INSTITUT FOR MATEMATIK OG DATALOGI

ODENSE UNIVERSITET

Skriftlig eksamen

D M 1 5

Fredag den 20. januar 1995 kl. 9.00 - 13.00

Opgavesættet består af 6 nummererede sider og indeholder 4 opgaver.
Opgaverne vægtes sådan:

- Opgave 1: 25%
- Opgave 2: 25%
- Opgave 3: 30%
- Opgave 4: 20%

Alle skriftlige hjælpemidler samt brug af lommeregner er tilladt.

Opgave 1

Denne opgave handler om regulære sprog.
Lad L være den regulære mængde defineret ved udtrykket $a((bab)^* \cup (baa)^*)$.

spørgsmål a

Tegn tilstandsgrafen for en nondeterministisk endelig automat med λ-overgange (NFA-λ), der accepterer L.

spørgsmål b

Skriv et regulært udtryk for mængden

$$\{x \in \{a,b\}^* \mid \exists y \in L\colon\ \mathit{length}(x) = \tfrac{1}{2}\mathit{length}(y)\}.$$

spørgsmål c

Lad $M = (Q, \Sigma, \delta, q_0, F)$ være en DFA. Definer v.h.a. M en NFA-λ, der accepterer

$$\{x \in \Sigma^* \mid \exists y \in L(M)\colon\ \mathit{length}(x) = \tfrac{1}{2}\mathit{length}(y)\}.$$

spørgsmål d

Vis, at der eksisterer et regulært sprog L over alfabetet $\{a,b\}$, sådan at

$$\{x \in \{a,b\}^* \mid \exists y \in L\colon\ \mathit{length}(x) = (\mathit{length}(y))^4\}$$

ikke er et regulært sprog.

Opgave 2

Denne opgave handler om kontekstfri og kontekstsensitive sprog.

En kontekstfri grammatik $G = (V, \Sigma, P, S)$ er *variabelbegrænset*, hvis og kun hvis der eksisterer et heltal $k \geq 1$, sådan at der for alle derivationer $S \stackrel{*}{\Rightarrow} \alpha$, $\alpha \in (V \cup \Sigma)^*$ gælder, at der højst er k variabler i α. Et sprog L kaldes for *variabelbegrænset*, hvis og kun hvis der eksisterer en variabelbegrænset grammatik for L.

spørgsmål a

Lad $G = (\{S, X, B, C\}, \{a, b, c\}, P, S)$, hvor P består af

$$S \rightarrow aaXBC$$
$$X \rightarrow aaXB \mid \lambda$$
$$B \rightarrow b$$
$$C \rightarrow cC \mid \lambda$$

Er G variabelbegrænset? Hvorfor eller hvorfor ikke?

spørgsmål b

Find en kontekstfri grammatik G', der er variabelbegrænset, selv om $L(G')$ er ubegrænset (d.v.s. at $L(G')$ indeholder uendeligt mange strenge).

spørgsmål c

Find et kontekstfrit sprog L, der ikke er variabelbegrænset. D.v.s. at man skal finde et sprog for hvilket, der ikke eksisterer nogen variabelbegrænset grammatik. Angiv sproget L v.h.a. en kontekstfri grammatik.

spørgsmål d

Find en kontekstsensitiv grammatik $G = (V, \Sigma, P, S)$ for dit sprog L fra spørgsmål c, sådan at

1. hver regel har mindst én variabel på venstresiden, og
2. der for alle derivationer $S \stackrel{*}{\Rightarrow} \alpha$ gælder, at der højst er to variabler i α.

(Hvis du ikke har svaret på spørgsmål c, eller hvis du bare synes, at det er nemmere, så kan du bruge $L = \{wcwcw^R \mid w \in \{a, b\}^*\}$ i stedet for dit sprog fra spørgsmål c.)

Opgave 3

Denne opgave handler om afgørlighed og MIRANDA. Husk, at hvis din løsning ligner noget i lærebogen, så er det ikke nødvendigt at kopiere det hele – bare fortæl hvilke ændringer, der er nødvendige.

spørgsmål a

Lad Σ være et alfabet, og lad $\{(u_i, v_i) \mid 1 \leq i \leq n\}$ være en mængde af ordnede par, hvor $u_i, v_i \in \Sigma^+$. Det *Modificerede Post Correspondence Problem* (MPCP) er følgende:

Eksisterer der $i_1, ..., i_k$, hvor $1 \leq i_j \leq n$, sådan at $u_1 u_{i_1} ... u_{i_k} = v_1 v_{i_1} ... v_{i_k}$?

