Teorija programskih jezikov: 3. izpit

8. julij 2020

1. naloga (20 točk)

V λ-računu, razširjenem s seznami, definirajmo:

$$sum = \mathtt{rec\,fun}\, s\; \ell.\, \mathtt{match}\, \ell\, \mathtt{with}\, [] \mapsto 0 \mid h :: t \mapsto h + s\, t$$

a) Zapišite vsa pravila, ki določajo operacijsko semantiko malih korakov za izraz $\mathtt{rec\,fun}\, f\; x.e$.

b) Zapišite vse korake v evalvaciji izraza $sum\,(19 :: (23 :: []))$ v semantiki malih korakov.

c) Izračunajte najbolj splošen tip izraza sum.

2. naloga (20 točk)

Naj bo $(D, \leq)$ domena in $x, y \in D$. Pravimo, da je x *daleč pod* y, kar označimo z $x \ll y$, kadar vsaka veriga, katere supremum preseže y, preseže x že po končno mnogo členih. Torej, če za vsako verigo $w_0 \leq w_1 \leq w_2 \leq \cdots$, za katero velja $y \leq \bigvee_i w_i$, obstaja nek j, da velja $x \leq w_j$.

a) Dokažite, da za poljubna $x, y \in D$ iz $x \ll y$ sledi $x \leq y$.

b) Dokažite, da za poljubne $x, y, z \in D$ iz $x \ll y$ in $y \leq z$ sledi $x \ll z$.

c) Dokažite, da za poljubne $x, y, z \in D$ iz $x \leq y$ in $y \ll z$ sledi $x \ll z$.

d) Poiščite primer domene $(D, \leq)$ ter elementa $x \leq y$, za katere *ne velja* $x \ll y$.

3. naloga (20 točk)

V λ-račun dodamo nedeterministično izvajanje, v katerem se lahko izrazi evalvirajo v več kot eno možno vrednost:

$$e ::= x \mid \mathtt{true} \mid \mathtt{false} \mid \mathtt{if}\, e\, \mathtt{then}\, e_1\, \mathtt{else}\, e_2 \mid \lambda x.e \mid e_1\, e_2 \mid e_1 \oplus e_2$$

a) Zapišite pravilo za določitev tipa za izraz $e_1 \oplus e_2$, ki se nedeterministično odloči, ali bo nadaljeval kot e_1 ali kot e_2.

b) Operacijsko semantiko za razširjeni λ-račun lahko podamo na dva načina. Prvi je, da v semantiko malih korakov dodamo pravili:

$$\frac{}{e_1 \oplus e_2 \rightsquigarrow e_1} \qquad \frac{}{e_1 \oplus e_2 \rightsquigarrow e_2}$$

Drugi pa je semantika velikih korakov oblike $e \Downarrow \{v_1, \ldots, v_n\}$, kjer so $v_1, \ldots, v_n$ vse možne vrednosti, v katere se lahko evalvira izraz e. Zapišite pravila, ki določajo takšno semantiko.

4. naloga (20 točk)

Polimorfni λ-račun oziroma *sistem F* je λ-račun, razširjen z eksplicitnimi univerzalno kvantificiranimi tipi ter izrazoma za abstrakcijo in aplikacijo tipov. Sintaksa njegovih tipov, izrazov in vrednosti je:

$$
\begin{aligned}
A &::= \texttt{bool} \mid \texttt{int} \mid A \to B \mid \alpha \mid \forall \alpha . A \\
e &::= \cdots \mid \Lambda \alpha . e \mid e\, A \\
v &::= \cdots \mid \Lambda \alpha . e
\end{aligned}
$$

pravila za operacijsko semantiko in tipe novih izrazov pa so

$$
\frac{e \rightsquigarrow e'}{e\, A \rightsquigarrow e'\, A}
\qquad
\frac{}{(\Lambda \alpha . e)\, A \rightsquigarrow e[A/\alpha]}
\qquad
\frac{\Gamma, \alpha \vdash e : A}{\Gamma \vdash \Lambda \alpha . e : \forall \alpha . A}
\qquad
\frac{\Gamma \vdash e : \forall \alpha . A}{\Gamma \vdash e\, B : A[B/\alpha]}
$$

pri čemer lahko konteksti Γ vsebujejo tako proste spremenljivke $x : A$ kot proste spremenljivke za tipe α. Poleg tega za vsak $\Gamma \vdash e : A$ zahtevamo, da se vse proste spremenljivke α v izrazu e in tipu A pojavijo v Γ.

