Göteborgs Universitet och Chalmers Tekniska Högskola 1999–02–22
Datavetenskap INN11

Tentamen i Beräkningsbarhet och lambda-kalkyl

Fredagen den 26 februari 1999, kl 8.45 – 12.45 i sal ML16.
Ansvarig lärare: Bengt Nordström, tel 1033, eller 13 78 14 (hem)
Tillåtna hjälpmedel: Inga.
Börja varje uppgift på nytt blad. Skriv endast på en sida av papperet. Den här skriftliga tentamen utgör 75 % av den totala examinationen, de resterande 25 % består av de inlämingsuppgifter som har delats ut under kursens gång. Klarar ni 50 % av hela examinationen kommer ni att få godkänt. Lösningar och tid för examensvisning kommer att anslås på kursens hemsida.

1. (a) Om A och B är två mängder, vad är en partiell funktion från A till B? (2)

   Svar: En partiell funktion f är en binär relation över A och B så att $f(a, b)$ och $f(a, c)$ implicerar $b = c$. Ett alternativt svar är att f är en delmängd av den kartesiska produkten $A \times B$ så att $(a, b) \in f$ och $(a, c) \in f$ implicerar att $b = c$

   (b) Vad menas med att en funktion är χ- beräkningsbar? (4)

   Svar: Se föreläsningsanteckningarna i kapitlet Computability

   (c) Ange vilka variabler som är fria respektive bundna i $(\lambda v.xy)(\lambda x.v(\lambda z.z))$

   Svar: Variablerna x, y, v är fria. Variablerna v, x, z är bundna. (2)

   (d) Vad menas med β-normal-form? (3)

   Svar: Att ett lambda-uttryck är på β-normal-form betyder att det inte har något β -redex.

2. Bevisa att det är omöjligt att ha ett programmeringsspråk som precis innehåller de beräkningsbara funktioner som tar ett naturligt tal som argument och ger ett naturligt tal som resultat. (8)

   Svar: Svaret finns på hemsidan

3. Bevisa att $\mathsf{List}(\mathsf{N})$, mängden av ändliga listor över N, är en uppräknelig mängd. (7)

   Svar: läroboken sid 32

4. Beskriv språket PRF, mängden av de primitivt rekursiva funktionerna. Ge den abstrakta syntaxen, ge en informell beskrivning av semantiken för de olika konstruktionerna i språket. Ge ett exempel (bevis ej nödvändigt) på ett program som ej kan uttryckas i språket. (12)

   Svar: sid 23 i boken

5. Ge en induktiv definitition av abstrakta syntaxen för programmen i λ-kalkyl. Blir detta en uppräknelig mängd? Motivera svaret. (14)

   Svar: Vi har tre sätt att bilda uttryck i lambda-kalkyl. Applikation, abstraktion och variabler. Den konkreta syntaxen kan ju beskrivas som $\mathsf{E} := (\mathsf{EE})|\mathsf{x}|\lambda\mathsf{x}.\mathsf{E}$. Vi får då följande induktiva definition av den abstrakta syntaxen, vi definierar mängden Λ, som beror på en mängd Id som representerar variabler (Jämför definitionen av χ s abstrakta syntax!):

$$\begin{aligned} \text{apply}(e_1,\ e_2) \in \Lambda &\quad \text{if} \quad e_1, e_2 \in \Lambda \\ \text{lambda}(i, e) \in \Lambda &\quad \text{if} \quad i \in \text{Id}, e \in \Lambda \\ \text{var}(i) \in \Lambda &\quad \text{if} \quad i \in \text{Id} \end{aligned}$$

   Är den här mängden uppräknelig? Om mängden av identifierare är uppräknelig så blir den det (mängden Λ är ju bara en blandning av binära träd (applikationer) och listor (abstraktionerna). Anta att g' är en injektion från Id till N. Vi vill konstruera g, en injektion från Λ till N:

$$\begin{aligned} g(\text{apply}(e_1,\ e_2)) &= 2^2 \cdot 3^{g(e_1)} \cdot 5^{g(e_2)} \\ g(\text{lambda}(i, e)) &= 2^3 \cdot 3^{g'(i)} \cdot 5^{g(e)} \\ g(\text{var}(i)) &= 2^4 \cdot 3^{g'(i)} \end{aligned}$$

   Detta blir en injektion, om $g(a) = g(b)$ så måste $g(a)$ och $g(b)$ ha samma 2-exponent. Om denna är 2 vet vi tre saker. Dels att a och b har samma yttre form (nämligen apply) samt att 3 - och 5-exponenterna är identiska. Men 3 - och 5-exponenterna är koden av argumenten till apply. Om 2-exponenten är 3 vet vi att den yttre formen är lambda, och att 3 - och 5-exponenten ger delarna. Slutligen, om 2-exponenten är 4 vet vi att att den yttre formen är var, etc.

   Om jag har tid kommer jag att ge ett mer fullständigt bevis...

6. Visa att λ-termen Z definierad av

$$\mathsf{Z} = \lambda \mathsf{f}.((\lambda \mathsf{x}.\mathsf{f}(\lambda \mathsf{z}.\mathsf{xxz}))(\lambda \mathsf{x}.\mathsf{f}(\lambda \mathsf{z}.\mathsf{xxz})))$$

   är en fixpunktskombinator. (10)

   Svar: sid 72 i övningshäftet.

7. Förklara varför existensen av en själv-evaluator för språket χ kan ses som ett bevis av att den operationella semantiken för språket är χ-beräkningsbart. (3)

   Svar: Att det finns en själv-evaluator för språket betyder att det finns ett program $eval$ som när det appliceras på koden av ett program beräknas till koden av programmets värde. Det vill säga $\mathit{eval}\ \overline{\mathsf{f}}$ beräknar till $\overline{\mathsf{v}}$ omm f beräknar till v. Att den operationella semantiken (som kan ses som en partiell funktion från ett program till dess värde) är beräkningsbar betyder att det finns en funktion som har precis samma egenskap som eval ovan.

8. Ge en ändlig automat som känner igen de strängar över alfabetet $\{0, 1\}$ som kan skrivas som $\{11^n01^m\}$. (10)

   Svar: sid 196 i läroboken

Lycka till!