INSTITUT FOR MATEMATIK OG DATALOGI

ODENSE UNIVERSITET

Skriftlig eksamen

D M 1 5

Fredag den 21. januar 1994 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 handler om regulære sprog. Husk, at selv om man ikke har fået regnet alle spørgsmålene, må man gerne benytte resultaterne fra tidligere spørgsmål til at løse de senere.

spørgsmål a

Tegn tilstandsgrafen for en nondeterministisk endelig automat med lambda-transitioner (NFA-λ), der accepterer sproget

$$\{abbb\} \cup \{a^*b^n \mid n \text{ er lige, } n \geq 0\}.$$

spørgsmål b

Vis, at hvis $L_1 \cup L_2$ er et regulært sprog, og L_1 indeholder præcis én streng, så er L_2 også et regulært sprog.

spørgsmål c

Vis, at hvis $L_1 \cup L_2$ er et regulært sprog, og L_1 er endeligt, så er L_2 også et regulært sprog.

spørgsmål d

Vis, at hvis $L_1 \cup L_2$ er et regulært sprog, og L_1 også er et regulært sprog, så er L_2 ikke nødvendigvis et regulært sprog.

Opgave 2

Denne opgave handler om kontekstfri sprog og afgørlighed.
Lad $\Sigma = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}$.
Vi benytter notationen w^R til at betyde strengen w *reverseret*. Dvs. hvis $w = d_1 d_2 \cdot d_k$, så er $w^R = d_k \cdot d_2 d_1$.

spørgsmål a

Lad $L_1 = \{w_1 c w_2 c w_2^R c w_1^R \mid w_1 \in \Sigma^+,\ w_2 \in \{1, 2, \ldots, n\}^+,\ c \notin \Sigma\}$.
Vis, at L_1 er kontekstfrit.

spørgsmål b

Lad $L_2 = \{w_1 c w_2 c w_1^R c w_2^R \mid w_1 \in \Sigma^+,\ w_2 \in \{1, 2, \ldots, n\}^+,\ c \notin \Sigma\}$.
Vis, at L_2 ikke er kontekstfrit.

spørgsmål c

Antag, at $\mathbf{x} = (x_1, \ldots, x_n)$ og $\mathbf{y} = (y_1, \ldots, y_n)$, hvor $x_i, y_i \in \Sigma^+$, $1 \leq i \leq n$.
Lad $G_{\mathbf{x},\mathbf{y}} = (V_{\mathbf{x},\mathbf{y}}, \Sigma_{\mathbf{x},\mathbf{y}}, P_{\mathbf{x},\mathbf{y}}, S_{\mathbf{x},\mathbf{y}})$, hvor

- $V_{\mathbf{x},\mathbf{y}} = \{S_{\mathbf{x},\mathbf{y}}, A, B\}$
- $\Sigma_{\mathbf{x},\mathbf{y}} = \Sigma \cup \{c\} \cup \{1, 2, \ldots, n\}$, hvor $1, 2, \ldots, n, c \notin \Sigma$
- $$P_{\mathbf{x},\mathbf{y}} = \begin{array}{ll} & \{S_{\mathbf{x},\mathbf{y}} \rightarrow AcB,\ A \rightarrow c,\ B \rightarrow c\} \\ \cup & \{A \rightarrow x_i A i,\ B \rightarrow i B y_i^R \mid i = 1, 2, \ldots, n\} \end{array}$$

Vis, at det generelt (dvs. for vilkårlige $\mathbf{x}$ og $\mathbf{y}$) er uafgørligt om $L(G_{\mathbf{x},\mathbf{y}}) \cap L_1 = \emptyset$.

spørgsmål d

Antag, at $G = (V, \Sigma, P, S)$ er en kontekstfri grammatik. Vis, at det er afgørligt om en streng $w \in L(G)$. Man skal altså beskrive en algoritme, der tager en grammatik G og en streng w som input og som så svarer på om $w \in L(G)$.

Opgave 3

I denne opgave skal man implementere et regelsystem i MIRANDA.

Et *minisystem* består af et alfabet Σ og en mængde regler P på formen $x \to y$, hvor $x \in \Sigma$ og $y \in \Sigma^*$. Som sædvanlig betyder $x \stackrel{*}{\Rightarrow} y$, at y kan deriveres fra x gennem et endeligt antal anvendelser af reglerne i systemet.

