Theorie der Programmierung SoSe 2016

Klausur [Probe]

Rev.13134

Bitte vermerken Sie auf Ihrer Abgabe Ihren Namen und Ihre Matrikelnummer.

Aufgabe 1 Konfluenz und Terminierung (9 Punkte)

Wir definieren ein Termersetzungssystem über der aus zwei binären Funktionssymbolen $\Uparrow$ und $\Downarrow$ (in Infixnotation geschrieben) bestehenden Signatur Σ durch

$$x \Uparrow (y \Uparrow z) \to_0 x \Uparrow (y \Downarrow y)$$
$$x \Downarrow (x \Downarrow y) \to_0 x \Downarrow y$$

1. Zeigen Sie mittels Polynomordnungen, dass das System stark normalisierend ist.
2. Ist das System konfluent? Geben Sie einen Beweis bzw. ein Gegenbeispiel an.

Aufgabe 2 Zahlen in System F (9 Punkte)

Man erinnere sich an folgende auf den Church-Numeralen definierte Funktionen:

$$\begin{aligned}
\mathit{zero} &= \lambda f\, a.\ a \\
\mathit{one} &= \lambda f\, a.\ f\ a \\
\mathit{two} &= \lambda f\, a.\ f\ (f\ a) \\
\mathit{three} &= \lambda f\, a.\ f\ (f\ (f\ a))
\end{aligned}$$

$$\mathit{succ}\ n = \lambda f\, a.\ f\ (n\ f\ a)$$

$$\begin{aligned}
\mathit{add}\ \ n\ m &= n\ \mathit{succ}\ m \\
\mathit{mult}\ n\ m &= m\ (\mathit{add}\ n)\ \mathit{zero}
\end{aligned}$$

$$\mathit{pow2}\ n = n\ (\mathit{mult}\ \mathit{two})\ \mathit{one}$$

1. Geben sie die ersten vier $\beta\delta$-Reduktionsschritte des Terms $\mathit{pow2}\ \mathit{three}$ unter a) normaler und b) applikativer Reduktion an. Markieren Sie (durch Unterstreichen) in jedem Schritt den zu reduzierenden Redex. *Hinweis:* Die Reduktionsstrategie bezieht sich ausdrücklich auch auf δ-Reduktion. D.h. auch δ-Reduktionen sind bei normaler Reduktion zuerst auf linke Seiten von Funktionsanwendungen (also die Funktionen) anzuwenden, von außen nach innen, und erst dann auf Argumente von Funktionsanwendungen, bei applikativer Reduktion dagegen zuerst auf Argumente von Funktionsanwendungen.
2. Man erinnere sich ferner, dass die Church-Numerale in System F den Typ $\mathbb{N} := \forall a.(a \to a) \to a \to a$ haben. Zeigen Sie unter der Annahme, dass *one* und *two* beide den Typ $\mathbb{N}$ haben und dass *mult* den Typ $\mathbb{N} \to \mathbb{N} \to \mathbb{N}$ hat, dass der Term $\lambda n.\ n\ (\mathit{mult}\ \mathit{two})\ \mathit{one}$ Typ $\mathbb{N} \to \mathbb{N}$ hat; d.h. geben sie eine Typherleitung in System F für

   $$\{\mathit{mult} : \mathbb{N} \to \mathbb{N} \to \mathbb{N}, \mathit{one} : \mathbb{N}, \mathit{two} : \mathbb{N}\} \vdash \lambda\ n.\ n\ (\mathit{mult}\ \ \mathit{two})\ \ \mathit{one} : \mathbb{N} \to \mathbb{N}$$

   an.

