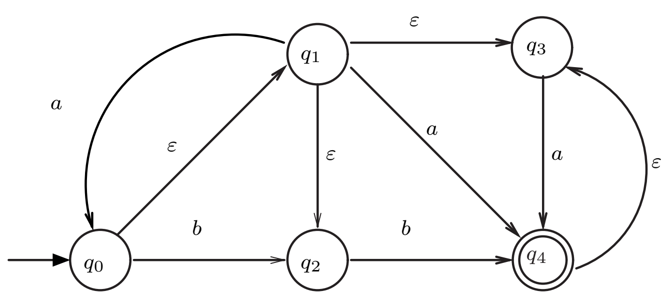
Syntaks og semantik – Skriftlig eksamen – Vejledende løsning

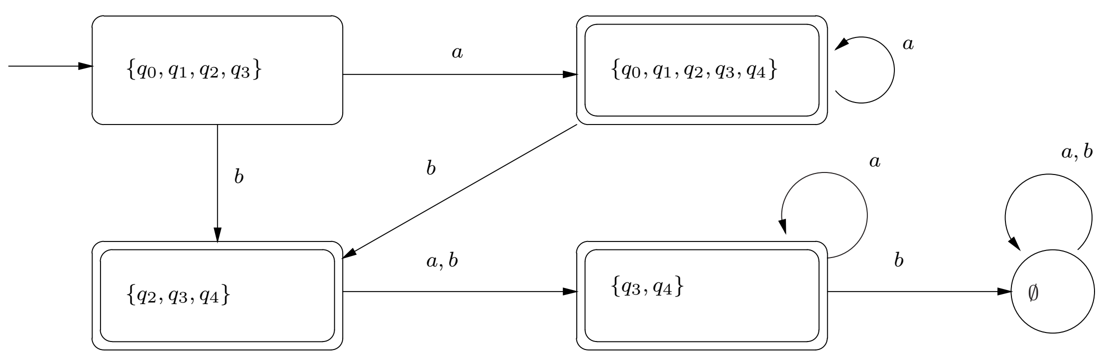
1. juni 2012 kl. 9.00-12.00

1. *(15 point)* Herunder er en nondeterministisk automat.



Anvend den i kurset beskrevne metode til at konvertere denne NFA til en DFA.

Svar



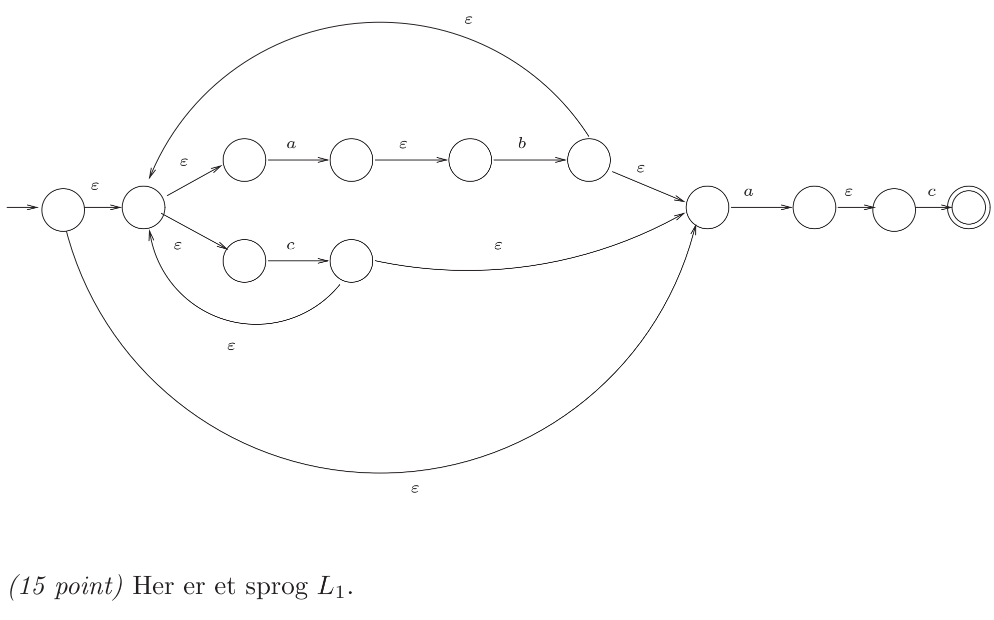
2. *(10 point)* Her er påstande om regulære og kontekstfrie sprog. Sæt kryds ved de svar, du mener er de korrekte.

