

Tentamen i Beräkningsmodeller (TDA181-182/INN110)

1 Juni 2007, kl 8.30 – 12.30 i 14.00 – 17.00 i M-huset.

Ansvarig lärare: Bengt Nordström, tel 0730-79 42 89.

Tillåtna hjälpmedel: Inga.

Börja varje uppgift på nytt blad. Skriv endast på en sida av papperet. Varje svar skall motiveras! Svar utan motivering ger inga poäng. Komplicerade lösningar och motiveringar kan ge poängavdrag.

Poäng från hemuppgifter inlämnade under 2007 kan tillgodoräknas. Betygsgränser för 5 p: CTH: 3=80p, 4=120p, 5=160p, GU: G=80p, VG=140p. Betygsgränserna för motsvarande 4 p kurs är 80 % av dessa.

Examensvisning kommer att äga rum fredagen den 8 juni 13.30 – 13.45 i Bengt Nordströms tjänsterum. Lösningar till tentan kommer att finnas tillgängliga från kursen Beräkningsmodellens hemsida.

1. (a) Definiera (i Haskell) funktionen `iter` med egenskapen (5)

```
iter :: Int -> (a -> a) -> a -> a
iter n f a = f (f ... (f a)...)

```

där det finns n st förekomster av f i höger led!

- (b) Definiera (fortfarande i Haskell) funktionen `iter` utan hjälp av rekursion. Detta skall göras genom att använda funktionen `rec` som motsvarar operatorn för primitiv rekursion. Funktionen `rec` har följande definition: (15)

```
rec 0    d e = d
rec (n+1) d e = e n (rec n d e)
rec _    d e = error "negative argument to rec"

```

- (c) I beräkningsmodellen PRF finns en konstruktion som liknar `rec`. Beskriv denna genom att ge abstrakt syntax och operationell semantik. (10)
- (d) På vilket sätt är `iter` använt i λ -kalkyl? (10)
- (e) Implementera funktionen `rec` i χ . Visa först hur de naturliga talen representeras i χ . (10)

2. Låt f vara en beräkningsbar funktion från de naturliga talen till de naturliga talen. Bevisa att det finns oändligt många syntaktiskt olika λ -termer som beräknar $f!$ (Två λ -termer är syntaktiskt lika om de är alfa-konvertibla.). (10)
3. Trots resultatet i föregående uppgift finns det strikt fler matematiska funktioner i $\mathbf{N} \rightarrow \mathbf{N}$ än det finns beräkningsbara funktioner i $\mathbf{N} \rightarrow \mathbf{N}$. Bevisa detta genom att lösa följande uppgifter:
- (a) Definiera vad det betyder att en funktion i $\mathbf{N} \rightarrow \mathbf{N}$ är beräkningsbar i λ -kalkyl. (5)
 - (b) Bevisa att mängden av program är uppräknelig (oberoende av programmeringsspråk). (10)
 - (c) Bevisa att mängden $\mathbf{N} \rightarrow \mathbf{N}$ ej är uppräknelig. (15)
4. I beräkningsmodellen PRF används inte variabler, i stället används projektion och en generalisering av funktionskomposition.
- (a) Ge en informell beskrivning av dessa två konstruktioner! (10)
 - (b) Ge deras abstrakta syntax! (10)
 - (c) Ge deras operationella semantik! (10)
5. Förklara hur en Turing-maskin fungerar. Förklara även begreppet positionerad remsa. (10)
6. Den operationella semantiken för språket χ är en relation $p \rightarrow q$ som betyder att programmet p beräknar till q . Denna relation är en funktion.
- (a) Vad betyder det att en relation är en funktion? (5)
 - (b) Är relationen \rightarrow sedd som en funktion χ -beräkningsbar? Du behöver inte ge ett detaljerat bevis, bara en skiss. Du måste kunna förklara vad det betyder att funktionen är beräkningsbar och kunna förklara vad som behöver bevisas för att så är fallet. (15)

Lycka till!