Syntaks og semantik

Skriftlig eksamen

13. juni 2019 kl. 10.00-13.00

Navn

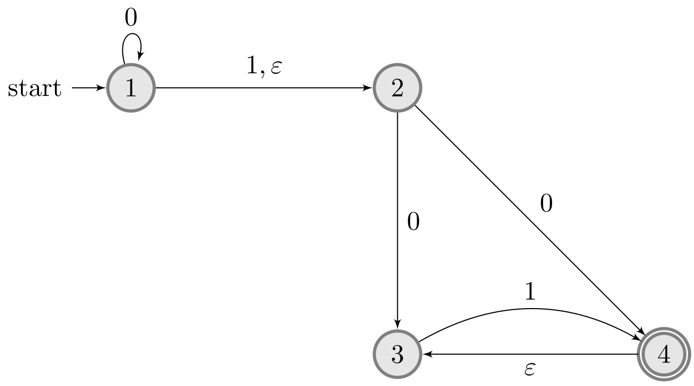
VIGTIGT: Læs dette, inden du går i gang!

- Dette eksamenssæt er på i alt 7 trykte sider, inklusive denne forside.
- I besvarelsen skal du udfylde de tomme kasser med tekst og figurer. ***Skriv ikke på bagsiden af arkene – kun de trykte sider af besvarelserne bliver kopieret.***
- Eneste tilladte hjælpemidler er blyant/kuglepen/viskelæder o.lign. og den formelsamling, som er udleveret sammen med dette eksamenssæt.
- Du kan få kladdepapir ved at henvende dig til en eksamensvagt.
- Du kan kun aflevere selve besvarelsen på disse ark (du kan altså *ikke* aflevere supplerende besvarelse på kladdepapir!).

Opgave 1

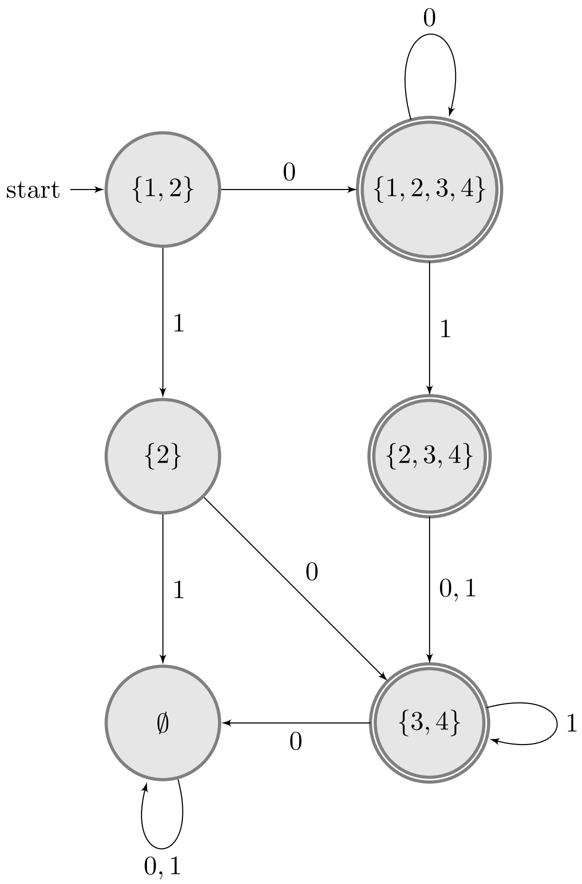
(10 point)

Herunder er en nondeterministisk endelig automat.



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

Løsning:



Opgave 2

(10 point)

Her er fem påstande om regulære og kontekstfrie sprog. Sæt kryds ved de svar, du mener er de korrekte.

	Ja	Nej
Der findes regulære sprog, som ikke er kontekstfrie sprog.		**Løsning:** NEJ
Sproget $\{a^i b^i \mid i \geq 0\}$ er et kontekstfrit sprog.	**Løsning:** JA	
Hvis $L_1 \subseteq L_2$ og L_2 er regulært, er L_1 også regulært.		**Løsning:** NEJ
Hvis L er et regulært sprog, kan der kun være endeligt mange strenge i L.		**Løsning:** NEJ
Sproget $\{ww \mid w \in \{a,b\}^* \geq 0\}$ er kontekstfrit.		**Løsning:** NEJ

