Formal Methods and Functional Programming

Übungsstunde 9: IMP State und Expressions

Department of Computer Science, ETH Zürich

Robin Ebersberger, 16.04.2025

Überblick

- Common Mistakes Serie
- Recap Vorlesung
- State Updates - Regeln
- Substitution Lemma Aexp
- Substitution bei “unused variables”
- Derivation Trees für IMP

Common Mistakes Serie

- Notation: statt $\forall n \geq 2.$ ... ab jetzt $\forall n.\ n \geq 2 \Rightarrow$ ...
- Strong Induction mit Weak Induction: nicht vergessen $\forall j.\ 0 \leq j \leq n \Rightarrow P(j)$ im Step Case wieder zu beweisen
- Bei Haskell Induction Proofs auf Fallunterscheidungen achten:

```
rep 0 v = []                    -- (R1)
rep n v = v:(rep (n-1) v)       -- (R2)
```

IMP Recap

- $\mathcal{N} : \text{Numeral} \to \text{Val}$ übersetzt Numerals, z.B. $\mathcal{N}[[101]] = 7$
- $\text{State} : \text{Var} \to \text{Val}$ gibt Variablen einen Wert, z.B. $\sigma(x) = 5$
- $\mathcal{A} : \text{Aexp} \to \text{State} \to \text{Val}$ wertet Aexp (arithm. Ausdrücke) unter einem State aus, z.B. $\mathcal{A}[[x + 5]]\sigma = 9$ falls $\sigma(x) = 4$
- $\mathcal{B} : \text{Bexp} \to \text{State} \to \text{Val}$ wertet Bexp (boolesche Ausdrücke) unter einem State zu truth values (*tt* oder *ff*) aus, z.B. $\mathcal{B}[[z + 3 \leq 7 \text{ and } tt]]\sigma = tt$ falls $\sigma(z) \leq 4$
- Structural Induction over Programs

```
data Aexp = Bin Op Aexp Aexp
          | Var String
          | Num Integer
data Op   = Add | Sub | Mul
```

$$\frac{\Gamma \vdash P(x) \quad \Gamma \vdash P(n) \quad \Gamma, P(e_1), P(e_2) \vdash P(e_1 \ op \ e_2)}{\Gamma \vdash \forall e \in Aexp.P(e)} \ (*)$$

$^*(x, n, e_1, e_2, op \text{ not free in } \Gamma)$

Operational Semantics

- Big-Step (Natural) Semantics vs. Small-Step (Structural Operational) Semantics
- Big-Step: Welcher State (Ergebnis) nach Ausführung
- Small-Step: Welche Schritte (steps) während Ausführung

Big-Step Semantics

Configurations

- $\langle s, \sigma \rangle$, also führe Stmt s in State σ aus
- σ, also Programm fertig (final state)

Inference Rules

$$\frac{}{\langle \text{skip}, \sigma \rangle \to \sigma}\,\text{Skip}_{\text{NS}} \qquad \frac{}{\langle x := e, \sigma \rangle \to \sigma[x \mapsto \mathcal{A}[[e]]\sigma]}\,\text{Ass}_{\text{NS}}$$

$$\frac{\langle s, \sigma \rangle \to \sigma'}{\langle \text{if } b \text{ then } s \text{ else } s' \text{ end}, \sigma \rangle \to \sigma'}\,\text{IfT}_{\text{NS}} \quad \text{if } \mathcal{B}[[b]]\sigma = \text{tt}$$

$$\frac{\langle s', \sigma \rangle \to \sigma'}{\langle \text{if } b \text{ then } s \text{ else } s' \text{ end}, \sigma \rangle \to \sigma'}\,\text{IfF}_{\text{NS}} \quad \text{if } \mathcal{B}[[b]]\sigma = \text{ff}$$

Derivation Trees

- Mithilfe der Regeln erhalten wir einen Finite Derivation Tree T für z.B.

$$\langle x := 1; \text{if } x < 2 \text{ then } x := x + 1 \text{ else skip end}, \sigma_{\text{zero}} \rangle \to \sigma_{\text{zero}}[x \mapsto 2]$$

