Einführung in den π-Kalkül: Übungen zur Vorlesung

Michael Arndt **Blatt 6**
Universität Tübingen, WSI **SS 2004**

Aufgabe 1

Ein Puffer der Kapazität 1 sei durch folgende Gleichungen erklärt:

$$\begin{aligned} B_e^{(1)}(l,r) &\stackrel{def}{=} l(x).B_f^{(1)}\langle l,r,x\rangle \\ B_f^{(1)}(l,r,x) &\stackrel{def}{=} \overline{r}\langle x\rangle.B_e^{(1)}\langle l,r\rangle \end{aligned}$$

Ein Puffer der Kapazität 2 ist dann durch folgende Definition gegeben:

$$B_e^{(2)}(l,r) \stackrel{def}{=} (\pi m)(B_e^{(1)}\langle l,m\rangle \parallel B_e^{(1)}\langle m,r\rangle)$$

Geben Sie für den Prozeß

$$P = \bar{l}\langle x_1\rangle.\bar{l}\langle x_2\rangle.\bar{l}\langle x_3\rangle \parallel B_e^{(2)}\langle l,r\rangle \parallel r(y_1).r(y_2).r(y_3).Q$$

den Prozeß $\widetilde{P}$ an, in dem alle Definitionsgleichungen direkt durch Replikationsausdrücke repräsentiert sind. Überzeugen Sie sich (informell) davon, daß $\widetilde{P} \approx P$.

Aufgabe 2

Abstrakte Datentypen lassen sich im π-Kalkül auf besonders eindrückliche Weise darstellen, weil mit der Charakterisierung durch einen π-Ausdruck auch sofort eine dynamische Realisierung erklärt ist. Jedes Exemplar einer Datenstruktur muß über eine eindeutige *Adresse* oder ein *Handle* verfügen. Deswegen ist ein Datentyp selbst immer durch Prozeßabstraktionen erklärt. Die Definitionsgleichungen haben dabei Prozeßabstraktionen als Parameter; auf diese Weise wird das Zusammenfügen von Daten modelliert.

Für Listen bietet sich folgende Definition an:

$$\begin{aligned} \mathbf{Nil} &\stackrel{def}{=} (h).h(n,c).\overline{n} \\ [V|L] &\stackrel{def}{=} (h).(\pi v,l)(h(n,c).\overline{c}\langle v,l\rangle \parallel V\langle v\rangle \parallel L\langle l\rangle) \\ [V_1,\ldots,V_n] &\stackrel{def}{=} [V_1|\ldots[V_n|\mathbf{Nil}]\ldots] \end{aligned}$$

Durch folgende Definitionsgleichung ist ein Konditional erklärt, das, zusammen mit einer Liste an demselben Handle, zum Prozeß P bzw. zur Applikation $F\langle v,l\rangle$ reagiert.

$$\mathbf{ListCond}(P,F) \stackrel{def}{=} (h).(\pi n,c)\overline{h}\langle n,c\rangle.(n.P + c(v,l).F\langle v,l\rangle)$$

(a) Geben Sie den Prozeßausdruck an, der zu der Liste $[G,H,[G,I]]$ gehört, wobei die G,H,I Prozeßabstraktionen sind, die beliebige andere Datenstrukturen repräsentieren. Beschreiben Sie, wie das doppelte Vorkommen von G behandelt wird.

(b) Zeigen Sie, daß folgende Reaktionen stattfinden können:

$$\begin{aligned} \mathbf{Nil}\langle k\rangle \parallel \mathbf{ListCond}(P,F)\langle k\rangle &\to^* P \\ [V|L]\langle k\rangle \parallel \mathbf{ListCond}(P,F)\langle k\rangle &\to^* (\pi v,l)(F\langle v,l\rangle \parallel V\langle v\rangle \parallel L\langle l\rangle) \end{aligned}$$

(c) Geben Sie einen Prozeßausdruck an, der aus einer Liste das zweite Element (also dessen Adresse!) ausgibt, sofern es ein zweites Element gibt, ansonsten einen Fehler an einem dafür reservierten Fehlerkanal e signalisiert.