know: $a_1 \Downarrow n_1$, $a_2 \Downarrow n_2$, and $n = n_1 + n_2$. By the induction hypothesis, $a_1 \to^* n_1$ and $a_2 \to^* n_2$. Therefore, $a_1 + a_2 \to^* n_1 + a_2 \to^* n_1 + n_2 \to n$. Thus, $a_1 + a_2 \to^* n$, proving $P(a_1 + a_2)$.

Inductive case for multiplication: The proof is similar to the addition case, using the (Mult) rule instead.

By the principle of induction, $\forall a.P(a)$ holds. □

1.1.2 Reverse Direction

We prove this direction by induction on the number of small-step reductions.

Proof. Base case: If $a \to^* n$ in zero steps, then $a = n$, and we have $n \Downarrow n$ by the (Const) rule.

Inductive step: Assume the statement holds for all expressions that reduce to a value in k or fewer steps. Consider $a \to^* n$ in $k + 1$ steps. Let $a \to a' \to^k n$. By the induction hypothesis, $a' \Downarrow n$.

We now consider cases for the first reduction $a \to a'$:

1) If $a = a_1 + a_2$ and $a' = a_1' + a_2$ where $a_1 \to a_1'$: By the induction hypothesis, $a' = a_1' + a_2 \Downarrow n$. This means $a_1' \Downarrow n_1$, $a_2 \Downarrow n_2$, and $n = n_1 + n_2$. Since $a_1 \to a_1'$ and $a_1' \Downarrow n_1$, by the induction hypothesis, $a_1 \Downarrow n_1$. Therefore, by the (Add) rule, $a = a_1 + a_2 \Downarrow n$.

2) The cases for (Add2), (Mult1), and (Mult2) are similar.

3) If $a = n_1 + n_2$ and $a' = n$ where $n = n_1 + n_2$: We have $n_1 \Downarrow n_1$ and $n_2 \Downarrow n_2$ by (Const), so $a = n_1 + n_2 \Downarrow n$ by (Add).

4) The case for (Mult3) is similar to (Add3).

By induction, the statement holds for all $a \to^* n$. □

(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. □

1.2 Boolean Expressions

First, let's recall the inference rules for boolean expressions:

$$\frac{}{true \Downarrow true}\ \text{(True)} \qquad \frac{}{false \Downarrow false}\ \text{(False)}$$

$$\frac{a_1 \Downarrow n_1 \quad a_2 \Downarrow n_2}{a_1 = a_2 \Downarrow true}\ \text{if } n_1 = n_2 \quad \text{(Eq-True)}$$

$$\frac{a_1 \Downarrow n_1 \quad a_2 \Downarrow n_2}{a_1 = a_2 \Downarrow false}\ \text{if } n_1 \neq n_2 \quad \text{(Eq-False)}$$

$$\frac{a_1 \Downarrow n_1 \quad a_2 \Downarrow n_2}{a_1 \neq a_2 \Downarrow true}\ \text{if } n_1 \neq n_2 \quad \text{(Neq-True)}$$

$$\frac{a_1 \Downarrow n_1 \quad a_2 \Downarrow n_2}{a_1 \neq a_2 \Downarrow false}\ \text{if } n_1 = n_2 \quad \text{(Neq-False)}$$

$$\frac{a_1 \Downarrow n_1 \quad a_2 \Downarrow n_2}{a_1 \leq a_2 \Downarrow true}\ \text{if } n_1 \leq n_2 \quad \text{(Leq-True)}$$

$$\frac{a_1 \Downarrow n_1 \quad a_2 \Downarrow n_2}{a_1 \leq a_2 \Downarrow false}\ \text{if } n_1 \rangle n_2 \quad \text{(Leq-False)}$$

$$\frac{a_1 \Downarrow n_1 \quad a_2 \Downarrow n_2}{a_1 \rangle a_2 \Downarrow true}\ \text{if } n_1 \rangle n_2 \quad \text{(Gt-True)}$$

$$\frac{a_1 \Downarrow n_1 \quad a_2 \Downarrow n_2}{a_1 \rangle a_2 \Downarrow false}\ \text{if } n_1 \leq n_2 \quad \text{(Gt-False)}$$

$$\frac{b \Downarrow true}{\neg b \Downarrow false}\ \text{(Not-True)} \qquad \frac{b \Downarrow false}{\neg b \Downarrow true}\ \text{(Not-False)}$$

$$\frac{b_1 \Downarrow true \quad b_2 \Downarrow true}{b_1 \&\& b_2 \Downarrow true}\ \text{(And-True)}$$

$$\frac{b_1 \Downarrow false}{b_1 \&\& b_2 \Downarrow false}\ \text{(And-False1)} \qquad \frac{b_1 \Downarrow true \quad b_2 \Downarrow false}{b_1 \&\& b_2 \Downarrow false}\ \text{(And-False2)}$$

For small-step semantics (only showing rules different from arithmetic expressions):

$$\frac{a_1 \to a_1'}{a_1 = a_2 \to a_1' = a_2} \quad \text{(Eq1)}$$

$$\frac{a_2 \to a_2'}{n_1 = a_2 \to n_1 = a_2'} \quad \text{(Eq2)}$$

$$\overline{n_1 = n_2 \to true} \quad \text{if } n_1 = n_2 \quad \text{(Eq3)}$$

$$\overline{n_1 = n_2 \to false} \quad \text{if } n_1 \neq n_2 \quad \text{(Eq4)}$$

