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

Tentamen i Beräkningsmodeller

Måndagen den 24 Januari 2001, kl 8.45 – 13.45 i TL1
Ansvarig lärare: Bengt Nordström, tel 1033
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 måndagen den 5 Februari kl 15.30 i Bengt Nordströms tjänsterum. Datum för omtentamen kommer att annonseras på kursens hemsida. Lösningar till den här tentan kommer också att finnas tillgänglig från kursens hemsida.

1. Visa hur man kan representera elementen i mängden **Bool** i lambda-kalkyl respektive χ! (15)

   Svar:

2. Ge två olika definitioner av vad det betyder att en mängd är uppräkningsbar! (20)

   Svar:

3. Kan man räkna upp $\wp(\mathbf{Bool}) \to \wp(\mathbf{Bool})$, mängden av alla (totala) funktioner från $\wp(\mathbf{Bool})$ till sig själv. Mängden $\wp(\mathbf{Bool})$ är mängden av alla delmängder av **Bool**. Motivera! (15)

   Svar: Mängden $\wp(\mathbf{Bool}) \to \wp(\mathbf{Bool})$ är ändlig eftersom den tar en ändlig mängd till en ändlig mängd, varje element i definitionsmängden kan ju avbildas på högst ändligt antal element. Alla ändliga mängder är uppräkneliga, om mängden har n element kan vi ju avbilda talet i på det i'te elementet i mängden för alla $i \leq n$ och avbilda alla andra tal på det n'te elementet.

4. Bevisa att man i Haskell (eller något annat funktionellt språk) inte kan skriva en funktion `halt :: (Nat -> Nat) -> Nat -> Bool` som är sådan att `(halt f i)` evaluerar till `true` om `(f i)` terminerar och annars evaluerar till `false` . (20)

   Svar: Detta är bara en enkel variant av det extensionella stopp-problemet. Om vi har en funktion `halt` enl ovan skulle vi kunna definiera en funktion `termcnv` genom

   ```
   termcnv f i = if (halt f i) then loop else 1
   ```

som har egenskapen att `termcnv f i` terminerar om och endast om `f i` ej terminerar. Detta gäller för alla funktioner `f`. Speciellt för den funktion `strange` som är definierad av

```
strange i = termcnv strange i
```

Definitionen av `strange` visar att `termcnv strange i` terminerar samtidigt som `strange i`, vilket motsäger egenskapen ovan.

5. Beskriv en Turing-maskin! (30)

   Svar: Se läroboken sid

6. Antag att vi skulle använda programspråket Java (ni kan byta ut Java mot Ada, Pascal, Basic, Fortran eller Cobol) som beräkningsmodell!

   - Vad betyder det att en partiell funktion $f \in \mathbf{N} \tilde{\rightarrow} \mathbf{N}$ är Java-beräkningsbar? (15)

     Svar: Vi måste först ha ett sätt att representera naturliga tal i Java, låt $\ulcorner n \urcorner$ vara det värde i Java som representerar det naturliga talet n. En funktion $f \in \mathbf{N} \tilde{\rightarrow} \mathbf{N}$ är Java-beräkningsbar om och endast om det finns en funktion (procedur, metod eller något annat som representerar funktioner) p i Java så att när man applicerar (anropar) p på $\ulcorner n \urcorner$ får värdet $\ulcorner f(n) \urcorner$ om och endast om $f(n)$ är definierad.

   - Beskriv vad som behöver göras för att bevisa att den operationella semantiken till Java är Java-beräkningsbar! (15)

     Svar: Först måste man definiera den abstrakta syntaxen för Java, sedan den operationella semantiken, dvs en relation $p \longrightarrow q$, vilken är sann om Java-programmet p beräknar till värdet q. Värdet kan vara tämligen komplicerat då det måste innehålla en beskriving av programmet p's interaktion med omvärlden. Sen måste vi ge ett sätt att representera ett godtycklig Java-program p som ett Java-värde $\ulcorner p \urcorner$. Sen måste vi ge en självinterpretator i Java, dvs ett program **eval** sådant att

     $$(\mathbf{eval}\ \ulcorner p \urcorner) \longrightarrow \ulcorner v \urcorner \quad \text{om och endast om} \quad p \longrightarrow v$$

   Svar:

7. Visa att λ-termen Z definierad av

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

   är en fixpunktskombinator. (20)

   Svar: sid 72 i övningshäftet.

Lycka till!