For eksempel, hvis $\Sigma = \{a, b\}$ og $P = \{a \to aba,\ b \to bb,\ a \to b,\ a \to \lambda\}$, så har vi $abaa \Rightarrow abaaba \Rightarrow abaabba \Rightarrow abaabb \Rightarrow bbaabb$, så $abaa \stackrel{*}{\Rightarrow} bbaabb$.

Vi benytter følgende typedefinitioner i MIRANDA:

- alphabet == char
- rule == (char,[char])
- mini_system == [rule]

I alle delspørgsmålene skal man angive typen af de funktioner, man skriver. Husk, at selv om man ikke har fået regnet alle spørgsmålene, må man gerne benytte funktionerne fra tidligere spørgsmål til at løse de senere. I de spørgsmål, hvor svaret er en liste af strenge, må denne liste gerne indeholde dubletter.

spørgsmål a

Opskriv P fra eksemplet ovenfor i MIRANDA i overensstemmelse med typedefinitionerne.

spørgsmål b

Skriv en funktion *step*, sådan at resultatet af *step s r* er en liste af alle de strenge, der kan nås i ét skridt fra strengen s gennem anvendelse af reglen r. For eksempel, hvis s er *bab*, og r er $b \to bb$, så skal resultatet være en liste bestående af de to strenge *bbab* og *babb*.

spørgsmål c

Skriv en funktion *step_all*, sådan at resultatet af *step_all s m* er en liste af alle de strenge, der kan nås i ét skridt fra strengen s gennem anvendelse af en eller anden regel i minisystemet m.

spørgsmål d

Skriv en funktion *generate*, sådan at *generate s m* producerer en liste af alle mulige strenge, der kan nås i ét eller flere skridt fra strengen s gennem anvendelse af regler i minisystemet m. Bemærk, at denne liste muligvis er uendelig lang. Det kræves, at *enhver* streng, der kan nås i ét eller flere skridt, vil blive genereret på et eller andet tidspunkt.

Opgave 4

Denne opgave handler om lambda-kalkylen. Man må gerne benytte sig af udtryk, man allerede kender fra lærebogen. Dvs. **pair**, **h**, **t**, **T**, **F**, **if-then-else**, **Y**, samt Church indkodningen af heltal med udtrykkene **succ**, **zero** og **pred**. Udtrykket **pred** er også omtalt på ugeseddel 27, men alt man behøver at vide er, at $\mathbf{pred}\, c_i = c_{i-1}$ for $i \geq 1$.

Bemærk, at det sidste spørgsmål er uafhængigt af de forgående.

spørgsmål a

Vi vil nu gerne repræsentere *lister* i lambda-kalkylen. Vi bruger notationen $[M_1, \ldots, M_n]$ for indkodningen af listen med elementerne M_1 til M_n. Indkodningen defineres nu rekursivt som følger:

$$[M] = M \qquad\qquad [M_1, M_2, \ldots, M_n] = \mathbf{pair}\, M_1\, [M_2, \ldots, M_n]$$

For eksempel indkodes listen bestående af de tre elementer a, b og c som $\mathbf{pair}\, a\, \mathbf{pair}\, b\, c$.

Angiv et lambda-udtryk, der tager en liste som argument, og som giver det femte element i listen. Man må gerne antage, at listerne altid er af længde mindst fem.

spørgsmål b

Angiv et lambda-udtryk P, der tager en liste l og et Church-indkodet tal c_k som argument. Der skal gælde, at $P\, l\, c_k$ giver det k'te element i listen l, hvis $k \in \{1, 2, \ldots, |l|\}$, hvor $|l|$ er længden af listen l.

spørgsmål c

Antag, at der findes et lambda-udtryk **single**, for hvilket der gælder, at $\mathbf{single}\, l$ reducerer til **T**, hvis l er en liste med kun ét element, og $\mathbf{single}\, l$ reducerer til **F** ellers.

Angiv et lambda-udtryk **concat**, der kan konkatenere to ikke-tomme lister. Dvs. at $\mathbf{concat}\, l_1\, l_2$ skal reducere til en liste af alle elementerne i l_1 fulgt af alle elementerne i l_2.

spørgsmål d

Angiv et lambda-udtryk A uden frie variabler, for hvilket der gælder, at $A\, x = x\, A$ for alle x.

Husk, at $FV(A) = \emptyset$ er definitionen på, at A er uden frie variabler.