Aufgabe 3 Strukturelle Induktion (9 Punkte)

Wir erinnern an den Datentyp der Listen und einige hierauf rekursiv definierte Standardfunktionen:

data List $a = \mathit{Nil} \mid \mathit{Cons}\ a\ (\mathbf{List}\ a)$

$$\begin{aligned} \mathit{snoc}\ \mathit{Nil}\ y &= \mathit{Cons}\ y\ \mathit{Nil} \\ \mathit{snoc}\ (\mathit{Cons}\ x\ \mathit{xs})\ y &= \mathit{Cons}\ x\ (\mathit{snoc}\ \mathit{xs}\ y) \end{aligned}$$

$$\begin{aligned} \mathit{Nil} \oplus \mathit{ys} &= \mathit{ys} \\ (\mathit{Cons}\ x\ \mathit{xs}) \oplus \mathit{ys} &= \mathit{Cons}\ x\ (\mathit{xs} \oplus \mathit{ys}) \end{aligned}$$

Beweisen Sie mittels struktureller Induktion, dass

$$\forall e,\ \mathit{xs},\ \mathit{ys}.\ \mathit{xs} \oplus (\mathit{Cons}\ e\ \mathit{ys}) = (\mathit{snoc}\ \mathit{xs}\ e) \oplus \mathit{ys}.$$

Geben Sie im Induktionsschritt die Induktionsannahme explizit an, und erläutern Sie alle Schritte des Beweises.

Aufgabe 4 Korekursion und Koinduktion (9 Punkte)

Ein (digitales) *Signal* ist ein zeitlich veränderlicher Wert zwischen $-\infty$ und $+\infty$, wobei wir die Zeit als diskret behandeln, entsprechend etwa fortgesetztem Sampling. Solche Signale lassen sich in natürlicher Weise durch einen Kodatentyp repräsentieren:

codata *Signal* **where**
$\quad \mathit{currentSample} : \mathit{Signal} \to \mathbf{Int}$
$\quad \mathit{discardSample} : \mathit{Signal} \to \mathit{Signal}$

Z.B. repräsentiert gemäß den folgenden korekursiven Definitionen $\mathit{flat}\ x$ ein konstantes Signal mit Wert x und $\mathit{square}\ x\ y$ ein zwischen x und y alternierendes Signal:

$$\begin{aligned} \mathit{currentSample}\ (\mathit{flat}\ x) &= x \\ \mathit{discardSample}\ (\mathit{flat}\ x) &= \mathit{flat}\ x \end{aligned}$$

$$\begin{aligned} \mathit{currentSample}\ (\mathit{square}\ x\ y) &= x \\ \mathit{discardSample}\ (\mathit{square}\ x\ y) &= \mathit{square}\ y\ x \end{aligned}$$

1. Definieren Sie korekursiv eine Funktion $\mathit{sampler} : \mathit{Signal} \to \mathit{Signal} \to \mathit{Signal}$, so dass der Wert des Signals $\mathit{sampler}\ t\ s$ der Wert von s ist, wenn der Wert von t größer als 0 ist, und 0 sonst. Insbesondere sollten z.B. die Gleichungen

   $$\mathit{sampler}\ (\mathit{square}\ 0\ 1)\ (\mathit{square}\ x\ 0) = \mathit{flat}\ 0 \tag{1}$$

   und (für alle x)

   $$\mathit{sampler}\ (\mathit{square}\ 1\ 0)\ (\mathit{flat}\ x) = \mathit{square}\ x\ 0 \tag{2}$$

   gelten. *Hinweis:* Sie dürfen bei der Definition die üblichen Operationen auf Zahlen (Addition, Vergleich etc.) und auf Booleans (and, or, if-then-else etc.) als gegeben annehmen.

2. Beweisen Sie dann per Koinduktion, dass für Ihre Definition von *sampler* tatsächlich die beiden Gleichungen (1) und (2) gelten. Erläutern Sie alle Beweisschritte.

Aufgabe 5 Reguläre Ausdrücke (4 Punkte)

Sei L die Sprache der wohlgeformten „geschachtelten Paare“ über Buchstaben a and b; d.h. mit $\Sigma = \{a, b, ;, (,)\}$ ist L die Menge aller durch die Grammatik

$$s, t := a \mid b \mid (s; t)$$

erzeugten Worte über Σ. Z.B. gehören a, b, $(a; a)$, $((a; b); a)$ und $((b; a); (a; b))$ zu L, nicht aber ε, (a), $(a; b)(b; a)$ und $(a;)$. Ist L regulär? Wenn ja, geben sie einen regulären Ausdruck für L an und erläutern Sie, warum dieser L definiert; andernfalls zeigen Sie mittels des Pumping-Lemmas, dass L nicht regulär ist.