	Ja	Nej
Hvis $L_1 \subseteq L_2$ og L_2 er et regulært sprog, så er L_1 også et regulært sprog.		X
Sproget $\{0^i 1^i \mid i \geq 0\}$ er kontekstfrit.		X
En kontekstfri grammatik på Chomsky-normalform kan have en produktion på formen $A \to BC$.	X	
Man kan konvertere enhver pushdown-automat til en ækvivalent deterministisk pushdownautomat.		X
Sproget $\{0^n 1^n 2^n \mid n \geq 0\}$ er kontekstfrit.		X

3. *(10 point)* Herunder er et regulært udtryk. Anvend den i kurset beskrevne metode til at konstruere en ækvivalent NFA (der er yderligere plads på næste side).

$$(\mathsf{ab} \cup \mathsf{c})^* \mathsf{ac}$$

Svar



4. *(15 point)* Her er et sprog L_1.

$$L_1 = \{a^k b^{2k} \mid k \geq 0\}$$

a. Bevis, at L_1 ikke er regulært.

Svar

Anvendelse af Pumping Lemma for regulære sprog. Antag at p er pumpelængden for L_1. Vi vælger nu en $s \in L_1$ så $|s| \geq p$ og viser at ingen opsplitning $s = xyz$ kan overholde betingelserne 1–3 samtidig.

Vælg $s = a^p b^{2p}$. Det er klart at $|s| \geq p$. Skal betingelsen $|xy| \leq p$ overholdes, må y kun bestå af a'er. Skal betingelsen $|y| > 0$ også overholdes, må vi have at $y = a^k$ for et $k > 0$. Men så vil betingelsen at $xy^i z \in L_1$ for alle $i \geq 0$ blive overtrådt, thi $xy^2 z \notin L_1$, da $a^{p+k} b^{2p} \notin L_1$.

b. Bevis, at L_1 er kontekstfrit.

Her er en kontekstfri grammatik, der definerer L_1.

$$S \to aSbb \mid \varepsilon$$

5. *(20 point)* Her er et forkert forsøg på at vise at et sprog ikke er kontekstfrit.

> Betragt sproget
>
> $$L_2 = \{w \in \{a, b, c\}^* \mid w \text{ har lige mange } a\text{'er, } b\text{'er og } c\text{'er}\}$$
>
> Vi bruger Pumping Lemma til at vise at sproget ikke er kontekstfrit. Vælg $s = abcabcabc$. Så må vi have $u = \varepsilon, v = abc$, $x = abc$, $y = abc$ og $z = \varepsilon$. Men $uv^2xy^2z \notin L_2$.

a. Forklar de væsentligste grunde til at ovenstående forsøg på et bevis er forkert.

Svar

Man skal vælge en pumpelængde p, betragte en $s \in L_2$ så $|s| \geq p$ og vise at *alle* opsplitninger af s vil overtræde mindst én af de tre betingelser i Pumping Lemma.

b. Giv et korrekt bevis. (Det er en god idé *ikke* at forsøge at reparere det forkerte bevis!) Der er mere plads på næste side.

Svar

Anvendelse af Pumping Lemma for kontekstfrie sprog. Antag at p er pumpelængden for L_2. Vi vælger nu en $s \in L_2$ så $|s| \geq p$ og viser at ingen opsplitning $s = uvxyz$ kan overholde betingelserne 1–3 samtidig.

Vælg $s = a^p b^p c^p$. Det er klart at $|s| \geq p$.

Vi skal nu undersøge alle mulige opsplitninger $s = uvxyz$. Her er tre tilfælde:

i. v og y indeholder tilsammen alle tre slags tegn. Men så vil vxy blive for lang, da der er p b'er. Dvs. $|vxy| > p$, og betingelsen $|vxy| \leq p$ bliver overtrådt.

ii. v og y indeholder tilsammen to slags tegn. Men så vil $uv^2xy^2z \notin L_2$, da denne streng har for få af den tredje slags tegn, og betingelsen, at $uv^ixy^iz \in L_2$ for alle $i \geq 0$, bliver overtrådt.

iii. v of y indeholder tilsammen én bestemt slags tegn. Men igen vil $uv^2xy^2z \notin L_2$, da denne streng har for få af de to andre slags tegn, og betingelsen, at $uv^ixy^iz \in L_2$ for alle $i \geq 0$, bliver også her overtrådt.

