ETH zürich

P. Müller and D. Basin

Formal Methods and Functional Programming

Session Sheet 10: IMP States and Expressions

Assignment 1 (Simplifying State Updates)

Task 1.1: Prove that for all states σ and variables x, it holds that $\sigma[x \mapsto \sigma(x)] = \sigma$.

Solution. Note that no induction is necessary here. We assume the state σ and the variable x be arbitrary and need to show that $\forall y \cdot \sigma[x \mapsto \sigma(x)](y) = \sigma(y)$. To do so, we assume the variable y to be arbitrary. Using the definition of state update, we get

$$\sigma[x \mapsto \sigma(x)](y) = \left\{\begin{array}{ll} \sigma(x) & \text{if } y \equiv x \\ \sigma(y) & \text{if } y \not\equiv x \end{array}\right\} = \sigma(y).$$

Task 1.2: Assume that for all states σ, for all variables x, y, and for all values v, w:

$$x \not\equiv y \implies \sigma[x \mapsto v][y \mapsto w] = \sigma[y \mapsto w][x \mapsto v] \tag{1}$$

The proof of this statement is left for the exercise sheet.

Prove that for all variables x, for all values v, for all natural numbers n, for all sequences of length n of variables $\vec{y} \equiv \langle y_1, \ldots, y_n \rangle$ and corresponding values $\vec{w} \equiv \langle w_1, \ldots, w_n \rangle$, and for all states σ:

$$x \notin \vec{y} \implies \sigma[x \mapsto v][\vec{y} \mapsto \vec{w}] = \sigma[\vec{y} \mapsto \vec{w}][x \mapsto v].$$

Note: By $x \notin \vec{y}$, we mean that $x \not\equiv y_i$, for all $i \in \{1, \ldots, n\}$.

Note: We use $\sigma[\vec{y} \mapsto \vec{w}]$ to denote the sequence of updates $\sigma[y_1 \mapsto w_1] \ldots [y_n \mapsto w_n]$.

Solution. Let x and v be arbitrary and let

$$P(n) \equiv \forall \sigma, \vec{y}, \vec{w} \cdot |\vec{y}| = |\vec{w}| = n \wedge x \notin \vec{y} \implies \sigma[x \mapsto v][\vec{y} \mapsto \vec{w}] = \sigma[\vec{y} \mapsto \vec{w}][x \mapsto v],$$

where $|\vec{y}|$ and $|\vec{w}|$ denote the length of the sequences $\vec{y}$ and $\vec{w}$, respectively. We show $\forall n \cdot P(n)$ by weak induction on n.

- **Base Case:** We have to show that $P(0)$ holds. Let σ, $\vec{y}$, and $\vec{w}$ be arbitrary. Since $n = 0$, the sequences $\vec{y}$ and $\vec{w}$ can only be empty. Thus, after assuming (the vacuous property) $x \notin \vec{y}$, we are left with showing that $\sigma[x \mapsto v] = \sigma[x \mapsto v]$, which is trivially true.

- **Step Case:** As our induction hypothesis, we assume that $P(n)$ holds for some natural number n. We have to show that $P(n+1)$ holds. Let σ be arbitrary, and let $\vec{y}$, and $\vec{w}$ be arbitrary sequences of length $n+1$. We need to show that

  $$x \notin \vec{y} \implies \sigma[x \mapsto v][\vec{y} \mapsto \vec{w}] = \sigma[\vec{y} \mapsto \vec{w}][x \mapsto v].$$

  We assume $x \notin \vec{y}$ and seek to prove $\sigma[x \mapsto v][\vec{y} \mapsto \vec{w}] = \sigma[\vec{y} \mapsto \vec{w}][x \mapsto v]$. Since the sequences are of length at least 1, there have to be first elements y_1 and w_1, respectively. By (1), with $x \notin \vec{y} \implies x \not\equiv y_1$, we obtain $\sigma[x \mapsto v][y_1 \mapsto w_1] = \sigma[y_1 \mapsto w_1][x \mapsto v]$. Thus, we have

  $$\begin{aligned} & \sigma[x \mapsto v][y_1 \mapsto w_1][y_2 \mapsto w_2] \ldots [y_{n+1} \mapsto w_{n+1}] \\ = {} & \sigma[y_1 \mapsto w_1][x \mapsto v][y_2 \mapsto w_2] \ldots [y_{n+1} \mapsto w_{n+1}] \\ = {} & \sigma[y_1 \mapsto w_1][y_2 \mapsto w_2] \ldots [y_{n+1} \mapsto w_{n+1}][x \mapsto v], && \text{(IH)} \end{aligned}$$

  as required (note that in the last step, we instantiated $P(n)$ with $\sigma \rightsquigarrow \sigma[y_1 \mapsto w_1]$, and $\vec{y} \rightsquigarrow \langle y_2, \ldots, y_{n+1} \rangle$, $\vec{w} \rightsquigarrow \langle w_2, \ldots, w_{n+1} \rangle$, which are both sequences of length n).

