Theorie der Programmierung SoSe 2024

Klausur [Probe]

Aufgabe 1 Konfluenz und Terminierung (19 Punkte)

Wir definieren ein Termersetzungssystem über der aus den beiden unären Funktionssymbolen f und g sowie den beiden binären Funktionssymbolen $\oplus$ und $\sqcap$ bestehenden Signatur Σ durch

$$\begin{aligned} f(x \oplus y) &\longrightarrow_0 f(y) \sqcap f(x) && (1)\\ x \sqcap (y \sqcap z) &\longrightarrow_0 (x \sqcap y) \sqcap z && (2)\\ (x \oplus y) \oplus z &\longrightarrow_0 x \oplus (y \oplus z) && (3)\\ g(x) &\longrightarrow_0 f(f(f(x))) && (4) \end{aligned}$$

1. Zeigen Sie mittels Polynomordnung, dass das System stark normalisierend ist. (10 P.)

   Hinweis: Verwenden Sie $p_f(x) = x^2$. Für $p_\oplus$ und $p_\sqcap$ können Sie jeweils lineare Polynome wählen.

2. Entscheiden Sie (mit Begründung), ob das gegebene System konfluent ist. (9 P.)

   Hinweis: Das System hat genau drei kritische Paare.

Aufgabe 2 System F (20 Punkte)

Man erinnere sich, dass der parametrische Typ der Listen in System F als

$$\mathbb{L}\ a := \forall\ r.\ r \to (a \to r \to r) \to r$$

kodiert wird. Wir betrachten folgende Funktionen auf Listen:

$$\begin{aligned} &\mathit{nil}\ :\ \forall\ a.\ \mathbb{L}\ a && \mathit{snoc}\ :\ \forall\ a.\ a \to \mathbb{L}\ a \to \mathbb{L}\ a\\ &\mathit{nil}\ = \lambda u\, f.\ u && \mathit{snoc} = \lambda x\, l.\ \lambda u\, f.\ l\ (f\ x\ u)\ f \end{aligned}$$

Dabei ist nil eine leere Liste und snoc hängt ein Element hinten an eine Liste heran. Wir wollen nun den angegebenen Typ von snoc im leeren Kontext herleiten. Rückwärts vom Ziel ausgehend verbleibt nach sechs Regelanwendungen das folgende Teilziel:

$$x : a,\ l : \mathbb{L}\ a,\ u : r,\ f : a \to r \to r \vdash l\ (f\ x\ u)\ f : r$$

1. Geben Sie die sechs angewendeten Regeln an, rückwärts vom Ziel ausgehend. Es genügt hierbei ausdrücklich die Angabe der Regelnamen. (3 P.)

2. Vervollständigen Sie nun den Beweis, indem Sie die verbleibende Typisierung herleiten, diesmal unter Angabe der vollständigen Berechnung (d.h. nicht nur der Regelnamen, sondern auch der Typisierungsurteile in allen Zwischenschritten). (9 P.)

3. Definieren Sie eine Funktion $\mathit{rev}\ :\ \forall\ a.\ \mathbb{L}\ a \to \mathbb{L}\ a$, die eine gegebene Liste umkehrt. Leiten Sie dann den Typ von rev im Kontext $\Gamma := \mathit{nil}\ :\ \forall\ a.\ \mathbb{L}\ a,\ \mathit{snoc}\ :\ \forall\ a.\ a \to \mathbb{L}\ a \to \mathbb{L}\ a$ her. (8 P.)

Aufgabe 3 Strukturelle Induktion und Folds (21 Punkte)

Wir betrachten im Folgenden einen parametrischen Typ *HList a b* von *heterogenen Listen*, also Listen, die sowohl Elemente vom Typ a als auch Elemente von Typ b (in beliebiger Kombination) enthalten können, sowie einige auf ihm definierte Funktionen:

$$
\begin{array}{l}
\textbf{data}\ \mathit{HList}\ a\ b\ \textbf{where} \\
\quad \mathit{HNil} : ()\ \to \mathit{HList}\ a\ b \\
\quad \mathit{ConsA} : a \to \mathit{HList}\ a\ b \to \mathit{HList}\ a\ b \\
\quad \mathit{ConsB} : b \to \mathit{HList}\ a\ b \to \mathit{HList}\ a\ b
\end{array}
$$

$$
\begin{array}{lcl}
\multicolumn{3}{l}{\mathit{mapA} : (a \to c) \to \mathit{HList}\ a\ b \to \mathit{HList}\ c\ b} \\
\mathit{mapA}\ f\ \mathit{HNil} & = & \mathit{HNil} \\
\mathit{mapA}\ f\ (\mathit{ConsA}\ x\ zs) & = & \mathit{ConsA}\ (f\ x)\ (\mathit{mapA}\ f\ zs) \\
\mathit{mapA}\ f\ (\mathit{ConsB}\ y\ zs) & = & \mathit{ConsB}\ y\ \ \ \ \ (\mathit{mapA}\ f\ zs)
\end{array}
$$