- Wir schreiben:
  $\text{root}(T) \equiv \langle x := 1; \text{if } x < 2 \text{ then } x := x + 1 \text{ else skip end}, \sigma_{\text{zero}} \rangle \to \sigma_{\text{zero}}[x \mapsto 2]$
- Zeichnen Derivation Trees: side-conditions!
- $\vdash \langle s, \sigma \rangle \to \sigma' \Leftrightarrow \exists T.\ \text{root}(T) \equiv \langle s, \sigma \rangle \to \sigma'$
- Termination of $\langle s, \sigma \rangle$:
  “terminates successfully”: $\exists \sigma'.\ \vdash \langle s, \sigma \rangle \to \sigma'$
  “fails to terminate”: $\nexists \sigma'.\ \vdash \langle s, \sigma \rangle \to \sigma'$

State Updates I

Zeige, dass:

$$\forall \sigma, x.\ \sigma[x \mapsto \sigma(x)] = \sigma$$

Erinnerung:

$$\text{State} : \text{Var} \to \text{Val}$$

$$(\sigma[y \mapsto v])(x) = \begin{cases} v & \text{if } x \equiv y \\ \sigma(x) & \text{if } x \not\equiv y \end{cases}$$

$$\sigma_1 = \sigma_2 \Leftrightarrow \forall x. \sigma_1(x) = \sigma_2(x)$$

State Updates I - Lösung

Show $\forall x, \sigma.\ \sigma[x \mapsto \sigma(x)] = \sigma$
Let state σ, variable x arbitrary but fixed.

For $\sigma[x \mapsto \sigma(x)] = \sigma$ we need to show that $\forall y.\ (\sigma[x \mapsto \sigma(x)])(y) = \sigma(y)$
Let variable y arbitrary but fixed.

$$\sigma[x \mapsto \sigma(x)](y) = \begin{cases} \sigma(x) & \text{if } x \equiv y \\ \sigma(y) & \text{if } x \not\equiv y \end{cases} = \sigma(y)$$

State Updates II

Wir können annehmen (Serie 9), dass:

$$\forall \sigma, x, y, v, w.\ x \not\equiv y \Longrightarrow \sigma[x \mapsto v][y \mapsto w] = \sigma[y \mapsto w][x \mapsto v]$$

Zeige, dass:

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

Wobei: $\vec{y} \equiv \langle y_1, ..., y_n \rangle, \vec{w} \equiv \langle w_1, ..., w_n \rangle$ und $\sigma[\vec{y} \mapsto \vec{w}] = \sigma[y_1 \mapsto w_1]...[y_n \mapsto w_n]$
$x \notin \vec{y}$ bedeutet einfach $x \not\equiv y_i$ für alle $i \in \{1, ..., n\}$

State Updates II - Lösung

$$\text{Ziel: } \forall \sigma, \vec{y}, \vec{w}, x, v : x \notin \vec{y} \Longrightarrow \sigma[x \mapsto v][\vec{y} \mapsto \vec{w}] = \sigma[\vec{y} \mapsto \vec{w}][x \mapsto v]$$

Let x, v be arbitrary but fixed.
Let $P(n) \equiv \forall \sigma, \vec{y}, \vec{w}.\ (|\vec{y}| = |\vec{w}| = n \wedge x \notin \vec{y}) \Rightarrow \sigma[x \mapsto v][\vec{y} \mapsto \vec{w}] = \sigma[\vec{y} \mapsto \vec{w}][x \mapsto v]$

Prove $\forall n.P(n)$ by weak induction on n.

Base Case: Let $\sigma, \vec{y}, \vec{w}$ arbitrary but fixed st. LHS of implication holds. Show $P(0)$.

Since $|\vec{y}| = |\vec{w}| = 0$, we have $\sigma[x \mapsto v][\vec{y} \mapsto \vec{w}] = \sigma[x \mapsto v] = \sigma[\vec{y} \mapsto \vec{w}][x \mapsto v]$

State Updates II - Lösung

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

Step Case: Let $n \in \mathbb{N}$ arbitrary but fixed. Show $P(n+1)$ assuming IH: $P(n)$.
Let $\sigma, \vec{y}, \vec{w}$ arbitrary but fixed st. LHS of implication holds.
We know $\vec{y} = \langle y_1, ..., y_{n+1} \rangle$ and $\vec{w} = \langle w_1, ..., w_{n+1} \rangle$. Let $\vec{y}' = \langle y_2, ..., y_{n+1} \rangle$ and $\vec{w}' = \langle w_2, ..., w_{n+1} \rangle$.