Assignment 2 (Substitution on Arithmetic Expressions)

Intuitively, $e[x \mapsto e']$ denotes the arithmetic expression e with all occurrences of x replaced with the arithmetic expression e'. Recall the formal definition:

$$e[x \mapsto e'] \equiv \begin{cases} n & \text{if } e \equiv n \text{ for some numerical value } n \\ e' & \text{if } e \equiv y \text{ for some variable } y \text{ with } y \equiv x \\ y & \text{if } e \equiv y \text{ for some variable } y \text{ with } y \not\equiv x \\ e_1[x \mapsto e'] \mathbin{op} e_2[x \mapsto e'] & \text{if } e \equiv e_1 \mathbin{op} e_2\text{, for some } e_1, e_2\text{, and } op \end{cases}$$

Task: Prove the following statement:

$$\forall \sigma, e, e', x \cdot \big(\mathcal{A}[\![e[x \mapsto e']]\!]\sigma = \mathcal{A}[\![e]\!](\sigma[x \mapsto \mathcal{A}[\![e']\!]\sigma])\big)$$

Hint: Define a suitable predicate $P(e)$ and prove $\forall e \cdot P(e)$ by either *weak structural induction* or *strong structural induction* on the arithmetic expression e. If you choose to do a strong structural induction, you have to prove $P(e)$ for some arbitrary e and may assume $\forall e'' \sqsubset e \cdot P(e'')$ as your induction hypothesis. Note that, here, $e'' \sqsubset e$ denotes that e'' is a proper sub-expression of e. Since arithmetic expressions are finite, the relation $\sqsubset$ is a well-founded ordering. Thus, strong structural induction on arithmetic expressions can be seen as a special case of well-founded induction.

Solution. Let σ, x and e' be arbitrary. We define

$$P(e) \equiv \big(\mathcal{A}[\![e[x \mapsto e']]\!]\sigma = \mathcal{A}[\![e]\!](\sigma[x \mapsto \mathcal{A}[\![e']\!]\sigma])\big)$$

and prove $\forall e.\ P(e)$ by strong structural induction on e (note that a weak structural induction would also work). We have to show $P(e)$ for some arbitrary arithmetic expression e and assume $\forall e'' \sqsubset e \cdot P(e'')$ as our induction hypothesis. We proceed by a case analysis on e:

- **Case** $e \equiv n$, for some numerical value n: We have

$$\mathcal{A}[\![n[x \mapsto e']]\!]\sigma = \mathcal{A}[\![n]\!]\sigma = \mathcal{N}[\![n]\!] = \mathcal{A}[\![n]\!]\big(\sigma[x \mapsto \mathcal{A}[\![e']\!]\sigma]\big).$$

- **Case** $e \equiv y$, for some variable y: We make a further case distinction:
  - **Subcase** $y \equiv x$: We have

$$\mathcal{A}[\![x[x \mapsto e']]\!]\sigma = \mathcal{A}[\![e']\!]\sigma = \big(\sigma[x \mapsto \mathcal{A}[\![e']\!]\sigma]\big)(y) = \mathcal{A}[\![x]\!]\big(\sigma[x \mapsto \mathcal{A}[\![e']\!]\sigma]\big).$$

  - **Subcase** $y \not\equiv x$: We have

$$\mathcal{A}[\![y[x \mapsto e']]\!]\sigma = \mathcal{A}[\![y]\!]\sigma = \sigma(y) = \big(\sigma[x \mapsto \mathcal{A}[\![e']\!]\sigma]\big)(y) = \mathcal{A}[\![y]\!]\big(\sigma[x \mapsto \mathcal{A}[\![e']\!]\sigma]\big).$$

- **Case** $e \equiv e_1\, op\, e_2$, for some arithmetic expressions e_1, e_2 and some arithmetic operator op: Note that $e_1 \sqsubset e$ and $e_2 \sqsubset e$. Thus, by the induction hypothesis, we get

$$\mathcal{A}[\![e_i[x \mapsto e']]\!]\sigma = \mathcal{A}[\![e_i]\!]\big(\sigma[x \mapsto \mathcal{A}[\![e']\!]\sigma]\big), \tag{2}$$

for $i \in \{1, 2\}$, and can conclude that

$$\begin{aligned}
\mathcal{A}[\![(e_1\, op\, e_2)[x \mapsto e']]\!]\sigma &= \mathcal{A}[\![(e_1[x \mapsto e']\ \ op\ \ e_2[x \mapsto e'])]\!]\sigma \\
&= \mathcal{A}[\![e_1[x \mapsto e']]\!]\sigma\ \overline{op}\ \mathcal{A}[\![e_2[x \mapsto e']]\!]\sigma \\
&\overset{(??)}{=} \mathcal{A}[\![e_1]\!]\big(\sigma[x \mapsto \mathcal{A}[\![e']\!]\sigma]\big)\ \overline{op}\ \mathcal{A}[\![e_2]\!]\big(\sigma[x \mapsto \mathcal{A}[\![e']\!]\sigma]\big) \\
&= \mathcal{A}[\![(e_1\, op\, e_2)]\!]\big(\sigma[x \mapsto \mathcal{A}[\![e']\!]\sigma]\big).
\end{aligned}$$