(Similar rules for $\neq, \leq, \rangle$)

$$\frac{b \to b'}{\neg b \to \neg b'} \quad \text{(Not1)}$$

$$\overline{\neg true \to false} \quad \text{(Not2)} \qquad \overline{\neg false \to true} \quad \text{(Not3)}$$

$$\frac{b_1 \to b_1'}{b_1 \&\& b_2 \to b_1' \&\& b_2} \quad \text{(And1)}$$

$$\overline{false \&\& b_2 \to false} \quad \text{(And2)}$$

$$\overline{true \&\& b_2 \to b_2} \quad \text{(And3)}$$

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

1.2.1

(i) We will induct on the structure of the boolean expression b.
(ii) The property $P(b)$ we will induct on is:
For all $v \in \{true, false\}$, if $b \Downarrow v$, then $b \to^* v$.
(iii) Inductive reasoning principle:
To prove $\forall b.P(b)$, it suffices to show:

- Base cases: $P(true)$ and $P(false)$ hold.
- Inductive cases:
  - If $P(a_1)$ and $P(a_2)$ hold, then $P(a_1 = a_2)$, $P(a_1 \neq a_2)$, $P(a_1 \leq a_2)$, and $P(a_1 \rangle a_2)$ hold.
  - If $P(b)$ holds, then $P(\neg b)$ holds.
  - If $P(b_1)$ and $P(b_2)$ hold, then $P(b_1 \&\& b_2)$ holds.

(iv) Proof:

Proof. (Forward Direction)
Base cases: For $true$ and $false$, we have $true \Downarrow true$ and $true \to^* true$ (in zero steps), and similarly for $false$. So $P(true)$ and $P(false)$ hold.
Inductive case for equality: Assume $P(a_1)$ and $P(a_2)$ hold. We need to prove $P(a_1 = a_2)$. Suppose $a_1 = a_2 \Downarrow v$. By the (Eq-True) or (Eq-False) rule, we know: $a_1 \Downarrow n_1$, $a_2 \Downarrow n_2$, and $v = true$ if $n_1 = n_2$, $v = false$ otherwise. By the induction hypothesis and the theorem for arithmetic

expressions, $a_1 \to^* n_1$ and $a_2 \to^* n_2$. Therefore, $a_1 = a_2 \to^* n_1 = n_2 \to v$. Thus, $a_1 = a_2 \to^* v$, proving $P(a_1 = a_2)$.

(Similar proofs for $\neq$, $\leq$, and $\rangle$)

Inductive case for negation: Assume $P(b)$ holds. We need to prove $P(\neg b)$. Suppose $\neg b \Downarrow v$. By the (Not-True) or (Not-False) rule, we know: $b \Downarrow \neg v$. By the induction hypothesis, $b \to^* \neg v$. Therefore, $\neg b \to^* \neg(\neg v) \to v$. Thus, $\neg b \to^* v$, proving $P(\neg b)$.

Inductive case for conjunction: Assume $P(b_1)$ and $P(b_2)$ hold. We need to prove $P(b_1 \&\& b_2)$. Suppose $b_1 \&\& b_2 \Downarrow v$. We have three cases: 1) If $b_1 \Downarrow false$, then $v = false$. By IH, $b_1 \to^* false$, so $b_1 \&\& b_2 \to^* false \&\& b_2 \to false$. 2) If $b_1 \Downarrow true$ and $b_2 \Downarrow false$, then $v = false$. By IH, $b_1 \to^* true$ and $b_2 \to^* false$, so $b_1 \&\& b_2 \to^* true \&\& b_2 \to^* false$. 3) If $b_1 \Downarrow true$ and $b_2 \Downarrow true$, then $v = true$. By IH, $b_1 \to^* true$ and $b_2 \to^* true$, so $b_1 \&\& b_2 \to^* true \&\& b_2 \to^* true$. In all cases, $b_1 \&\& b_2 \to^* v$, proving $P(b_1 \&\& b_2)$.

(Reverse Direction)

We will prove by induction on the number of small-step reductions.

Base case: If $b \to^* v$ in zero steps, then $b = v$, and we have $b \Downarrow v$ by the (True) or (False) rule.

Inductive step: Assume that for any boolean expression b' and value v', if $b' \to^n v'$ then $b' \Downarrow v'$. We need to prove that if $b \to^{n+1} v$, then $b \Downarrow v$.

If $b \to^{n+1} v$, then there exists some b' such that $b \to b'$ and $b' \to^n v$. By the induction hypothesis, $b' \Downarrow v$. We need to show that $b \Downarrow v$.

We proceed by cases on the structure of b:

1) If b is $true$ or $false$, it cannot reduce, so this case is impossible.

2) If b is $a_1 = a_2$, then b' must be of the form $a_1' = a_2$ or $n_1 = a_2'$ or $n_1 = n_2$. In the first two cases, we can apply the induction hypothesis to a_1 or a_2, and then use the (Eq-True) or (Eq-False) rule. In the last case, we directly apply (Eq-True) or (Eq-False).

3) Similar arguments apply for $\neq$, $\leq$, and $\rangle$.

4) If b is $\neg b_1$, then b' is either $\neg b_1'$ or a boolean value. In the first case, we apply the induction hypothesis to b_1 and then use (Not-True) or (Not-False). In the second case, we directly apply (Not-True) or (Not-False).

5) If b is $b_1 \&\& b_2$, then b' is either $b_1' \&\& b_2$, or b_2 (if $b_1 = true$), or a boolean value (if $b_1 = false$ or b_2 is a value). We apply the induction hypothesis and then use the appropriate (And-True), (And-False1), or (And-False2) rule.

In all cases, we can derive $b \Downarrow v$, completing the proof. $\square$