$$\begin{aligned}
& \sigma[x \mapsto v][\vec{y} \mapsto \vec{w}] \\
= \ & \sigma[x \mapsto v][y_1 \mapsto w_1][\vec{y}' \mapsto \vec{w}'] \\
= \ & (\sigma[y_1 \mapsto w_1])[x \mapsto v][\vec{y}' \mapsto \vec{w}'] \\
= \ & (\sigma[y_1 \mapsto w_1])[\vec{y}' \mapsto \vec{w}'][x \mapsto v] \quad (\text{IH}) \\
= \ & \sigma[y_1 \mapsto w_1][\vec{y}' \mapsto \vec{w}'][x \mapsto v] \\
= \ & \sigma[\vec{y} \mapsto \vec{w}][x \mapsto v]
\end{aligned}$$

Substitution Lemma Aexp

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

Zeige, dass:

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

Tipp: Wähle $P(e)$ und zeige $\forall e.P(e)$ über strong structural induction!
Schreibe: $e'' \sqsubset e$ für e'' “proper subexpression of” e

$$\begin{aligned} \mathcal{A}[[x]]\sigma &= \sigma(x) \\ \mathcal{A}[[n]]\sigma &= \mathcal{N}[[n]] \\ \mathcal{A}[[e_1 \text{ op } e_2]]\sigma &= \mathcal{A}[[e_1]]\sigma \ \overline{\text{op}} \ \mathcal{A}[[e_2]]\sigma \quad \text{for op} \in \text{Op} \end{aligned}$$

Substitution Lemma Aexp - Lösung

$$\text{Ziel: } \forall \sigma, e, e', x : \mathcal{A}[[e[x \mapsto e']]]\sigma = \mathcal{A}[[e]](\sigma[x \mapsto \mathcal{A}[[e']]\sigma])$$

Let σ, x, e' arbitrary but fixed. (Deal w/ inner quantifiers first)
Let $P(e) \equiv \mathcal{A}[[e[x \mapsto e']]]\sigma = \mathcal{A}[[e]](\sigma[x \mapsto \mathcal{A}[[e']]\sigma])$
Prove $\forall e.P(e)$ by strong structural induction on e.
For some arbitrary arithm. expr e prove $P(e)$ assuming IH: $\forall e'' \sqsubset e.\ P(e'')$
Case analysis on e:

- Case $e \equiv n$ for some numeral n:

$$\begin{aligned} & \mathcal{A}[[n[x \mapsto e']]]\sigma \\ = \ & \mathcal{A}[[n]]\sigma \\ = \ & \mathcal{N}[[n]] \\ = \ & \mathcal{A}[[n]](\sigma[x \mapsto \mathcal{A}[[e']]\sigma]) \end{aligned}$$

Substitution Lemma Aexp - Lösung

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

- Case $e \equiv y$ for some variable y:
  Futher case distinction on y
  - Subcase $y \equiv x$:

$$\begin{aligned} & \mathcal{A}[[x[x \mapsto e']]]\sigma \\ = \; & \mathcal{A}[[e']]\sigma \\ = \; & (\sigma[x \mapsto \mathcal{A}[[e']]\sigma])(y) \\ = \; & \mathcal{A}[[x]](\sigma[x \mapsto \mathcal{A}[[e']]\sigma]) \end{aligned}$$

Substitution Lemma Aexp - Lösung

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

- Case $e \equiv y$ for some variable y:
  Futher case distinction on y
  - Subcase $y \not\equiv x$:

$$\begin{aligned} & \mathcal{A}[[y[x \mapsto e']]]\sigma \\ = \; & \mathcal{A}[[y]]\sigma \\ = \; & \sigma(y) \\ = \; & (\sigma[x \mapsto \mathcal{A}[[e']]\sigma])(y) \\ = \; & \mathcal{A}[[y]](\sigma[x \mapsto \mathcal{A}[[e']]\sigma]) \end{aligned}$$

