INSTITUT FOR MATEMATIK OG DATALOGI

ODENSE UNIVERSITET

Skriftlig eksamen

D M 1 5

Tirsdag den 1. juni 1993 kl. 9.00 - 13.00

Opgavesættet består af 5 nummererede sider og indeholder 4 opgaver.
Opgaverne vægtes ens.
Alle skriftlige hjælpemidler samt brug af lommeregner er tilladt.

Opgave 1

Denne opgave omhandler sprog over alfabetet $\{a\}$.

spørgsmål a

Konstruer en nondeterministisk endelig automat (NFA), der accepterer sproget givet ved det regulære udtryk $\lambda \cup a \cup (aa(aaa)^*) \cup (aaa(aaa)^*)$.

spørgsmål b

Antag, at L er et sprog over alfabetet $\{a\}$ og antag at $u_1, u_2, u_3, \ldots$ er alle (muligvis uendeligt mange) strenge i L skrevet op i *voksende* længder. Mere formelt kan man formulere dette:

$$\forall i \geq 1\colon\ \mathit{length}(u_i) < \mathit{length}(u_{i+1})$$

Da kaldes L for *ultimativt periodisk*, hvis og kun hvis der eksisterer positive heltal n og p, sådan at

$$\forall i \geq n\colon\ \mathit{length}(u_{i+1}) - \mathit{length}(u_i) = \mathit{length}(u_{i+p+1}) - \mathit{length}(u_{i+p})$$

Det vil altså sige, at fra et eller andet tidspunkt, n, begynder strengenes længde at vokse med et mønster, der gentager sig med perioden p.

F.eks. er

$$E = \{\lambda, a, aa, aaa, aaaaa, aaaaaa, aaaaaaaa, aaaaaaaaa, \ldots\}$$

svarende til længderne $0, 1, 2, 3, 5, 6, 8, 9, 11, 12, 14, 15, \ldots$ ultimativt periodisk. Man kan f.eks. anvende $n = 3$ og $p = 2$ for at indse dette. Så ser man nemlig, at $3 - 2 = 6 - 5$ og $5 - 3 = 8 - 6$ osv.

Vis, at ethvert ultimativt periodisk sprog over $\{a\}$ er regulært.

Hjælp: Bemærk, at E kan skrives:

$$\{\lambda, a\} \cup (\{aa, aaaaa, aaaaaaaa, \ldots\} \cup \{aaa, aaaaaa, aaaaaaaaa, \ldots\})$$

spørgsmål c

Følgende kontekstfri grammatik G genererer også sproget E. Grammatikken er defineret ved $G = (\{S, A, B, C, P\}, \{a\}, P, S)$, hvor P indeholder produktionerne:

$S \to A$	$A \to a$	$B \to C$	$C \to aaP$	$P \to aaaP$
$S \to ABC$	$A \to \lambda$	$C \to \lambda$	$C \to aaaP$	$P \to \lambda$

Konstruer en grammatik G' *ved anvendelse af teknikken i Sudkamp Theorem 5.1.5 side 103*, sådan at følgende er opfyldt:

- startsymbolet i G' er ikke-rekursivt (nonrecursive).
- G' er ikke-sammentrækkende; bortset fra startsymbolet (essentially noncontracting).
- $L(G') = L(G)$.

Man må *ikke* anvende ad hoc metoder, men *skal* bruge metoden fra Sudkamp.

Opgave 2

Betragt sproget $L_1 = \{a^{2i}b^ic^{2i} \mid i \geq 0\}$. Dette sprog indeholder bl.a. strengene λ, $aabcc$ og $aaaabbcccc$.

spørgsmål a

Vis, at L_1 ikke er kontekstfrit.

spørgsmål b

Lad nu L_2 være ligesom L_1 bortset fra, at alle a'er erstattes af 01, alle b'er erstattes af 10, og alle c'er erstattes også af 10.

Vi vil bruge notationen $L_2 = L_1[a \leftarrow 01, b \leftarrow 10, c \leftarrow 10]$ for denne transformation.

De tre eksempelstrenge ovenfor vil altså give anledning til, at strengene λ, 0101101010 og 01010101101010101010 ligger i L_2.

Vis, at L_2 er kontekstfrit.

spørgsmål c

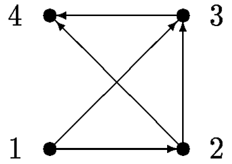
Antag, at vi har en Turingmaskine $M = (Q, \{a, b\}, \{a, b, B\}, \delta, q_0)$. Gør rede for, hvordan man skal definere en ny Turingmaskine $M' = (Q', \{0, 1\}, \{0, 1, B\}, \delta', q_0')$, sådan at $L(M') = L(M)[a \leftarrow 01, b \leftarrow 10]$. Hvis det er nogen hjælp, må man gerne selv bestemme, hvor på båndet inputstrengen placeres.

spørgsmål d

Vis, at der ikke findes nogen algoritme, der givet en Turingmaskine som input afgør, om Turingmaskinen skriver strengen 111 (det vil altså sige, om der på noget tidspunkt findes tre ettaller i træk på båndet).

Opgave 3

I denne opgave ser vi på en bestemt repræsentation af grafer i MIRANDA. Som et eksempel repræsenteres følgende orienterede graf



af MIRANDA-udtrykket `[(1,[2,3]),(2,[3,4]),(3,[4]),(4,[])]`.
Det vil altså sige, at for hver knude x er der præcis ét tupel (x, l), hvor l er en liste af de knuder y, hvor der er en kant i grafen fra x til y. I denne opgave vil vi kalde et sådan MIRANDA-udtryk, der har typen $[(num, [num])]$, for en *graf*. Listen l kaldes for x's *kantliste*. F.eks. angiver $(1, [2, 3])$, at der fra 1 er en kant til 2 og en kant til 3.

