Exercise session 2
– Structural induction –

Exercise 6. Unfolding Loops in **IMP**

On the lecture we completed Case "⇒" of the equivalence proof of

$$\langle \texttt{while } b \texttt{ do } s \texttt{ end}, \sigma\rangle \to \sigma'' \Leftrightarrow \langle \texttt{if } b \texttt{ then } s\texttt{; while } b \texttt{ do } s \texttt{ end}, \sigma\rangle \to \sigma''$$

Prove the other direction!

Exercise 7. Free variables

Let σ and σ' be two states satisfying that $\sigma(\mathtt{x}) = \sigma'(\mathtt{x})$ for all $\mathtt{x}$ in $\mathrm{FV}(e)$. Using structural induction prove that $\mathcal{A}[\![e]\!]\sigma = \mathcal{A}[\![e]\!]\sigma'$.

Substitution
The notation $e[y \mapsto e']$ means replacing each occurence of a variable y in an arithmetic expression e with another arithmetic expression e'. The formal definition is as follows:

$$\begin{aligned}
i[y \mapsto e'] &= i \\
x[y \mapsto e'] &= \begin{cases} e' & \text{if } x = y \\ x & \text{if } x \neq y \end{cases} \\
(e_1 \ op \ e_2)[y \mapsto e'] &= (e_1[y \mapsto e']) \ op \ (e_2[y \mapsto e'])
\end{aligned}$$

For example $(x{+}1)[x \mapsto 3] = 3{+}1$ and $(x{+}y{*}x)[x \mapsto y{-}5] = (y{-}5){+}y{*}(y{-}5)$.

Exercise 8. Substitution

Prove that $\mathcal{A}[\![e[y \mapsto e_0]]\!]\sigma = \mathcal{A}[\![e]\!](\sigma[y \mapsto \mathcal{A}[\![e_0]\!]\sigma])$ for all states σ. What is the intuition of the theorem?

Exercise 9. Extending **Bexp**

The syntactic category **Bexp'** is defined as the following extension of **Bexp**:

$$
\begin{array}{rcl}
b & ::= & \texttt{true} \mid \texttt{false} \mid e_1 = e_2 \mid e_1 \,\#\, e_2 \mid e_1 <= e_2 \mid e_1 >= e_2 \\
& \mid & e_1 < e_2 \mid e_1 > e_2 \mid \texttt{not}\ b \mid b_1\ \texttt{and}\ b_2 \mid b_1\ \texttt{or}\ b_2 \\
& \mid & b_1\ \texttt{implies}\ b_2 \mid b_1\ \texttt{equivalent}\ b_2
\end{array}
$$

Give a compositional extension of the semantic function $\mathcal{B}$.

Exercise 10. Equivalence of **Bexp** and **Bexp'**

Two boolean expressions b_1 and b_2 are *equivalent* if for all states σ,

$$\mathcal{B}[\![b_1]\!]\sigma = \mathcal{B}[\![b_2]\!]\sigma$$

Show that for each b' of **Bexp'** there exists a boolean expression b of **Bexp** such that b' and b are equivalent.

Solutions

Exercise 6. Unfolding Loops in **IMP**

On the lecture we used the following proof idea:

- Consider the derivation tree for one transition
- Show that there is a derivation tree for the other transition

In Case "$\Leftarrow$" we assume that there is a derivation tree T for

$$\langle \texttt{if } b \texttt{ then } s\texttt{;while } b \texttt{ do } s \texttt{ end else skip end}, \sigma\rangle \to \sigma'' \qquad (**)$$

and have to show that there is a derivation tree for

$$\langle \texttt{while } b \texttt{ do } s \texttt{ end}, \sigma\rangle \to \sigma''. \qquad (*)$$

The last rule application in the construction of T can only be one of the rules for `if`.

For the case $\mathcal{B}[\![b]\!]\sigma = tt$, T was constructed using the `if` rule with condition $\mathcal{B}[\![b]\!]\sigma = tt$. Thus we get a derivation tree T_1 with root

$$\langle s\texttt{;while } b \texttt{ do } s \texttt{ end}, \sigma\rangle \to \sigma''$$

The statement has the form $s_1; s_2$ thus the only rule that could give this is the sequential composition rule. Therefore there are derivation trees T_2 and T_3 for $\langle s, \sigma\rangle \to \sigma'$ and $\langle \texttt{while } b \texttt{ do } s \texttt{ end}, \sigma'\rangle \to \sigma''$, respectively, for some state σ'.
Now using the fact that $\mathcal{B}[\![b]\!]\sigma = tt$, we can construct (*) using the `while` rule:

$$\frac{T_2, T_3}{\langle \texttt{while } b \texttt{ do } s \texttt{ end}, \sigma\rangle \to \sigma''} \quad \text{if } \mathcal{B}[\![b]\!]\sigma = tt$$