Substitution Lemma Aexp - Lösung

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

- Case $e \equiv (e_1 \text{ op } e_2)$ for some arithm. exprs e_1, e_2 and some arithm. operator op:
  From $e_1 \sqsubset e, e_2 \sqsubset e$ we get $P(e_1), P(e_2)$ per IH

$$\begin{aligned}
&\mathcal{A}[[(e_1 \text{ op } e_2)[x \mapsto e']]]\sigma \\
&= \mathcal{A}[[(e_1[x \mapsto e'] \text{ op } e_2[x \mapsto e'])]]\sigma \\
&= \mathcal{A}[[e_1[x \mapsto e']]]\sigma \ \overline{\text{op}} \ \mathcal{A}[[e_2[x \mapsto e']]]\sigma \\
&= \mathcal{A}[[e_1]](\sigma[x \mapsto \mathcal{A}[[e']]\sigma]) \ \overline{\text{op}} \ \mathcal{A}[[e_2]](\sigma[x \mapsto \mathcal{A}[[e']]\sigma]) \quad \text{(IH)} \\
&= \mathcal{A}[[(e_1 \text{ op } e_2)]](\sigma[x \mapsto \mathcal{A}[[e']]\sigma])
\end{aligned}$$

Substitution bei “unused variables”

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

Zeige, dass:

$$\forall e, e', x.\ x \notin \mathrm{FV}(e) \Longrightarrow e[x \mapsto e'] \equiv e$$

Substitution bei “unused variables” - Lösung

$$\text{Ziel: } \forall e, e', x.\ x \notin \mathrm{FV}(e) \Longrightarrow e[x \mapsto e'] \equiv e$$

Let e', x be arbitrary but fixed.
Let $P(e) \equiv (x \notin \mathrm{FV}(e) \Rightarrow e[x \mapsto e'] \equiv e)$
Prove $\forall e.P(e)$ by strong structural induction on e.
For arbitrary arithm. expr e prove $P(e)$ assuming IH: $\forall e'' \sqsubset e.\ P(e'')$
Case analysis on e:

- Case $e \equiv n$ for some numeral n: We get $n[x \mapsto e'] \equiv n$ by substitution
- Case $e \equiv y$ for some variable y: Assuming LHS of implication ($x \notin \mathrm{FV(e)}$) we get $x \not\equiv y$, so $y[x \mapsto e'] \equiv y$

Substitution bei “unused variables” - Lösung

$$P(e) \equiv (x \notin \mathrm{FV}(e) \Rightarrow e[x \mapsto e'] \equiv e)$$

- Case $e \equiv (e_1 \ \mathrm{op} \ e_2)$ for some arithm. exprs e_1, e_2 and some arithm. operator op. From $e_1 \sqsubset e, e_2 \sqsubset e$ we get $P(e_1), P(e_2)$ per IH.

  We know $\mathrm{FV}(e) = \mathrm{FV}(e_1) \cup \mathrm{FV}(e_2)$ and assuming $x \notin \mathrm{FV}(e)$ we immediatly get $x \notin \mathrm{FV}(e_1)$ and $x \notin \mathrm{FV}(e_2)$.

$$\begin{aligned} & (e_1 \ \mathrm{op} \ e_2)[x \mapsto e'] \\ \equiv\ & e_1[x \mapsto e'] \ \mathrm{op} \ e_2[x \mapsto e'] \\ \equiv\ & e_1 \ \mathrm{op} \ e_2 \qquad (\mathrm{IH}) \end{aligned}$$

Big-Step Trees

$$\frac{\langle s, \sigma\rangle \to \sigma' \quad \langle s', \sigma'\rangle \to \sigma''}{\langle s; s', \sigma\rangle \to \sigma''}\,\mathrm{Seq}_{\mathrm{NS}} \qquad \frac{}{\langle x := e, \sigma\rangle \to \sigma[x \mapsto \mathcal{A}[[e]]\sigma]}\,\mathrm{Ass}_{\mathrm{NS}}$$

$$\frac{\langle s, \sigma\rangle \to \sigma'}{\langle \text{if } b \text{ then } s \text{ else } s' \text{ end}, \sigma\rangle \to \sigma'}\,\mathrm{IfT}_{\mathrm{NS}} \quad \text{if } \mathcal{B}[[b]]\sigma = \mathrm{tt}$$

Zeichne den Derivation Tree für:

$$\langle x := 1; \text{if } x < 2 \text{ then } x := x + 1 \text{ else skip end}, \sigma_{\mathrm{zero}}\rangle \to \quad ??$$