I alle delspørgsmålene skal man angive typen af de funktioner, man skriver.

spørgsmål a

Skriv en funktion *combinegraphs*, der tager to grafer g og g' som argumenter og leverer en ny graf g'' som resultat. Det skal være sådan, at hvis der f.eks. er en kant fra x til z i den første graf og en kant fra z til y i den anden, så skal der være en kant fra x til y i resultatet. Mere formelt skal der altså gælde for alle x og y, at y findes i x's kantliste i g'', hvis og kun hvis der findes et z, sådan at z findes i x's kantliste i g, og y findes i z's kantliste i g'. Man må gerne antage, at de to grafer har samme mængde knuder.

For eksempel er `combinegraphs g1 g2 = [(1,[3,4]),(2,[4]),(3,[]),(4,[])]`, hvis $g1$ og $g2$ har værdierne

```
g1  =  [(1,[2,3]),(2,[3,4]),(3,[4]),(4,[])]
g2  =  [(1,[2]),(2,[3]),(3,[4]),(4,[])]
```

Læg specielt mærke til, at 4 findes i 1's kantliste i resultatet, fordi 3 findes i 1's kantliste i førsteargumentet, og 4 findes i 3's kantliste i andetargumentet.

spørgsmål b

Skriv en funktion *transitiv*, der tager en graf g som argument og leverer en ny graf g' som resultat. I g' skal y findes i x's kantliste, hvis og kun hvis der i g findes en sti fra x til y. Det vil altså sige, at man kan komme fra x til y ved at følge kanterne i pilenes retning. F.eks. skal `transitiv [(1,[2]),(2,[3]),(3,[4]),(4,[])]` give resultatet `[(1,[2,3,4]),(2,[3,4]),(3,[4]),(4,[])]` .

spørgsmål c

En graf indeholder en *kreds*, hvis der findes en sti fra et eller andet x tilbage til x igen. F.eks. indeholder den meget simple graf $[(1, [2]), (2, [1])]$ en kreds, fordi man kan komme fra 1 til 2 og fra 2 tilbage til 1.

Skriv en funktion *testcycle*, der tager en graf som argument og returnerer enten *true* eller *false* afhængig af, om der er en kreds eller ej.

Opgave 4

I denne opgave skal vi se på indkodninger af heltalsfunktioner i lambda-kalkylen.

Husk, at vi i Henson har defineret udtrykkene **T**, **F**, **pair**, **zero** og **succ** på Church's indkodning af heltallene, hvor tallet n er givet ved $c_n = \lambda fx \cdot f^n(x)$. Husk også, at udtrykkene **T** og **F** står for *true* og *false*. Desuden har vi set, at man kan definere en predecessor-funktion **pred**, sådan at $\mathbf{pred}\, c_{n+1} = c_n$, for alle $n \geq 0$.

Vi har også set, at man vha. fikspunktsfinderen **Y** kan definere funktioner, som følgende eksempel illustrerer. Udtrykket

$$\mathbf{even} = \mathbf{Y}\, \lambda f \cdot \lambda x \cdot (\mathbf{zero}\, x)\, \mathbf{T}\, ((\mathbf{zero}\, (\mathbf{pred}\, x))\, \mathbf{F}\, (f\, (\mathbf{pred}\, (\mathbf{pred}\, x))))$$

afgør for eksempel, om et tal er lige eller ulige. Dvs.

$$\mathbf{even}\, c_n = \begin{cases} \mathbf{T}, & \text{hvis } n \text{ er lige} \\ \mathbf{F}, & \text{ellers} \end{cases}$$

spørgsmål a

Lad lambda-udtrykket **hint** være defineret ved:

$$\mathbf{hint} = \lambda xy \cdot (\mathbf{zero}\, x)\, c_0\, ((\mathbf{zero}\, y)\, x\, (\mathbf{pair}\, (\mathbf{pred}\, x)\, (\mathbf{pred}\, y)))$$

Beregn (reducér) de tre udtryk $\mathbf{hint}\, c_0\, c_2$, $\mathbf{hint}\, c_2\, c_0$ og $\mathbf{hint}\, c_1\, c_2$.

spørgsmål b

Normal subtraktion kan ikke defineres på de positive heltal, da f.eks. $5 - 7$ jo er et negativt tal. Derfor bruger man ofte *monadisk* subtraktion defineret ved:

$$x \ominus y = \begin{cases} x - y, & \text{hvis } x \geq y \\ 0, & \text{ellers} \end{cases}$$

F.eks. er $9 \ominus 4 = 5$, mens $3 \ominus 7 = 0$.

Definer et lambda-udtryk **ms**, så $\mathbf{ms}\, c_n\, c_m = c_{n \ominus m}$.

spørgsmål c

Ved almindelig division, x/y, kan man risikere, at resultatet ikke er et heltal. F.eks. er $7/4 = 1,75$. På positive heltal kan man definere *heltalsdivision*, *div*, ved at vedtage, at $x\ \mathit{div}\ y$ er det største *hele* tal mindre end eller lig med x/y. F.eks. er $7\ \mathit{div}\ 4 = 1$, mens $8\ \mathit{div}\ 4 = 2$ og $9\ \mathit{div}\ 4 = 2$.

Definer et lambda-udtryk **div**, så $\mathbf{div}\, c_n\, c_m = c_{n\ \mathit{div}\ m}$.