In the case when $\mathcal{B}[\![b]\!]\sigma = ff$, T was constructed using the `if` rule with condition $\mathcal{B}[\![b]\!]\sigma = ff$. Thus we get a derivation tree

$$\langle \texttt{skip}, \sigma\rangle \to \sigma''$$

and according to the `skip` axiom it must be the case that $\sigma = \sigma''$.
Now using the fact that $\mathcal{B}[\![b]\!]\sigma = ff$ and $\sigma = \sigma''$, we can construct (*) using the `while` axiom:

$$\frac{}{\langle \texttt{while } b \texttt{ do } s \texttt{ end}, \sigma\rangle \to \sigma''} \quad \text{if } \mathcal{B}[\![b]\!]\sigma = ff$$

□

Exercise 7. Free variables

We do the proof by induction on the structure of arithmetic expressions.

1. **Base cases:**

(a) The case when e is an integer, i: using the definition of $\mathcal{A}$ we have $\mathcal{A}[\![e]\!]\sigma = i$ and $\mathcal{A}[\![e]\!]\sigma' = i$. So $\mathcal{A}[\![e]\!]\sigma = \mathcal{A}[\![e]\!]\sigma'$ clearly holds in this case.

(b) The case when e is a variable, x: we have $\mathcal{A}[\![e]\!]\sigma = \sigma(x)$ and $\mathcal{A}[\![e]\!]\sigma' = \sigma'(x)$. From the assumption we have $\sigma(x) = \sigma'(x)$ because $x \in \mathrm{FV}(x)$. Thus, the statement holds in this case.

2. **Composite elements:** In this case e is of the form e_1 *op* e_2: we have $\mathcal{A}[\![e_1 \; op \; e_2]\!]\sigma = \mathcal{A}[\![e_1]\!]\sigma \; \overline{op} \; \mathcal{A}[\![e_2]\!]\sigma$ and $\mathcal{A}[\![e_1 \; op \; e_2]\!]\sigma' = \mathcal{A}[\![e_1]\!]\sigma' \; \overline{op} \; \mathcal{A}[\![e_2]\!]\sigma'$. Since e_i (for i=1,2) is an immediate subexpression of e_1 *op* e_2 and $\mathrm{FV}(e_i) \subseteq \mathrm{FV}(e_1 \; op \; e_2)$ we can apply the induction hypothesis to e_i and get $\mathcal{A}[\![e_i]\!]\sigma = \mathcal{A}[\![e_i]\!]\sigma'$.
It is now easy to see that the statement holds for e_1 *op* e_2.

Exercise 8. Substitution

Again we do structural induction on arithmetic expressions.

1. **Base cases:**

(a) The case when e is an integer, i:
We have to show that

$$\mathcal{A}[\![i[y \mapsto e_0]]\!]\sigma = \mathcal{A}[\![i]\!](\sigma[y \mapsto \mathcal{A}[\![e_0]\!]\sigma])$$

On the left-hand side first we use the definition of substitution and get $\mathcal{A}[\![i]\!]\sigma$. Then we can use the definition of the semantic function $\mathcal{A}$ and get i.
The right-hand side is of the form $\mathcal{A}[\![i]\!]\sigma'$ (where $\sigma' = \sigma[y \mapsto \mathcal{A}[\![e_0]\!]\sigma]$) and the definition of $\mathcal{A}$ gives i. Thus, this case is proven.

(b) The case when e is a variable, x:
We have to show that

$$\mathcal{A}[\![x[y \mapsto e_0]]\!]\sigma = \mathcal{A}[\![x]\!](\sigma[y \mapsto \mathcal{A}[\![e_0]\!]\sigma])$$

We have to differentiate two cases:

i. when $x = y$. In this case the left-hand side gives $\mathcal{A}[\![e_0]\!]\sigma$ when using the definition of substitution. On the right-hand side we first use the definition of $\mathcal{A}$ and get $(\sigma[y \mapsto \mathcal{A}[\![e_0]\!]\sigma])(x)$. Applying the definition of state-updates we get $\mathcal{A}[\![e_0]\!]\sigma$. Thus, the two sides are equal.

ii. when $x \neq y$. In this case on the left-hand side we first use the definition of substitution and get $\mathcal{A}[\![x]\!]\sigma$ and now the definition of $\mathcal{A}$ gives $\sigma(x)$. On the right-hand side we first use the definition of $\mathcal{A}$ and get $(\sigma[y \mapsto \mathcal{A}[\![e_0]\!]\sigma])(x)$. Now using the definition of state-updates we get $\sigma(x)$. Thus, the two sides are equal in this case.