Den eneste ændring fra det almindelige Post Correspondence Problem er, at de første strenge i MPCP skal være u_1 og v_1.
Vis, at MPCP er uafgørligt.

spørgsmål b

Vis, at Post Correspondence Problem er afgørligt, hvis $\textit{length}(u_i) = \textit{length}(v_i)$ for $1 \leq i \leq n$. Man skal altså beskrive en algoritme, der tager en sådan mængde af ordnede par $\{(u_i, v_i) \mid 1 \leq i \leq n\}$, hvor $u_i, v_i \in \Sigma^+$, som input, og som så svarer på, om der eksisterer $k \geq 1$ og $i_1, ..., i_k$, hvor $1 \leq i_j \leq n$, sådan at

$$u_{i_1} ... u_{i_k} = v_{i_1} ... v_{i_k}.$$

spørgsmål c

Post Correspondence Problem er afgørligt, hvis k er givet som en del af problemet. Nu består systemet af k og en mængde $\{(u_i, v_i) \mid 1 \leq i \leq n\}$ af ordnede par, hvor $u_i, v_i \in \Sigma^+$. Man spørger, om der eksisterer $i_1, ..., i_k$, hvor $1 \leq i_j \leq n$, sådan at

$$u_{i_1} ... u_{i_k} = v_{i_1} ... v_{i_k}.$$

Det er kun nødvendigt at undersøge alle muligheder, der bruger k brikker (dominoes). For eksempel, hvis mængden består af $\{(aaa, aa), (aa, aaaa)\}$, så følger bl.a:

- hvis $k = 1$, så er svaret "nej".
- hvis $k = 2$, så er svaret "nej".
- hvis $k = 3$, så er svaret "ja", fordi $(aaa, aa), (aaa, aa), (aa, aaaa)$, er en løsning.
- hvis $k = 4$, så er svaret "nej".

Skriv en funktion *combination* i MIRANDA, sådan at resultatet af *combination k xs* er en liste, der består af alle mulige lister bestående af k elementer fra xs. Husk, at det er muligt for en liste at indeholde et element mere end én gang. For eksempel er

combination 2 [(“aaa”, “aa”),(“aa”, “aaaa”)]
  = [[(“aaa”, “aa”),(“aaa”, “aa”)],[(“aaa”, “aa”),(“aa”,”aaaa”)],
  [(“aa”, “aaaa”),(“aaa”, “aa”)], [(“aa”, “aaaa”),(“aa”, “aaaa”)]]

Resultatet er alle mulige kombinationer af k brikker.
Typen er

- combination :: num → [([char],[char])] → [[([char],[char])]]

spørgsmål d

Skriv en funktion *check* i MIRANDA, der afgør, om en instans af Post Correspondence Problem har en løsning, der bruger præcis k brikker. Typen skal være

- check :: num → [([char],[char])] → bool

Hjælp: lav evt. først en funktion *make_dominoes*, der udfra enhver af de lister, man får fra funktionen *combination*, laver et par bestående af konkatenationen af henholdsvis alle førstekomponenterne og alle andenkomponenterne. Eksempelvis skal

make_dominoes (combination 2 [(“aaa”, “aa”),(“aa”, “aaaa”)])
  = [(“aaaaaa”, “aaaa”),(“aaaaa”, “aaaaaa”),
  (“aaaaa”, “aaaaaa”), (“aaaa”, “aaaaaaaa”)]

Opgave 4

Denne opgave handler om PCF og MIRANDA. Antag, at man har defineret følgende funktioner i PCF:

- En funktion *pairnums* af typen nat→nat→nat, sådan at *pairnums* er en injektiv afbildning fra $\mathbb{N} \times \mathbb{N}$ til $\mathbb{N} - \{0\}$. (En mulig definition er $pairnums\ n\ m = (n+m)*(n+m+1)/2+m+1$.)
- En funktion $projnum_1$ af typen nat→nat, sådan at $projnum_1\ (pairnums\ n\ m) = n$.
- En funktion $projnum_2$ af typen nat→nat, sådan at $projnum_2\ (pairnums\ n\ m) = m$.

Nu kan man implementere en liste, der består af naturlige tal. En tom liste repræsenteres af tallet 0. En liste med hovedet x og hale repræsenteret af ls kan repræsenteres af *pairnums x ls*. Så listen [1,3,5] i MIRANDA kan repræsenteres af

$$pairnums\ 1\ (pairnums\ 3\ (pairnums\ 5\ 0)).$$

spørgsmål a

Man kunne i stedet have forsøgt at repræsentere lister af naturlige tal i PCP v.h.a. $<,>$, $Proj_1$ og $Proj_2$ og lade $Proj_1\ ls$ indeholde længden af listen ls. For eksempel skulle listen [1,3,5] så repræsenteres af $< 3, < 1, < 3, 5 >>>$. Hvorfor er dette *ikke* nogen god idé? (Vink: Ville der være problemer med at definere funktioner, der tager lister som argumenter?)

spørgsmål b

Nu skal vi se på funktionen *map*, der kendes fra MIRANDA:

```
map :: (* → **) → [*] → [**]
map f ls = [f a | a <– ls]
```

Definer et udtryk F i PCF, så *map* kan laves som

$$map \stackrel{def}{=} fix_{(nat \to nat) \to nat \to nat} F.$$

Funktionen *map* skal have den samme effekt på repræsentationer af lister (som beskrevet ovenfor), som *map* har på lister i MIRANDA.

spørgsmål c

Antag nu, at *pairnums*, $projnum_1$ og $projnum_2$ er defineret sådan, at $projnum_1$ 7 reducerer til 3, og $projnum_2$ 7 reducerer til 0. Brug left-most reductions (normal order reduction) på

$$map\ (\lambda x : nat.x + x)\ 7$$

indtil en normalform nås. Medtag alle skridt.