$$
\begin{array}{lcl}
\multicolumn{3}{l}{\mathit{mapB} : (b \to c) \to \mathit{HList}\ a\ b \to \mathit{HList}\ a\ c} \\
\mathit{mapB}\ g\ \mathit{HNil} & = & \mathit{HNil} \\
\mathit{mapB}\ g\ (\mathit{ConsA}\ x\ zs) & = & \mathit{ConsA}\ x\ \ \ \ \ (\mathit{mapB}\ g\ zs) \\
\mathit{mapB}\ g\ (\mathit{ConsB}\ y\ zs) & = & \mathit{ConsB}\ (g\ y)\ (\mathit{mapB}\ g\ zs)
\end{array}
$$

(6 P.) 1. Definieren Sie rekursiv eine Funktion $\mathit{swap} : \mathit{HList}\ a\ b \to \mathit{HList}\ b\ a$, sodass gilt

$$\forall\ zs.\ \ \mathit{swap}\ (\mathit{swap}\ zs) = zs$$

und zeigen Sie diese Eigenschaft anschließend per struktureller Induktion.

(7 P.) 2. Beweisen Sie mittels struktureller Induktion, dass

$$\forall\ f,\ g,\ zs.\ \ \mathit{mapA}\ f\ (\mathit{mapB}\ g\ zs) = \mathit{mapB}\ g\ (\mathit{mapA}\ f\ zs).$$

Hinweis: Die beiden Fälle für *ConsA* und *ConsB* sind sehr ähnlich zueinander. Es ist daher genügend, wenn Sie nur einen dieser Fälle behandeln.

(4 P.) 3. Geben Sie den Typ und die Definition der Fold-Funktion *foldH* von *HList a b* an.

(4 P.) 4. Geben Sie alternative Definitionen von *mapA* und *mapB* unter Verwendung von *foldH* an.

Aufgabe 4 Korekursion und Koinduktion (18 Punkte)

Wir erinnern uns an den Kodatentyp der Streams über einem Datentyp a

$$
\begin{array}{l}
\textbf{codata Stream}\ a\ \textbf{where} \\
\quad \mathit{hd} : \textbf{Stream}\ a \to a \\
\quad \mathit{tl}\ \ : \textbf{Stream}\ a \to \textbf{Stream}\ a
\end{array}
$$

und definieren korekursiv folgende Funktionen:

$$
\begin{array}{l}
\mathit{map} : (a \to b)\ \to \textbf{Stream}\ a \to \textbf{Stream}\ b \\
\mathit{hd}\ (\mathit{map}\ f\ s) = f\ (\mathit{hd}\ s) \\
\mathit{tl}\ \ (\mathit{map}\ f\ s) = \mathit{map}\ f\ (\mathit{tl}\ \ s)
\end{array}
$$

$$
\begin{array}{l}
\mathit{transpose}\ : \textbf{Stream}\ (\textbf{Stream}\ a) \to \textbf{Stream}\ (\textbf{Stream}\ a) \\
\mathit{hd}\ (\mathit{transpose}\ \ s)\ = \mathit{map}\ \mathit{hd}\ s \\
\mathit{tl}\ \ (\mathit{transpose}\ \ s)\ = \mathit{transpose}\ (\mathit{map}\ \mathit{tl}\ s)
\end{array}
$$

1. Vervollständigen Sie die folgende Gleichung, sodass diese für alle s : **Stream** (**Stream** a) gilt. Sie dürfen diese Eigenschaft im Folgenden ohne weiteren Beweis verwenden. (2 P.)

$$\mathit{map}\ \mathit{hd}\ (\mathit{transpose}\ s) = \tag{5}$$

2. Zeigen sie per Koinduktion, dass für alle s : **Stream** (**Stream** a) die folgende Eigenschaft gilt: (8 P.)

$$\mathit{map}\ \mathit{tl}\ (\mathit{transpose}\ \ s) = \mathit{transpose}\ (\ \mathit{tl}\ \ s) \tag{6}$$

3. Zeigen sie per Koinduktion, dass für alle s : **Stream** (**Stream** a) die folgende Eigenschaft gilt: (8 P.)

$$\mathit{transpose}\ \ (\mathit{transpose}\ \ s) = s \tag{7}$$

Aufgabe 5 Minimierung endlicher Automaten (12 Punkte)

Minimieren Sie den abgebildeten deterministischen endlichen Automaten über dem Alphabet $\Sigma = \{0, 1\}$ mittels des tabellarischen Algorithmus aus der Vorlesung und zeichnen Sie den Minimalautomaten.