Göteborgs Universitet och Chalmers Tekniska Högskola
Datavetenskap

20 december 2002
INN111/TIN120

Tentamen i Beräkningsmodeller

Fredagen den 20 december 2002, kl 8.45 – 12.45.
Ansvarig lärare: Bengt Nordström, tel 1033 eller 0708 - 96 69 14.
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 inlämningsuppgifter som har delats ut under kursens gång. För årets och förra 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 fredagen den 10 januari kl 11.00 i MD 9. Lösningar till tentan kommer att finnas tillgängligt från kursens hemsida.

1. Finns det en injektiv och surjektiv funktion i $\mathrm{N} \tilde{\to} \mathrm{N}$ som inte är total? (5)

2. Ange om följande påståenden är sanna eller falska samt ge ett bevis för detta!

   (a) Alla partiella funktioner från N till $\{1\}$ är beräkningsbara. (10)

   (b) Alla totala funktioner från N till $\{1\}$ är beräkningsbara. (10)

   (c) Om variabeln x är fri i lambda-uttrycket e och om e's normalform är d så är x fri i d. (10)

3. Visa ett sätt att representera ett naturligt tal som en positionerad remsa! (5)

4. Konstruera en Turing-maskin som beräknar funktionen $a \in \mathrm{N} \to \mathrm{N}$ som definieras av $a(n) = n + 3$. (15)

5. Ge standard-representationen för programmen $\lambda x \to x$ och $\mathsf{True}\langle\rangle$ i χ. (15)

6. Det finns fem programkonstruktioner i språket PRF, de första fyra har en syntax som kan beskrivas informellt på följande sätt: (30)

$$
\begin{aligned}
\mathrm{z} &\in \mathrm{PRF}_0 \\
\mathrm{s} &\in \mathrm{PRF}_1 \\
\mathrm{proj}_i^n &\in \mathrm{PRF}_{n+1} \quad \text{if} \quad i \le n \\
\mathrm{comp}(g, f_1, \ldots, f_m) &\in \mathrm{PRF}_n \quad \text{if } g \in \mathrm{PRF}_m, f_i \in \mathrm{PRF}_n, 1 \le i \le m
\end{aligned}
$$

och semantiken beskrivs informellt som:

$$
\begin{aligned}
\mathrm{z}() &= 0 \\
\mathrm{s}(j) &= j + 1 \\
\mathrm{proj}_i^n(j_0, \ldots, j_n) &= j_i \\
\mathrm{comp}(g, f_1, \ldots, f_m)(j_1, \ldots, j_n) &= g(f_1(j_1, \ldots, j_n), \ldots, f_m(j_1, \ldots, j_n))
\end{aligned}
$$

Ge en informell beskrivning av den konstruktion som saknas! Visa också hur man kan uttrycka funktionen som räknar ut predecessorn till ett naturligt tal i PRF ! (Predecessorn är den funktion som för argumentet 0 ger svaret 0 och för argumentet $n+1$ returnerar n). För den sista uppgiften är det viktigt att du motiverar svaret, dvs antingen visa att programmet verkligen uppfyller de ekvationer som skall gälla för predecessor-funktionen eller också visa att ditt sätt att komma fram till programmet är sådant att programmet är korrekt. Det räcker alltså inte att bara ge programmet.

7. Skriv ett program isnat_1 i språket χ med följande egenskap: (42)

$$\text{isnat}_1\ n = \mathsf{True}\langle\rangle \quad \text{om } n \text{ är ett naturligt tal}$$

Man skulle ju vilja kunna skriva ett program isnat_2 med egenskapen:

$$\text{isnat}_2\ n = \begin{cases} \mathsf{True}\langle\rangle & \text{om } n \text{ är ett naturligt tal,} \\ \mathsf{False}\langle\rangle & \text{om } n \text{ evalueras till en konstruerar-applikation som inte är ett tal} \end{cases}$$

men det verkar omöjligt. 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!

Språkets operationella semantik ges av följande definition:

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

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

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

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

$$\text{string}(i) \longrightarrow \text{string}(i)$$

$$\text{lambda}(i, e) \longrightarrow \text{lambda}(i, e)$$

$$\text{constr}(i, e) \longrightarrow \text{constr}(i, e)$$

$$\text{struct}(t) \longrightarrow \text{struct}(t)$$

Lycka till!