Göteborgs Universitet och Chalmers Tekniska Högskola 2000–01–01
Datavetenskap INN11

Tentamen i Beräkningsbarhet och lambda-kalkyl

Måndagen den 10 januari 1999, kl 8.45 – 13.45 i sal VV.
Ansvarig lärare: Bengt Nordström, tel 1033, eller 55 45 25 (hem)
Tillåtna hjälpmedel: Inga.
Börja varje uppgift på nytt blad. Skriv endast på en sida av papperet. Varje svar skall motiveras! Den här skriftliga tentamen utgör en del (75 %) av den totala examinationen, den andra delen (dvs. 25 %) består av de inlämingsuppgifter som har delats ut under kursens gång. För årets elever gäller alltså att summan av poängen från inlämningsuppgifterna och den skriftliga tentan skall vara minst 100 för att få godkänt på kursen. Examensvisning kommer att äga rum den 28 februari kl 14.15 i MD8.

1. Vad menas med att en funktion är Turing-beräkningsbar? (5)

2. Hur formulerar man och bevisar stopp-problemet för Turing-maskiner? (15)

3. Kan man räkna upp alla totala funktioner från $\mathbf{N}$ till $\{1\}$? Motivera! (10)

4. Kan man räkna upp alla partiella funktioner från $\mathbf{N}$ till $\{1\}$? Motivera! (10)

5. Låt A vara mängden $\{\mathbf{sur}, \mathbf{tot}, \mathbf{inj}\}$ och låt B vara mängden av partiella funktioner från $\mathbf{N}$ till $\mathbf{N}$. Definiera nu en relation P mellan B och A på följande sätt: Funktionen f är P-relaterad till elementet **sur** om f är surjektiv, den är P-relaterad till **tot** om den är total och P-relaterad till **inj** om den är injektiv. Låt nu A' beteckna mängden av alla delmängder av A. Vi definierar sen en relation $P3$ mellan B och A' på följande sätt: Funktionen g är $P3$-relaterad till elementet U om det för alla element x i U gäller att f är P-relaterad till x.

   (a) Relationen $P3$ är inte en partiell funktion från B till A, varför? och $\{\mathbf{sur}, \mathbf{tot}\}$.

   (b) Definiera en ny relation $P'3$ mellan B och A' som blir en funktionell version av $P3$.

   (20)

6. (a) Ge en induktiv definition av abstrakta syntaxen för programmen i λ-kalkyl.

   (b) Blir detta en uppräknelig mängd?

   (c) Definiera en förenklad substitutionsoperation för λ-kalkyl. Värdet av funktionen $\mathsf{closedsubst}(e, x, e')$ skall vara resultatet av att substituera uttrycket e' för all fria förekomster av variabeln x i uttrycket e. Uttrycket e' skall vara slutet.

   (d) Är $\mathsf{closedsubst}(e, x, e')$ alltid öppet om e är öppet? (30)

7. Visa hur man skall definiera språket PRL(k), mängden av primitivt rekursiva funktioner över heltalslistor. Ett element i mängden skall stå för en funktion som tar k stycken heltalslistor som argument och returnerar en heltalslista som resultat. Du skall ge den abstrakta syntaxen och den informella semantiken. (30)

   Kommentar: Språket skall definieras analogt med språket $\mathbf{PRF}_n$, mängden av primitivt rekursiva funktioner med n argument. Varje argument är ett naturligt tal. Det vill säga ett element f i mängden $\mathbf{PRF}_n$ står för en funktion i $\underbrace{\mathbf{N} \times \ldots \times \mathbf{N}}_{n \text{ stycken}} \to \mathbf{N}$. Elementen i mängden $\mathbf{PRF}_n$ byggs upp induktivt enligt följande klausuler:

$$
\begin{aligned}
\mathbf{z} &\in \mathbf{PRF}_1 \\
\mathbf{s} &\in \mathbf{PRF}_1 \\
\mathbf{proj}_i^n &\in \mathbf{PRF}_n \quad \text{om} \quad 1 \le i \le n \\
\mathbf{compose}(g, f_1, \ldots, f_m) &\in \mathbf{PRF}_n \quad \text{om} \quad g \in \mathbf{PRF}_m, f_i \in \mathbf{PRF}_n, 1 \le i \le m \\
\mathbf{rec}(g, h) &\in \mathbf{PRF}_{n+1} \quad \text{om} \quad g \in \mathbf{PRF}_n, h \in \mathbf{PRF}_{n+2}
\end{aligned}
$$

