Theorie der Programmierung SoSe 2025

Klausur [Probe]

Aufgabe 1 Konfluenz und Terminierung (20 Punkte)

1. Wir definieren ein Termersetzungssystem über der aus den beiden unären Funktionssymbolen f und g bestehenden Signatur Σ_1 durch

$$f(g(x)) \longrightarrow_0 g(f(x)) \tag{1}$$

   Welche der folgenden Optionen stellen polynomielle Interpretationen dar, mit deren Hilfe man die starke Normalisierung dieses Termersetzungssystems zeigen kann? Wir legen jeweils die Domäne $\{n \in \mathbb{N} \mid n \geq 2\}$ zugrunde. Begründen Sie jeweils kurz Ihre Antwort.

   (a) $p_f(x) = x^2$, $p_g(x) = x^3$ (2 P.)

   (b) $p_f(x) = 2x$, $p_g(x) = 2x + 1$ (2 P.)

2. Wir erweitern die Signatur Σ_1 um das binäre Funktionssymbol $\multimap$ und erhalten die Signatur Σ_2. Über dieser Signatur betrachten wir *zusätzlich zu* (1) die Regeln

$$f(x \multimap y) \longrightarrow_0 f(x) \multimap f(y) \tag{2}$$
$$g(x \multimap y) \longrightarrow_0 g(y) \multimap g(x) \tag{3}$$
$$x \multimap (y \multimap z) \longrightarrow_0 (x \multimap y) \multimap z \tag{4}$$

   Das erweiterte Termersetzungssystem hat vier nichttriviale kritische Paare; hier sind zwei dieser Paare:

$$\begin{tikzcd} & f(g(x \multimap y)) \arrow[dl, "(3)"'] \arrow[dr, "(1)"] & \\ f(g(y) \multimap g(x)) & & g(f(x \multimap y)) \end{tikzcd} \qquad \begin{tikzcd} & x' \multimap (x \multimap (y \multimap z)) \arrow[dl, "(4)"'] \arrow[dr, "(4)"] & \\ x' \multimap ((x \multimap y) \multimap z) & & (x' \multimap x) \multimap (y \multimap z) \end{tikzcd}$$

   (a) Zeigen Sie, dass die beiden gegebenen kritischen Paare zusammenführbar sind. (5 P.)

   (b) Finden Sie die zwei weiteren nichttrivialen kritischen Paare und zeigen Sie, dass auch diese zusammenführbar sind. (8 P.)

3. Wir erweitern das Termersetzungssystem zusätzlich um eine fünfte Regel (5 P.)

$$x \multimap f(y) \longrightarrow_0 f(x \multimap y) \tag{5}$$

   Zeigen Sie, dass das auf diese Weise erweiterterte System nicht stark normalisierend ist.

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:

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

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:

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

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

(9 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).

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

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:

$$
\begin{aligned}
&\textbf{data}\ HList\ a\ b\ \textbf{where}\\
&\quad HNil : ()\ \to HList\ a\ b\\
&\quad ConsA : a \to HList\ a\ b \to HList\ a\ b\\
&\quad ConsB : b \to HList\ a\ b \to HList\ a\ b\\[1ex]
&mapA : (a \to c) \to HList\ a\ b \to HList\ c\ b\\
&mapA\ f\ HNil = HNil\\
&mapA\ f\ (ConsA\ x\ zs) = ConsA\ (f\ x)\ (mapA\ f\ zs)\\
&mapA\ f\ (ConsB\ y\ zs) = ConsB\ y\ \ (mapA\ f\ zs)\\[1ex]
&mapB : (b \to c) \to HList\ a\ b \to HList\ a\ c\\
&mapB\ g\ HNil = HNil\\
&mapB\ g\ (ConsA\ x\ zs) = ConsA\ x\ \ (mapB\ g\ zs)\\
&mapB\ g\ (ConsB\ y\ zs) = ConsB\ (g\ y)\ (mapB\ g\ zs)
\end{aligned}
$$

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

$$\forall\ zs.\ \ swap\ (swap\ zs) = zs$$

und zeigen Sie diese Eigenschaft anschließend per struktureller Induktion.

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

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

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 (17 Punkte)

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

$$\begin{aligned}
&\textbf{codata Stream } a \textbf{ where}\\
&\quad hd : \textbf{Stream } a \to a\\
&\quad tl \;: \textbf{Stream } a \to \textbf{Stream } a
\end{aligned}$$

und definieren korekursiv folgende Funktionen:

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

$$\begin{aligned}
&prepend : a \to \textbf{Stream } a \to \textbf{Stream } a\\
&hd\ (prepend\ x\ s) = x\\
&tl\ \ (prepend\ x\ s) = prepend\ (hd\ s)\ (tl\ s)
\end{aligned}$$

$$\begin{aligned}
&transpose : \textbf{Stream } (\textbf{Stream } a) \to \textbf{Stream } (\textbf{Stream } a)\\
&hd\ (transpose\ s) = map\ hd\ s\\
&tl\ \ (transpose\ s) = transpose\ (map\ tl\ s)
\end{aligned}$$

1. Vervollständigen Sie die folgenden Gleichungen, sodass diese für alle s : **Stream** (**Stream** a) gelten. Sie dürfen diese Eigenschaften im Folgenden ohne weiteren Beweis verwenden. (4 P.)

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

$$map\ tl\ (transpose\ s) = \tag{7}$$

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

$$transpose\ (transpose\ s) = s \tag{8}$$

3. Wir betrachten die folgende korekursiv definierte Funktion: (5 P.)

$$\begin{aligned}
&f : \textbf{Stream } (\textbf{Stream } a) \to \textbf{Stream } a\\
&hd\ (f\ s) = hd\ (hd\ s)\\
&tl\ \ (f\ s) = f\ (transpose\ (prepend\ (tl\ (hd\ s))\ (tl\ s)))
\end{aligned}$$

Beschreiben Sie die durch f berechnete Funktion in eigenen Worten.

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.