Marco Kunze (makunze@cs.tu-berlin.de)
Sebastian Nowozin (nowozin@cs.tu-berlin.de)

SS 2002
13.06.2002

Lösungen zum 8. Übungsblatt TheGI2

Lösung zu Aufgabe 1:

Variablensystem: Erweiterung von $\Sigma = (S, OP)$ zu $\Sigma' = (S, OP, X)$ mit

$$\begin{array}{lccl} \text{vars} & x, y & = & \text{data} \\ & z & = & \text{string} \end{array} \quad X_{data} = \{x, y\}, X_{string} = \{z\}$$

Definition der Variablenbelegung $ass_{data} : X_{data} \Rightarrow A_{data}$:

$$\begin{array}{rcl} ass_{data}(x) & = & a \\ ass_{data}(y) & = & b \end{array}$$

Definition der Variablenbelegung $ass_{string} : X_{string} \Rightarrow A_{string}$:

$$ass_{string}(z) = abba$$

Auswertung des Terms $radd(radd(empty, d2), x)$:

$$\begin{array}{rl} & xeval(ass)_{string}(radd(radd(empty, d2), x)) \\ = & radd_A(xeval(ass)_{string}(radd(empty, d2)), xeval(ass)_{data}(x)) \\ = & radd_A(radd_A(xeval(ass)_{string}(empty), xeval(ass)_{data}(d2)), xeval(ass)_{data}(x)) \\ = & radd_A(radd_A(empty_A, d2_A), (ass)_{data}(x)) \\ = & radd_A(radd_A(\lambda, b), a) \\ = & radd_A(b, a) \\ = & ba \end{array}$$

Lösung zu Aufgabe 2:

Auswertung des Terms $make - l(new - line, make - l(new - line, radd(empty, y)))$:

$$\begin{array}{rl} & xeval(ass)_{line}(make - l(new - line, make - l(new - line, radd(empty, y)))) \\ = & make - l_A(xeval(ass)_{line}(new - line), xeval(ass)_{line}(make - l(new - line, radd(empty, y)))) \\ = & make - l_A(new - line_A, make - l_A(xeval(ass)_{line}(new - line), xeval(ass)_{string}(radd(empty, y)))) \\ = & make - l_A(new - line_A, make - l_A(new - line_A, radd_A(xeval(ass)_{string}(empty), xeval(ass)_{data}(y)))) \\ = & make - l_A(new - line_A, make - l_A(new - line_A, radd_A(empty_A, (ass)_{data}(y)))) \\ = & make - l_A(\triangleright, make - l_A(\triangleright, radd_A(\lambda, b))) \\ = & make - l_A(\triangleright, make - l_A(\triangleright, b)) \\ = & make - l_A(\triangleright, b\triangleright) \\ = & b\triangleright \end{array}$$

Lösung zu Aufgabe 3:

Die Variablendefinition sei dieselbe wie aus Aufgabe 1. Sei $ass_{data} : X_{data} \Rightarrow A_{data}$ eine beliebige Variablenbelegung.

Gezeigt werden soll $A \models e_1$ mit

$$e_1 : make - l(new - line, radd(empty, x) = insert(new - line, x)$$

$A \models e_1$ gilt gena dann, wenn für alle Variablenbelegungen $ass : X \Rightarrow A$ gilt:

$$xeval(ass)_{line}(make - l(new - line, radd(empty, x)) = xeval(ass)_{line}(insert(new - line, x))$$

Beweis:

$$\begin{array}{rcl}
xeval(ass)_{line}(make - l(new - line, radd(empty, x)) & = & xeval(ass)_{line}(insert(new - line, x)) \\
\Leftrightarrow make - l_A(xeval(ass)_{line}(new - line), & = & insert_A(xeval(ass)_{line}(new - line), xeval(ass)_{data}(x) \\
xeval(ass)_{string}(radd(empty, x)) & & \\
\Leftrightarrow make - l_A(new - line_A, radd_A(xeval(ass)_{string}(empty), & = & insert_A(new - line_A, ass_{data}(x)) \\
xeval(ass)_{data}(x))) & & \\
\Leftrightarrow make - l_A(new - line_A, radd_A(empty_A, ass_{data}(x))) & = & insert_A(new - line_A, ass_{data}(x))
\end{array}$$