2. **Composite elements:**
In this case e is of the form e_1 *op* e_2:
We have to show that

$$\mathcal{A}[\![(e_1\ op\ e_2)[y \mapsto e_0]]\!]\sigma = \mathcal{A}[\![e_1\ op\ e_2]\!](\sigma[y \mapsto \mathcal{A}[\![e_0]\!]\sigma])$$

In the first step we use the definition of substitution on the left-hand side and get

$$\mathcal{A}[\![e_1[y \mapsto e_0]\ op\ e_2[y \mapsto e_0]]\!]\sigma$$

As a second step we can use the definition of $\mathcal{A}$ and get

$$\mathcal{A}[\![e_1[y \mapsto e_0]]\!]\sigma\ \overline{op}\ \mathcal{A}[\![e_2[y \mapsto e_0]]\!]\sigma$$

Now we can use the induction hypothesis which gives

$$\mathcal{A}[\![e_1]\!](\sigma[y \mapsto \mathcal{A}[\![e_0]\!]\sigma])\ \overline{op}\ \mathcal{A}[\![e_2]\!](\sigma[y \mapsto \mathcal{A}[\![e_0]\!]\sigma])$$

Now using the definition of $\mathcal{A}$ we can see that this equals

$$\mathcal{A}[\![e_1\ op\ e_2]\!](\sigma[y \mapsto \mathcal{A}[\![e_0]\!]\sigma])$$

□

The intuition of the theorem is that we can either

- first do syntactical substitution on e and then apply $\mathcal{A}$ in state σ or
- leave e unchanged and apply $\mathcal{A}$ in the updated state.

Exercise 9. Extending **Bexp**

The only extensions to **Bexp** are `true`, `false`, `implies` and `equivalent`. The definitions are as follows:

$$\mathcal{B}'[\![\texttt{true}]\!]\sigma = tt$$

$$\mathcal{B}'[\![\texttt{false}]\!]\sigma = ff$$

$$\mathcal{B}'[\![b_1'\ \texttt{implies}\ b_2']\!]\sigma = \begin{cases} tt & \text{if } \mathcal{B}'[\![b_1']\!]\sigma = ff \text{ or } \mathcal{B}'[\![b_2']\!]\sigma = tt \\ ff & \text{otherwise} \end{cases}$$

$$\mathcal{B}'[\![b_1'\ \texttt{equivalent}\ b_2']\!]\sigma = \begin{cases} tt & \text{if } \mathcal{B}'[\![b_1']\!]\sigma = \mathcal{B}'[\![b_2']\!]\sigma \\ ff & \text{otherwise} \end{cases}$$

Exercise 10. Equivalence of **Bexp** and **Bexp'**

We use induction on the structure of boolean expressions of **Bexp'**.

1. (base case) $b' = \texttt{true}$.
   Let's guess that a corresponding $b \in$ **Bexp** expression is $1 = 1$. We have to show that

   $$\mathcal{B}'[\![\texttt{true}]\!]\sigma = \mathcal{B}[\![\texttt{1=1}]\!]\sigma$$

   Using the definition of $\mathcal{B}$ we get $\mathcal{B}[\![\texttt{1=1}]\!]\sigma = (\mathcal{A}[\![\texttt{1}]\!]\sigma = \mathcal{A}[\![\texttt{1}]\!]\sigma) = (1 = 1) = tt$ which gives the same definition as $\mathcal{B}'[\![\texttt{true}]\!]\sigma$.

2. (base case) $b' = \texttt{false}$.
   Let's guess that a corresponding $b \in \mathbf{Bexp}$ expression is $1 = 0$. We have to show that

$$\mathcal{B}'[\![\texttt{false}]\!]\sigma = \mathcal{B}[\![\texttt{1=0}]\!]\sigma$$

   Using the definition of $\mathcal{B}$ we get $\mathcal{B}[\![\texttt{1=0}]\!]\sigma = (\mathcal{A}[\![\texttt{1}]\!]\sigma = \mathcal{A}[\![\texttt{0}]\!]\sigma) = (1 = 0) = ff$ which gives the same definition as $\mathcal{B}'[\![\texttt{false}]\!]\sigma$.

