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

SS 2002
21.5.2002

Lösungen zum 4. Übungsblatt TheGI2

Lösung zu Aufgabe 1:

Die verwendeten Algebren:

Σ	A	B
$data$	$A_{data} := \{a, b\}$	$B_{data} := \{\mathbb{N}\}$
$string$	$A_{string} := \{a, b\}^*$	$B_{string} := \{\mathbb{N}\}^*$
$line$	$A_{line} := \{\#\}$	$B_{line} := \{\#\}$
$d1 :\rightarrow data$	$d1_A :\rightarrow A_{data}$ $\rightarrow b$	$d1_B :\rightarrow B_{data}$ $\rightarrow 0$
$d2 :\rightarrow data$	$d2_A :\rightarrow A_{data}$ $\rightarrow a$	$d2_B :\rightarrow B_{data}$ $\rightarrow 1$
$empty :\rightarrow string$	$empty_A :\rightarrow A_{string}$ $\rightarrow \lambda$	$empty_B :\rightarrow B_{string}$ $\rightarrow \lambda$
$radd : string\ data \rightarrow string$	$radd_A : A_{string}\ A_{data} \rightarrow A_{string}$ $(v, w) \rightarrow vw$	$radd_B : B_{string}\ B_{data} \rightarrow B_{string}$ $(v, w) \rightarrow vw$
$newline :\rightarrow line$	$newline_A :\rightarrow A_{line}$ $\rightarrow \#$	$newline_B :\rightarrow B_{line}$ $\rightarrow \#$
$make - l : line\ string \rightarrow line$	$make - l_A : A_{line}\ A_{string} \rightarrow A_{line}$ $(\#, v) \rightarrow \#$	$make - l_B : B_{line}\ B_{string} \rightarrow B_{line}$ $(\#, v) \rightarrow \#$
$insert : line\ data \rightarrow line$	$insert_A : A_{line}\ A_{data} \rightarrow A_{line}$ $(u, v) \rightarrow \#$	$insert_B : B_{line}\ B_{data} \rightarrow B_{line}$ $(u, v) \rightarrow \#$

(a) Es existiert ein Homomorphismus $h : A \rightarrow B$.

s	$(h_s : A_s \rightarrow B_s)_{s \in S}$
$data$	$h_{data} = \{b \rightarrow 0, a \rightarrow 1\}$
$string$	$h_{string}(v) = \begin{cases} \lambda & v = \lambda \\ 1h_{string}(w) & v = bw \\ 0h_{string}(w) & v = aw \end{cases}$ wobei $w = \{a, b\}^*$
$line$	$h_{line} = \{\# \rightarrow \#\}$

Beweis:

- Konstantensymbole

$$h_{data}(d1_A) = h_{data}(b) = 0 = d1_B$$

$$h_{data}(d2_A) = h_{data}(a) = 1 = d2_B$$

$$h_{string}(empty_A) = h_{string}(\lambda) = \lambda = empty_B$$

$$h_{line}(newline_A) = h_{line}(\#) = \# = newline_B$$

- Operationen
  - radd
    $\forall w \in A_{string}, y \in A_{data}$:

$$h_{string}(radd_A(y, w)) = h_{string}(yw) = \begin{cases} 1h_{string}(v) & w = bv \\ 0h_{string}(v) & w = av \end{cases}$$

$$radd_B(h_{data}(y), h_{string}(w)) = \begin{cases} radd_B(0, h_{string}(w)) = 0h_{string}(w) & y = b \\ radd_B(1, h_{string}(w)) = 1h_{string}(w) & y = a \end{cases}$$

- make-l

$$h_{line}(make - l_A(\#, v)) = h_{line}(\#) = \#$$

$$make - l_B(h_{line}(\#), h_{string}(v)) = make - l_B(\#, h_{string}(v)) = \#$$

- insert

$$h_{line}(insert_A(\#, v)) = h_{line}(\#) = \#$$

$$insert_B(h_{line}(\#), h_{data}(v)) = insert_B(\#, h_{data}(v)) = \#$$

Damit ist gezeigt dass ein solcher Homomorphismus existiert.