- 1. Fall: $ass_{data}(x) = a$

$$\begin{array}{rcl}
make - l_A(new - line_A, radd_A(empty_A, a)) & = & insert_A(new - line_A, a) \\
\Leftrightarrow make - l_A(\triangleright, radd_A(\lambda, a)) & = & insert_A(\triangleright, a) \\
\Leftrightarrow make - l_A(\triangleright, a) & = & a \triangleright \\
\Leftrightarrow a\triangleright & = & a\triangleright
\end{array}$$

- 2. Fall: $ass_{data}(x) = b$ (Beweis analog zum 1. Fall)

$$\begin{array}{rcl}
make - l_A(new - line_A, radd_A(empty_A, b)) & = & insert_A(new - line_A, b) \\
\Leftrightarrow make - l_A(\triangleright, radd_A(\lambda, b)) & = & insert_A(\triangleright, b) \\
\Leftrightarrow make - l_A(\triangleright, b) & = & b \triangleright \\
\Leftrightarrow b\triangleright & = & b\triangleright
\end{array}$$

$\Rightarrow A \models e_1$

Damit ist der notwendige Nachweis erbracht.

Lösung zu Aufgabe 4:

(a) $A \models e1$ gilt. Beweis:

$A \models e1 \Leftrightarrow$ für alle möglichen Variablenbelegungen ass : $X \rightarrow A$ gilt:

$$\begin{array}{rl}
 & xeval(\text{ass})_{Stack}(pop(push(d, s))) = xeval(\text{ass})_{Stack}(s) \\
\Leftrightarrow & pop_A(xeval(\text{ass})_{Stack}(push(d, s))) = xeval(\text{ass})_{Stack}(s) \\
\Leftrightarrow & pop_A(push_A(xeval(\text{ass})_{Data}(d), xeval(\text{ass})_{Stack}(s))) = xeval(\text{ass})_{Stack}(s) \\
\Leftrightarrow & pop_A(xeval(\text{ass})_{Stack}(s)xeval(\text{ass})_{Data}(d))) = xeval(\text{ass})_{Stack}(s) \\
\Leftrightarrow & xeval(\text{ass})_{Stack}(s) = xeval(\text{ass})_{Stack}(s)
\end{array}$$

Also gilt: $A \models e1$.

(b) $A \models e2$ gilt nicht. Beweis:

$A \models e2 \Leftrightarrow$ für alle möglichen Variablenbelegungen ass : $X \to A$ gilt:

$$\begin{aligned}
& \mathit{xeval}(\text{ass})_{\mathit{Stack}}(\mathit{push}(\mathit{top}(s), \mathit{pop}(s))) = \mathit{xeval}(\text{ass})_{\mathit{Stack}}(s) \\
\Leftrightarrow\ & \mathit{push}_A(\mathit{xeval}(\text{ass})_{\mathit{Data}}(\mathit{top}(s)), \mathit{xeval}(\text{ass})_{\mathit{Stack}}(\mathit{pop}(s)) = \mathit{xeval}(\text{ass})_{\mathit{Stack}}(s) \\
\Leftrightarrow\ & \mathit{push}_A(\mathit{top}_A(\mathit{xeval}(\text{ass})_{\mathit{Stack}}(s)), \mathit{pop}_A(\mathit{xeval}(\text{ass})_{\mathit{Stack}}(s))) = \mathit{xeval}(\text{ass})_{\mathit{Stack}}(s)
\end{aligned}$$

Da diese Aussage für alle möglichen Belegungen gelten muss, nehmen wir eine konkrete Belegung an und führen die Aussage zum Widerspruch.