3. (composite element) $b' = b'_1$ `implies` b'_2.
   Our guess here is $b =$ `not` b_1 `or` b_2, where $\mathcal{B}'[\![b'_1]\!] = \mathcal{B}[\![b_1]\!]$ and $\mathcal{B}'[\![b'_2]\!] = \mathcal{B}[\![b_2]\!]$. From the induction hypothesis we know that there exist such b_1 and $b_2 \in \mathbf{Bexp}$ expressions.

   We have to show that

$$\mathcal{B}'[\![b'_1 \texttt{ implies } b'_2]\!]\sigma = \mathcal{B}[\![\texttt{not } b_1 \texttt{or } b_2]\!]\sigma$$

   Using the definition of **Bexp'** and the induction hypothesis, we get

$$\mathcal{B}'[\![b'_1 \texttt{ implies } b'_2]\!]\sigma \;=\; \begin{cases} tt & \text{if } \mathcal{B}[\![b_1]\!]\sigma = ff \text{ or } \mathcal{B}[\![b_2]\!]\sigma = tt \\ ff & \text{otherwise} \end{cases}$$

   Using the definition of **Bexp** we get

$$\begin{aligned} \mathcal{B}[\![\texttt{not } b_1 \texttt{ or } b_2]\!]\sigma \;&=\; \begin{cases} tt & \text{if } \mathcal{B}[\![\texttt{not } b_1]\!]\sigma = tt \text{ or } \mathcal{B}[\![b_2]\!]\sigma = tt \\ ff & \text{otherwise} \end{cases} \\ &=\; \begin{cases} tt & \text{if } \mathcal{B}[\![b_1]\!]\sigma = ff \text{ or } \mathcal{B}[\![b_2]\!]\sigma = tt \\ ff & \text{otherwise} \end{cases} \end{aligned}$$

□

4. (composite element) $b' = b'_1$ `equivalent` b'_2.
   Our guess this time is

$$b \;=\; (b_1 \texttt{ and } b_2) \texttt{ or } (\texttt{not } b_1 \texttt{ and not } b_2)$$

   where $\mathcal{B}'[\![b'_1]\!] = \mathcal{B}[\![b_1]\!]$ and $\mathcal{B}'[\![b'_2]\!] = \mathcal{B}[\![b_2]\!]$. Again, from the induction hypothesis we know that there exist such b_1 and $b_2 \in \mathbf{Bexp}$ expressions.

   We have to show that

$$\mathcal{B}'[\![b'_1 \texttt{ equivalent } b'_2]\!]\sigma = \mathcal{B}[\![(b_1 \texttt{ and } b_2) \texttt{ or } (\texttt{not } b_1 \texttt{ and not } b_2)]\!]\sigma$$

   Using the definition of **Bexp'** and the induction hypothesis, we get

$$\mathcal{B}'[\![b'_1 \texttt{ equivalent } b'_2]\!]\sigma \;=\; \begin{cases} tt & \text{if } \mathcal{B}[\![b_1]\!]\sigma = \mathcal{B}[\![b_2]\!]\sigma \\ ff & \text{otherwise} \end{cases}$$

   Using the definition of **Bexp** we get

$$\mathcal{B}[\![(b_1 \texttt{ and } b_2) \texttt{ or } (\texttt{not } b_1 \texttt{ and not } b_2)]\!]\sigma = \begin{cases} tt & \text{if } \mathcal{B}[\![b_1 \texttt{ and } b_2]\!]\sigma = tt \text{ or} \\ & \quad \mathcal{B}[\![\texttt{not } b_1 \texttt{ and not } b_2]\!]\sigma = tt \\ ff & \text{otherwise} \end{cases}$$

$$= \begin{cases} tt & \text{if } \mathcal{B}[\![b_1]\!]\sigma = tt \text{ and } \mathcal{B}[\![b_2]\!]\sigma = tt \quad \text{or} \\ & \quad \mathcal{B}[\![\texttt{not } b_1]\!]\sigma = tt \text{ and } \mathcal{B}[\![\texttt{not } b_2]\!]\sigma = tt \\ ff & \text{otherwise} \end{cases}$$

$$= \begin{cases} tt & \text{if } \mathcal{B}[\![b_1]\!]\sigma = tt \text{ and } \mathcal{B}[\![b_2]\!]\sigma = tt \quad \text{or} \\ & \quad \mathcal{B}[\![b_1]\!]\sigma = ff \text{ and } \mathcal{B}[\![b_2]\!]\sigma = ff \\ ff & \text{otherwise} \end{cases}$$

$$= \begin{cases} tt & \text{if } \mathcal{B}[\![b_1]\!]\sigma = \mathcal{B}[\![b_2]\!]\sigma \\ ff & \text{otherwise} \end{cases}$$

□

5. Other composite elements are just applications of the induction hypothesis.