(b) Es existiert ein Homomorphismus $h : B \to A$.

s	$(h_s : B_s \to A_s)_{s \in S}$
$data$	$h_{data} = \{0 \to b, 1 \to a\}$
$string$	$h_{string}(v) = \begin{cases} \lambda & v = \lambda \\ ah_{string}(w) & v \text{ ungerade} \\ bh_{string}(w) & v \text{ gerade} \end{cases}$ wobei $w = x\{\mathbb{N}\}^*, x \in \mathbb{N}$
$line$	$h_{line} = \{\# \to \#\}$

Beweis:

- Konstantensymbole

$$h_{data}(d1_B) = h_{data}(0) = b = d1_A$$

$$h_{data}(d2_B) = h_{data}(1) = a = d2_A$$

$$h_{string}(empty_B) = h_{string}(\lambda) = \lambda = empty_A$$

$$h_{line}(newline_B) = h_{line}(\#) = \# = newline_A$$

- Operationen
  - radd
    $\forall w \in B_{string}, y \in B_{data}$:

$$h_{string}(radd_B(y, w)) = h_{string}(yw) = \begin{cases} ah_{string}(v) & w = sv,\ s \text{ ungerade} \\ bh_{string}(v) & w = sv,\ s \text{ gerade} \end{cases}$$

$$radd_A(h_{data}(y), h_{string}(w)) = \begin{cases} radd_A(0, h_{string}(w)) = bh_{string}(w) & y = gerade \\ radd_A(1, h_{string}(w)) = ah_{string}(w) & y = ungerade \end{cases}$$

  - make-l

$$h_{line}(make - l_B(\#, v)) = h_{line}(\#) = \#$$

$$make - l_A(h_{line}(\#), h_{string}(v)) = make - l_A(\#, h_{string}(v)) = \#$$

  - insert

$$h_{line}(insert_B(\#, v)) = h_{line}(\#) = \#$$

$$insert_A(h_{line}(\#), h_{data}(v)) = insert_A(\#, h_{data}(v)) = \#$$

Damit ist gezeigt dass ein solcher Homomorphismus existiert.

Lösung zu Aufgabe 2:

(a)

$$
\begin{aligned}
& eval(A)_{Data}\left(top\left(push\left(d1, empty\right)\right)\right) \\
= \;& top_A\left(eval(A)_{Stack}\left(push\left(d1, empty\right)\right)\right) \\
= \;& top_A\left(push_A\left(eval(A)_{Data}\left(d1\right), eval(A)_{Stack}\left(empty\right)\right)\right) \\
= \;& top_A\left(push_A\left(d1_A, empty_A\right)\right) \\
= \;& top_A\left(push_A\left(0, \lambda\right)\right) \\
= \;& top_A(0) \\
= \;& 0
\end{aligned}
$$

$$
\begin{aligned}
& eval(A)_{Data}\left(top\left(clean\left(pop\left(empty\right)\right)\right)\right) \\
= \;& top_A\left(eval(A)_{Stack}\left(clean\left(pop\left(empty\right)\right)\right)\right) \\
= \;& top_A\left(clean_A\left(eval(A)_{Stack}\left(pop\left(empty\right)\right)\right)\right) \\
= \;& top_A\left(clean_A\left(pop_A\left(eval(A)_{Stack}\left(empty\right)\right)\right)\right) \\
= \;& top_A\left(clean_A\left(pop_A\left(empty_A\right)\right)\right) \\
= \;& top_A\left(clean_A\left(pop_A\left(\lambda\right)\right)\right) \\
= \;& top_A\left(clean_A\left(\lambda\right)\right) \\
= \;& top_A\left(\lambda\right) \\
= \;& 0
\end{aligned}
$$

(b)

$$
\begin{aligned}
& eval(A)_{Stack}\left(clean\left(pop\left(push\left(d1, empty\right)\right)\right)\right) \\
= \;& clean_A\left(eval(A)_{Stack}\left(pop\left(push\left(d1, empty\right)\right)\right)\right) \\
= \;& clean_A\left(pop_A\left(eval(A)_{Stack}\left(push\left(d1, empty\right)\right)\right)\right) \\
= \;& clean_A\left(pop_A\left(push_A\left(eval(A)_{Data}\left(d1\right), eval(A)_{Stack}\left(empty\right)\right)\right)\right) \\
= \;& clean_A\left(pop_A\left(push_A\left(d1_A, empty_A\right)\right)\right) \\
= \;& clean_A\left(pop_A\left(push_A\left(0, \lambda\right)\right)\right) \\
= \;& clean_A\left(pop_A\left(0\right)\right) \\
= \;& clean_A\left(\lambda\right) \\
= \;& \lambda
\end{aligned}
$$