Assignment 3 (Big-Step Semantics)

Let s be the following statement:

```
y := 1;
while x > 0 do
   y := y * 2;
   x := x - 1
end
```

Task 3.1. What function does the **IMP** statement s compute when the variable $\mathtt{x}$ initially stores a non-negative integer?

Solution. The statement s stores 2^X in variable $\mathtt{y}$ where X is the initial value of variable $\mathtt{x}$. The variable $\mathtt{x}$ is set to 0 by executing the statement s.

Task 3.2. Let σ be a state with $\sigma(\mathtt{x}) = 2$. Prove that there is a state σ' with $\sigma'(\mathtt{y}) = 4$ such that $\langle s, \sigma \rangle \to \sigma'$ using the rules of the big-step semantics for **IMP**.

Solution.

We use the following abbreviations: s' is the statement `y := y*2; x := x-1` and s_w the statement `while x>0 do` s' `end`. Moreover, we abbreviate a state $\sigma[\mathtt{x}_1 \mapsto v_1]\dots[\mathtt{x}_k \mapsto v_k]$ as $\sigma[\mathtt{x}_1, \dots, \mathtt{x}_k \mapsto v_1, \dots, v_k]$. Finally, we also use the simplifying properties of state updates from task 1 and the fact that $\mathtt{x} \not\equiv \mathtt{y}$.

$$\dfrac{\dfrac{}{\langle \mathtt{y}\!:=\!1, \sigma\rangle \to \sigma[\mathtt{y} \mapsto 1]}\,(\textsc{Ass}_{NS}) \qquad \dfrac{\dfrac{\dfrac{}{\langle \mathtt{y}\!:=\!\mathtt{y}*2, \sigma[\mathtt{y} \mapsto 1]\rangle \to \sigma[\mathtt{y} \mapsto 2]}\,(\textsc{Ass}_{NS}) \quad \dfrac{}{\langle \mathtt{x}\!:=\!\mathtt{x}-1, \sigma[\mathtt{y} \mapsto 2]\rangle \to \sigma[\mathtt{x},\mathtt{y} \mapsto 1, 2]}\,(\textsc{Ass}_{NS})}{\langle s', \sigma[\mathtt{y} \mapsto 1]\rangle \to \sigma[\mathtt{x},\mathtt{y} \mapsto 1, 2]}\,(\textsc{Seq}_{NS}) \quad \dfrac{T_1}{\langle s_w, \sigma[\mathtt{x},\mathtt{y} \mapsto 1, 2]\rangle \to \sigma[\mathtt{x},\mathtt{y} \mapsto 0, 4]}}{\langle s_w, \sigma[\mathtt{y} \mapsto 1]\rangle \to \sigma[\mathtt{x},\mathtt{y} \mapsto 0, 4]}\,(\textsc{WhT}_{NS})}{\langle s, \sigma\rangle \to \sigma[\mathtt{x},\mathtt{y} \mapsto 0, 4]}\,(\textsc{Seq}_{NS})$$

where T_1 is the derivation tree:

$$\dfrac{\dfrac{\dfrac{}{\langle \mathtt{y}\!:=\!\mathtt{y}*2, \sigma[\mathtt{x},\mathtt{y} \mapsto 1, 2]\rangle \to \sigma[\mathtt{x},\mathtt{y} \mapsto 1, 4]}\,(\textsc{Ass}_{NS}) \quad \dfrac{}{\langle \mathtt{x}\!:=\!\mathtt{x}-1, \sigma[\mathtt{x},\mathtt{y} \mapsto 1, 4]\rangle \to \sigma[\mathtt{x},\mathtt{y} \mapsto 0, 4]}\,(\textsc{Ass}_{NS})}{\langle s', \sigma[\mathtt{x},\mathtt{y} \mapsto 1, 2]\rangle \to \sigma[\mathtt{x},\mathtt{y} \mapsto 0, 4]}\,(\textsc{Seq}_{NS}) \quad \dfrac{}{\langle s_w, \sigma[\mathtt{x},\mathtt{y} \mapsto 0, 4]\rangle \to \sigma[\mathtt{x},\mathtt{y} \mapsto 0, 4]}\,(\textsc{WhF}_{NS})}{\langle s_w, \sigma[\mathtt{x},\mathtt{y} \mapsto 1, 2]\rangle \to \sigma[\mathtt{x},\mathtt{y} \mapsto 0, 4]}\,(\textsc{WhT}_{NS})$$