8. Skriv ett program $\mathbf{isnat}_1$ i språket χ med följande egenskap:

   $(\mathbf{isnat}_1\ n) = \mathbf{true}()$ om n är ett naturligt tal

   Försök sedan att skriva ett program $\mathbf{isnat}_2$ med egenskapen:

$$
(\mathbf{isnat}_2\ n) = \begin{cases} \mathbf{true}() & \text{om } n \text{ är ett naturligt tal,} \\ \mathbf{false}() & \text{för övrigt} \end{cases}
$$

   Föreslå en utvidgning av språket så att man lätt kan uttrycka sådana program. Föreslå ny konkret syntax, ny abstrakt syntax och ge (informell och formell) semantik för utvidgningen! (30)

   Kommentar: Språket χ har följande konkret syntax:

$(e_1\ e_2)$	applikation
$\lambda x.e$	abstraktion
$c\ e$	konstruerar-applikation
case e **of** $\{c_1 : e_1, \ldots\}$	case-uttryck
$[l_1 = e_1; \ldots]$	struktur
$e.l$	projektion
rec $x = e$ **end**	rekursion
x	variabel

Den abstrakta syntaxen är följande:

$$
\begin{aligned}
\mathbf{apply}(e_1,\ e_2) \in \chi \quad &\text{om} \quad e_1, e_2 \in \chi \\
\mathbf{lambda}(\mathrm{i}, e) \in \chi \quad &\text{om} \quad \mathrm{i} \in \mathbf{Id}, e \in \chi \\
\mathbf{constr}(\mathrm{i}, e) \in \chi \quad &\text{om} \quad \mathrm{i} \in \mathbf{Id}, e \in \chi \\
\mathbf{case}(e, \mathrm{t}) \in \chi \quad &\text{om} \quad e \in \chi, \mathrm{t} \in \mathbf{T}(\mathbf{Id}, \chi) \\
\mathbf{struct}(\mathrm{t}) \in \chi \quad &\text{om} \quad \mathrm{t} \in \mathbf{T}(\mathbf{Id}, \chi) \\
\mathbf{proj}(e, \mathrm{i}) \in \chi \quad &\text{om} \quad e \in \chi, \mathrm{i} \in \mathbf{Id} \\
\mathbf{rec}(\mathrm{i}, e) \in \chi \quad &\text{om} \quad e \in \chi, \mathrm{i} \in \mathbf{Id} \\
\mathbf{var}(\mathrm{i}) \in \chi \quad &\text{om} \quad \mathrm{i} \in \mathbf{Id}
\end{aligned}
$$

Semantiken defineras av:

$$\frac{e_1 \longrightarrow \mathbf{lambda}(\mathrm{i}, e_3) \qquad e_3[\mathrm{i} \leftarrow e_2] \longrightarrow \mathrm{d}}{\mathbf{apply}(e_1, e_2) \longrightarrow \mathrm{d}}$$

$$\frac{e \longrightarrow \mathbf{constr}(\mathrm{i}, e') \qquad \mathbf{lookup}(\mathrm{t}, \mathrm{i}, e'') \qquad \mathbf{apply}(e'', e') \longrightarrow \mathrm{d}}{\mathbf{case}(e, \mathrm{t}) \longrightarrow \mathrm{d}}$$

$$\frac{e \longrightarrow \mathbf{struct}(\mathrm{t}) \qquad \mathbf{lookup}(\mathrm{t}, \mathrm{i}, e') \qquad e' \longrightarrow \mathrm{d}}{\mathbf{proj}(e, \mathrm{i}) \longrightarrow \mathrm{d}}$$

$$\frac{e[\mathrm{i} \leftarrow \mathbf{rec}(\mathrm{i}, e)] \longrightarrow \mathrm{d}}{\mathbf{rec}(\mathrm{i}, e) \longrightarrow \mathrm{d}}$$

$$\mathbf{lambda}(\mathrm{i}, e) \longrightarrow \mathbf{lambda}(\mathrm{i}, e)$$

$$\mathbf{constr}(\mathrm{i}, e) \longrightarrow \mathbf{constr}(\mathrm{i}, e)$$

$$\mathbf{struct}(\mathrm{t}) \longrightarrow \mathbf{struct}(\mathrm{t})$$

Lycka till!