6. *(15 point)* Her er opbygningsreglerne for kommandoer i **Bims**.

$$S ::= \quad x{:=}a \mid \texttt{skip} \mid S_1; S_2 \mid \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2 \mid \texttt{while}\ b\ \texttt{do}\ S$$

Betragt *big-step-semantikken* for **Bims**. Udfyld for hver af disse sproglige konstruktioner, de transitionsregler, der er nødvendige.

Assignment (tildeling)	
Sekventiel sammensætning	
While-løkker	

Svar

(ASS)	$\langle x := a, s\rangle \to s[x \mapsto v]$ hvor $s \vdash a \to_a v$
(COMP)	$\dfrac{\langle S_1, s\rangle \to s'' \quad \langle S_2, s''\rangle \to s'}{\langle S_1; S_2, s\rangle \to s'}$
(WHILE-SAND)	$\dfrac{\langle S, s\rangle \to s'' \quad \langle \texttt{while}\ b\ \texttt{do}\ S, s''\rangle \to s'}{\langle \texttt{while}\ b\ \texttt{do}\ S, s\rangle \to s'}$ hvis $s \vdash b \to_b tt$
(WHILE-FALSK)	$\langle \texttt{while}\ b\ \texttt{do}\ S, s\rangle \to s$ hvis $s \vdash b \to_b ff$

7. *(10 point)* Betragt *big-step-semantikken* for **Bims**. Herunder er angivet transitionsreglerne for repeat-løkker. Udfyld det, der mangler.

(REPEAT-SAND) $\dfrac{\boxed{}}{\langle \texttt{repeat}\ S\ \texttt{until}\ b, s\rangle \rightarrow s'}$

hvis $\boxed{}$

(REPEAT-FALSK) $\dfrac{\boxed{}}{\langle \texttt{repeat}\ S\ \texttt{until}\ b, s\rangle \rightarrow s'}$

hvis $\boxed{}$

Svar

(REPEAT-SAND) $\dfrac{\langle S, s\rangle \rightarrow s'}{\langle \texttt{repeat}\ S\ \texttt{until}\ b, s\rangle \rightarrow s'}$

hvor $s' \vdash b \rightarrow_b tt$

(REPEAT-FALSK) $\dfrac{\langle S, s\rangle \rightarrow s' \quad \langle \texttt{repeat}\ S\ \texttt{until}\ b, s'\rangle \rightarrow s''}{\langle \texttt{repeat}\ S\ \texttt{until}\ b, s\rangle \rightarrow s''}$

hvor $s' \vdash b \rightarrow_b ff$

---

8. *(20 point)* Herunder er en big-step-transitionsregel for procedurekald med parametre.

$$\text{(CALL)}\quad \frac{env'_V[x \mapsto l][\text{next} \mapsto l'], env''_P \vdash \langle S, sto\rangle \to sto'}{env_V, env_P \vdash \langle\ \texttt{call}\ p(y), sto\rangle \to sto'}$$

hvor $env_P\ p = (S, x, env'_V, env'_P), env_V\ y = l$
og $l' = env_V\,\text{next}$
og $env''_P = env_P[p \mapsto (S, x, env'_V, env'_P)]$

a. For hver af nedenstående bemærkninger skal du sætte en ring om det sted i reglen, som passer med bemærkningen, og skrive bemærkningens nummer ud for ringen. (For nogle af de nedenstående bemærkninger er der flere mulige steder, man kan sætte en ring – det er nok at vælge én af mulighederne.)

1. Her er selve procedurekaldet
2. Vi slår op og finder procedurens krop
3. Dette er den aktuelle parameter
4. Dette er den formelle parameter
5. Her er den aktuelle parameters lokation
6. Her er variabel- og procedureenvironments, der gjaldt, da p blev erklæret
7. Her er p's krop
8. Her er variabel- og procedurenvironments, der gjaldt, da p blev kaldt

b. Hvad kaldes de scoperegler, som reglen udtrykker? Begrund dit svar.

c. Hvad kaldes den parametermekanisme, som reglen udtrykker? Begrund dit svar.

d. Er kald af proceduren p inde i p's krop rekursive eller ikke-rekursive ifølge denne regel? Begrund dit svar.