Dokažite izreka o napredku in ohranitvi za polimorfni λ-račun.

Theory of programming languages: third exam

8 July 2020

Question 1 (20 marks)

In λ-calculus extended with lists, we define:

$$sum = \mathtt{rec\,fun}\, s\ \ell.\, \mathtt{match}\ \ell\ \mathtt{with}\, [\,] \mapsto 0 \mid h :: t \mapsto h + s\ t$$

a) Write down all the rules that specify the small-step operational semantics of the expression $\mathtt{rec\,fun}\, f\ x.e$.

b) Write down all the steps in the evaluation of the expression $sum\ (19 :: (23 :: [\,]))$ in the small-step semantics.

c) Compute the most general type of the expression sum.

Question 2 (20 marks)

Let $(D, \leq)$ be a domain and let $x, y \in D$. We say that x *is well below* y, written as $x \ll y$, when each chain whose supremum exceeds y, exceeds x after finitely many elements. In other words, if for any chain $w_0 \leq w_1 \leq w_2 \leq \cdots$ such that $y \leq \bigvee_i w_i$ there exists some j for which $x \leq w_j$.

a) Prove that $x \ll y$ implies $x \leq y$ for arbitrary $x, y \in D$.

b) Prove that $x \ll y$ and $y \leq z$ implies $x \ll z$ for arbitrary $x, y, z \in D$.

c) Prove that $x \leq y$ and $y \ll z$ implies $x \ll z$ for arbitrary $x, y, z \in D$.

d) Find an example of a domain $(D, \leq)$ and elements $x \leq y$ such that $x \ll y$ *does not hold*.

Question 3 (20 marks)

We extend λ-calculus with non-deterministic evaluation, where each expression can evaluate to more than one possible value:

$$e ::= x \mid \mathtt{true} \mid \mathtt{false} \mid \mathtt{if}\ e\ \mathtt{then}\ e_1\ \mathtt{else}\ e_2 \mid \lambda x.e \mid e_1\, e_2 \mid e_1 \oplus e_2$$

a) Write down the typing rule for the expression $e_1 \oplus e_2$, which non-deterministically chooses between proceeding as e_1 or as e_2.

b) Operational semantics for the extended λ-calculus can be given in two different ways. The first one is extending small-step semantics with rules:

$$\frac{}{e_1 \oplus e_2 \rightsquigarrow e_1} \qquad \frac{}{e_1 \oplus e_2 \rightsquigarrow e_2}$$

The second one is big step semantics of the form $e \Downarrow \{v_1, \ldots, v_n\}$, where $v_1, \ldots, v_n$ are all possible values into which e can evaluate. Write down all the rules that define such semantics.

Question 4 (20 marks)

Polymorphic λ-calculus or *system F* is λ-calculus, extended with explicit universally quantified types and expressions for type abstraction and application. The syntax of its types, expressions, and values is:

$$\begin{aligned} A &::= \texttt{bool} \mid \texttt{int} \mid A \to B \mid \alpha \mid \forall \alpha.A \\ e &::= \cdots \mid \Lambda \alpha.e \mid e\,A \\ v &::= \cdots \mid \Lambda \alpha.e \end{aligned}$$

The additional rules for operational semantics and typing judgements are:

$$\frac{e \rightsquigarrow e'}{e\,A \rightsquigarrow e'\,A} \qquad \frac{}{(\Lambda \alpha.e)A \rightsquigarrow e[A/\alpha]} \qquad \frac{\Gamma, \alpha \vdash e : A}{\Gamma \vdash \Lambda \alpha.e : \forall \alpha.A} \qquad \frac{\Gamma \vdash e : \forall \alpha.A}{\Gamma \vdash e\,B : A[B/\alpha]}$$

where the context Γ may contain both free variables of the form $x : A$ and free type variables α. We additionally require that in each $\Gamma \vdash e : A$ all free type variables α in the expression e or type A appear in Γ.

Prove progress and preservation theorems for polymorphic λ-calculus.