Sei $s \mapsto \lambda$. Dann:

$$\begin{aligned}
\Rightarrow\ & \mathit{push}_A(\mathit{top}_A(\lambda), \mathit{pop}_A(\lambda)) = \lambda \\
\Rightarrow\ & \mathit{push}_A(0, \lambda) = \lambda \\
\Rightarrow\ & 0 = \lambda
\end{aligned}$$

Damit liegt ein Widerspruch vor und $e2$ gilt nicht in A.

(c) $B \models e1$ gilt nicht. Beweis:

$B \models e1 \Leftrightarrow$ für alle möglichen Variablenbelegungen ass : $X \to B$ gilt:

$$\begin{aligned}
& \mathit{xeval}(\text{ass})_{\mathit{Stack}}(\mathit{pop}(\mathit{push}(d, s))) = \mathit{xeval}(\text{ass})_{\mathit{Stack}}(s) \\
\Leftrightarrow\ & \mathit{pop}_B(\mathit{xeval}(\text{ass})_{\mathit{Stack}}(\mathit{push}(d, s))) = \mathit{xeval}(\text{ass})_{\mathit{Stack}}(s) \\
\Leftrightarrow\ & \mathit{pop}_B(\mathit{push}_B(\mathit{xeval}(\text{ass})_{\mathit{Data}}(d), \mathit{xeval}(\text{ass})_{\mathit{Stack}}(s))) = \mathit{xeval}(\text{ass})_{\mathit{Stack}}(s)
\end{aligned}$$

Da diese Aussage für alle möglichen Belegungen gelten muss, nehmen wir eine konkrete Belegung an und führen die Aussage zum Widerspruch.

Sei $s \mapsto (v, \lambda), v \in \{0, \ldots, 9\}^*$ und $d \mapsto 0$. Dann:

$$\begin{aligned}
\Rightarrow\ & \mathit{pop}_B(\mathit{push}_B(0, (v, \lambda))) = (v, \lambda) \\
\Rightarrow\ & \mathit{pop}_B(v0, \lambda) = (v, \lambda) \\
\Rightarrow\ & (v, 0\lambda) = (v, \lambda)
\end{aligned}$$

Damit liegt ein Widerspruch vor und $e1$ gilt nicht allgemein in B.

(d) $B \models e2$ gilt nicht. Beweis:

$B \models e2 \Leftrightarrow$ für alle möglichen Variablenbelegungen ass : $X \to B$ gilt:

$$\begin{aligned}
& \mathit{xeval}(\text{ass})_{\mathit{Stack}}(\mathit{push}(\mathit{top}(s), \mathit{pop}(s))) = \mathit{xeval}(\text{ass})_{\mathit{Stack}}(s) \\
\Leftrightarrow\ & \mathit{push}_B(\mathit{xeval}(\text{ass})_{\mathit{Data}}(\mathit{top}(s)), \mathit{xeval}(\text{ass})_{\mathit{Stack}}(\mathit{pop}(s)) = \mathit{xeval}(\text{ass})_{\mathit{Stack}}(s) \\
\Leftrightarrow\ & \mathit{push}_B(\mathit{top}_B(\mathit{xeval}(\text{ass})_{\mathit{Stack}}(s)), \mathit{pop}_B(\mathit{xeval}(\text{ass})_{\mathit{Stack}}(s))) = \mathit{xeval}(\text{ass})_{\mathit{Stack}}(s)
\end{aligned}$$

Da diese Aussage für alle möglichen Belegungen gelten muss, nehmen wir eine konkrete Belegung an und führen die Aussage zum Widerspruch.

Sei $s \mapsto (\lambda, 0)$. Dann:

$$\begin{aligned}
\Rightarrow\ & \mathit{push}_B(\mathit{top}_B((\lambda, 0)), \mathit{pop}_B((\lambda, 0))) = (\lambda, 0) \\
\Rightarrow\ & \mathit{push}_B(0, (\lambda, 0)) = (\lambda, 0) \\
\Rightarrow\ & (0, \lambda) = (\lambda, 0)
\end{aligned}$$

Damit liegt ein Widerspruch vor und $e2$ gilt nicht allgemein in B.