Svar

a.

$$(\textsc{call})\quad \frac{env'_V[x \mapsto l][\text{next} \mapsto l'], env''_P \vdash \langle S, sto\rangle \to sto'}{env_V, env_P \vdash \langle \texttt{call}\ p\ y, sto\rangle \to sto'}$$

hvor $env_P\ p = (S, x, env'_V, env'_P)$, $env_V\ y = l$
og $l' = env_V \text{next}$
og $env''_P = env'_P[p \mapsto (S, x, env'_V, env'_P)]$

b. Der er tale om statiske scoperegler, da de environments, som anvendes under udførelse af procedurens krop er de environments, som var kendt, da proceduren blev erklæret.

c. Parametermekanismen er call-by-reference. Den formelle parameter er en reference til den aktuelle parameter (de deler lokation).

d. Procedurekald i kroppen af p vil altid være rekursive, da de procedurebindinger som er kendt i kroppen S, er bindingerne der gjaldt, *inden* p blev erklæret og altså ikke indbefatter p selv, men er blevet opdateret med en binding af p selv i env''_P.

9. *(15 point)* Her er to af small-step-transitionsreglerne for parallel sammensætning, men der mangler noget.

$$(\textsc{or-1})\quad \frac{\langle S_1, s\rangle \Rightarrow \boxed{}}{\langle S_1\ \texttt{par}\ S_2, s\rangle \Rightarrow \boxed{}}$$

$$(\textsc{or-2})\quad \frac{\langle S_1, s\rangle \Rightarrow s'}{\langle S_1\ \texttt{par}\ S_2, s\rangle \Rightarrow \boxed{}}$$

a. Udfyld det, der mangler i reglerne ovenfor.

b. Skriv de resterende regler herunder.

$\boxed{}$

Svar

$$\text{(PAR-1)} \quad \frac{\langle S_1, s\rangle \Rightarrow \langle S_1', s'\rangle}{\langle S_1 \ \texttt{par}\ S_2, s\rangle \Rightarrow \langle S_1' \ \texttt{par}\ S_2, s'\rangle}$$

$$\text{(PAR-2)} \quad \frac{\langle S_1, s\rangle \Rightarrow s'}{\langle S_1 \ \texttt{par}\ S_2, s\rangle \Rightarrow \langle S_2, s'\rangle}$$

$$\text{(PAR-3)} \quad \frac{\langle S_2, s\rangle \Rightarrow \langle S_2', s'\rangle}{\langle S_1 \ \texttt{par}\ S_2, s\rangle \Rightarrow \langle S_1 \ \texttt{par}\ S_2', s'\rangle}$$

$$\text{(PAR-4)} \quad \frac{\langle S_2, s\rangle \Rightarrow s'}{\langle S_1 \ \texttt{par}\ S_2, s\rangle \Rightarrow \langle S_1, s'\rangle}$$

10. *(15 point)* Betragt alfabetet $\Sigma = \{a, b\}$. Lad os definere præfiksordningen $u \sqsubseteq v$ for vilkårlige strenge $u, v \in \Sigma^*$ ved

$$u \sqsubseteq v \quad \text{hvis der findes en } w \in \Sigma^* \text{ så } uw = v$$

a) Find to strenge u og v, så $u \sqsubseteq v$.

b) Bevis, at $(\Sigma^*, \sqsubseteq)$ er en partielt ordnet mængde.

Svar

a. Eksempler er $u = aab$ og $v = aabaa$.

b. Vi skal vise, at $\sqsubseteq$ er refleksiv, antisymmetrisk og transitiv.

$\sqsubseteq$ **er refleksiv:** Vi skal vise at $x \sqsubseteq x$ for alle x. Men dette er klart, da vi ved at vælge $w = \varepsilon$ har at $x\varepsilon = x$.

$\sqsubseteq$ **er antisymmetrisk:** Vi skal vise at hvis $x \sqsubseteq y$ og $y \sqsubseteq x$ så har vi $x = y$. Hvis $x \sqsubseteq y$, så har vi en w_1 så $xw_1 = y$. Hvis $y \sqsubseteq x$, så har vi en w_2 så $yw_2 = x$. Men så ser vi ved indsætning, at $x = xw_1w_2$. Det kan kun lade sig gøre, hvis $w_1 = w_2 = \varepsilon$, og derfor har vi $x = y$.

$\sqsubseteq$ **er transitiv:** Vi skal vise at hvis $x \sqsubseteq y$ og $y \sqsubseteq z$ så har vi at $x \sqsubseteq z$. Hvis $x \sqsubseteq y$, så har vi en w_1 så $xw_1 = y$. Hvis $y \sqsubseteq z$, så har vi en w_2 så $yw_2 = z$. Men så har vi at $xw_1w_2 = z$, så ved at vælge w til w_1w_2 ser vi, at $x \sqsubseteq z$.