$$
\begin{aligned}
& eval(A)_{Stack}\left(push\left(d2, pop\left(push\left(d2, empty\right)\right)\right)\right) \\
= \;& push_A\left(eval(A)_{Data}\left(d2\right), eval(A)_{Stack}\left(pop\left(push\left(d2, empty\right)\right)\right)\right) \\
= \;& push_A\left(d2_A, pop_A\left(eval(A)_{Stack}\left(push\left(d2, empty\right)\right)\right)\right) \\
= \;& push_A\left(d2_A, pop_A\left(push_A\left(eval(A)_{Data}\left(d2\right), eval(A)_{Stack}\left(empty\right)\right)\right)\right) \\
= \;& push_A\left(d2_A, pop_A\left(push_A\left(d2_A, empty_A\right)\right)\right) \\
= \;& push_A\left(1, pop_A\left(push_A\left(1, \lambda\right)\right)\right) \\
= \;& push_A\left(1, pop_A\left(1\right)\right) \\
= \;& push_A\left(1, \lambda\right) \\
= \;& 1
\end{aligned}
$$

(c)

$$
\begin{aligned}
& eval(B)_{Stack}\left(clean\left(push\left(d2, pop\left(push\left(d2, empty\right)\right)\right)\right)\right) \\
= \;& clean_B\left(eval(B)_{Stack}\left(push\left(d2, pop\left(push\left(d2, empty\right)\right)\right)\right)\right) \\
= \;& clean_B\left(push_B\left(eval(B)_{Data}\left(d2\right), eval(A)_{Stack}\left(pop\left(push\left(d2, empty\right)\right)\right)\right)\right) \\
= \;& clean_B\left(push_B\left(d2_B, pop_B\left(eval(B)_{Stack}\left(push\left(d2, empty\right)\right)\right)\right)\right) \\
= \;& clean_B\left(push_B\left(d2_B, pop_B\left(push_B\left(eval(B)_{Data}\left(d2\right), eval(B)_{Stack}\left(empty\right)\right)\right)\right)\right) \\
= \;& clean_B\left(push_B\left(d2_B, pop_B\left(push_B\left(d2_B, empty_B\right)\right)\right)\right) \\
= \;& clean_B\left(push_B\left(1, pop_B\left(push_B\left(1, (\lambda, \lambda)\right)\right)\right)\right) \\
= \;& clean_B\left(push_B\left(1, pop\left(1, \lambda\right)\right)\right) \\
= \;& clean_B\left(push_B\left(1, (\lambda, 1)\right)\right) \\
= \;& clean_B\left(1, \lambda\right) \\
= \;& (1, \lambda)
\end{aligned}
$$

$$
\begin{aligned}
& eval(B)_{Stack}\left(clean\left(pop\left(push\left(d2, push\left(d2, empty\right)\right)\right)\right)\right) \\
= \;& clean_B\left(eval(B)_{Stack}\left(pop\left(push\left(d2, push\left(d2, empty\right)\right)\right)\right)\right) \\
= \;& clean_B\left(pop_B\left(eval(B)_{Stack}\left(push\left(d2, push\left(d2, empty\right)\right)\right)\right)\right) \\
= \;& clean_B\left(pop_B\left(push_B\left(eval(B)_{Data}\left(d2\right), eval(B)_{Stack}\left(push\left(d2, empty\right)\right)\right)\right)\right) \\
= \;& clean_B\left(pop_B\left(push_B\left(d2_B, push_B\left(eval(B)_{Data}\left(d2\right), eval(B)_{Stack}\left(empty\right)\right)\right)\right)\right) \\
= \;& clean_B\left(pop_B\left(push_B\left(d2_B, push_B\left(d2_B, empty_B\right)\right)\right)\right) \\
= \;& clean_B\left(pop_B\left(push_B\left(1, push_B\left(1, (\lambda, \lambda)\right)\right)\right)\right) \\
= \;& clean_B\left(pop_B\left(push_B\left(1, (1, \lambda)\right)\right)\right) \\
= \;& clean_B\left(pop_B\left(11, \lambda\right)\right) \\
= \;& clean_B\left(1, 1\right) \\
= \;& (1, \lambda)
\end{aligned}
$$