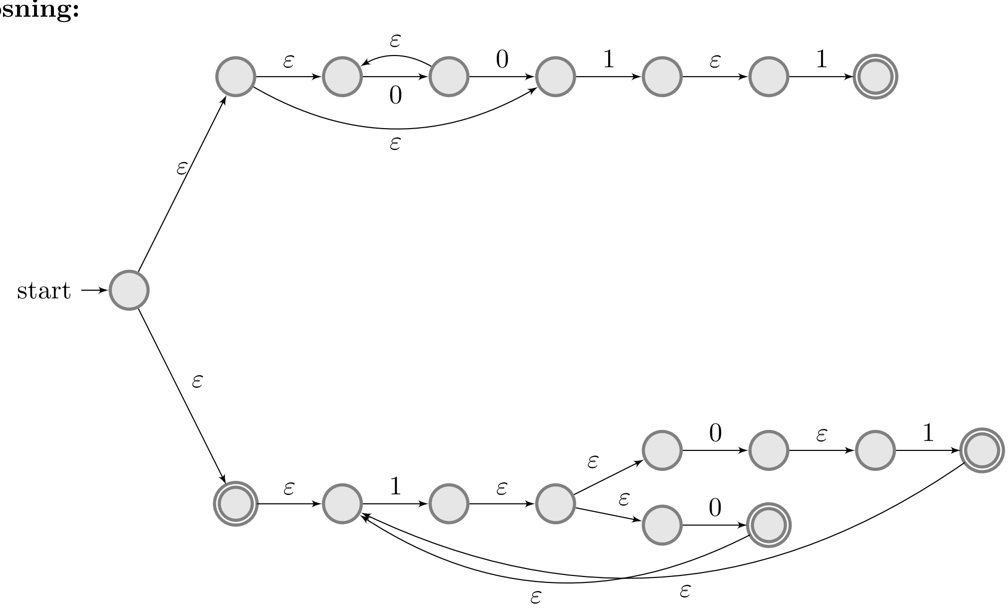
Opgave 3

(10 point) Her er et regulært udtryk R over alfabetet $\{0,1\}^*$.

$$(\mathtt{0}^*\mathtt{11}) \cup (\mathtt{1}(\mathtt{01} \cup \mathtt{0}))^*$$

Konstruer ved brug af resultaterne fra dette kursus en NFA, der er ækvivalent med R.

Løsning:



Opgave 4

(20 point)
Betragt sproget

$$L_1 = \{a^i b^j a^i \mid i, j \geq 0\}$$

1. Bevis at L_1 ikke er et regulært sprog.

   Løsning:

   Vi anvender Pumping Lemma for regulære sprog. Antag at L_1 var et regulært sprog. Da ville der være et $p > 0$ således at enhver $s \in L_1$ hvor $|s| \geq p$ ville have en opsplitning $s = xyz$ med

   a) $xy^iz \in L_1$ for alle $i \geq 0$
   b) $|y| > 0$
   c) $|xy| \leq p$

   Vælg $s = a^p b a^p$. Det er klart at $|s| = 2p + 1 > p$, så s burde have en god opsplitning. Men vi viser nu, at det ikke er tilfældet.

   Hvis betingelsen $|xy| \leq p$ skal gælde, kan x og y kun bestå af a'er fra den første delstreng af a'er. Dvs. at $y = a^k$. Hvis $|y| > 0$ også skal gælde, må vi have at $y = a^k$ for $k > 0$. Men så overtrædes betingelsen at $xy^iz \in L_1$ for alle $i \geq 0$, for $xy^2z = a^{p+k}ba^p \notin L_1$.

2. Bevis at L_1 er et kontekstfrit sprog.

   Løsning: For at vise at sproget L_1 er kontekstfrit, kan vi konstruere en kontekstfri grammatik, der beskriver det. Her er en sådan:

$$S \to \varepsilon \mid aSa \mid B$$
$$B \to bB \mid \varepsilon$$

Opgave 5

(10 point)
I din besvarelse af denne opgave skal du anvende *den simple tilstandsmodel.*

Til **Bims** tilføjer vi en ny kommando, `equate` x `with` y.

Ideen med denne nye kommando er følgende: Hvis variablerne x og y har samme værdi, skal der intet ske. Men hvis variablerne x og y ikke har samme værdi, skal x tildeles den værdi, som y har.

Giv en big-step-semantik for den nye kommando.

Løsning:

(EQUATE) $\langle \texttt{equate } x \texttt{ with } y, s\rangle \to s[x \mapsto v]$
hvor $v = sy$

Opgave 6

(20 point) I din besvarelse af denne opgave skal du anvende *den simple tilstandsmodel.*

Til **Bims** tilføjer vi en break-and-exit-kommando `brexit` b_1 S b_2 `end`, så opbygningsreglerne for **Kom** er udvidet på følgende vis.

$$S ::= \cdots \mid \texttt{brexit } b_1 \ S \ b_2 \texttt{ end}$$

Ideen er at `brexit` b_1 S b_2 `end` skal udføres således:

- Hvis betingelsen b_1 ved indgangen til løkken evaluerer til sand, skal S udføres; hvis b_2 efter udførelsen af S evaluerer til sand, gentages hele `brexit`-løkken, men hvis b_2 evalueres til falsk, skal `brexit`-løkken forlades.
- Hvis betingelsen b_1 ved indgangen til løkken evalueres til falsk, skal løkken forlades.

1. Giv en big-step-semantik for `brexit` b_1 S b_2 `end`, der svarer til den uformelle beskrivelse af kommandoen. Du skal udvide den oprindelige big-step-semantik for **Bims**.

   Løsning:

   (BREXIT-FALSK) $\langle \texttt{brexit } b_1 \ S \ b_2 \texttt{ end}, s\rangle \to s$ hvis $s \vdash b_1 \to_b \mathit{ff}$

   (BREXIT-SAND-1) $$\frac{\langle S, s\rangle \to s'}{\langle \texttt{brexit } b_1 \ S \ b_2 \texttt{ end}, s\rangle \to s'} \quad \text{hvis } s \vdash b_1 \to_b \mathit{tt} \text{ og } s' \vdash b_2 \to_b \mathit{ff}$$

   (BREXIT-SAND-2) $$\frac{\langle S, s\rangle \to s' \quad \langle \texttt{brexit } b_1 \ S \ b_2 \texttt{ end}, s'\rangle \to s''}{\langle \texttt{brexit } b_1 \ S \ b_2 \texttt{ end}, s\rangle \to s''} \quad \text{hvis } s \vdash b_1 \to_b \mathit{tt} \text{ og } s' \vdash b_2 \to_b \mathit{tt}$$

2. Giv en small-step-semantik for `brexit` b_1 S b_2 `end`, der , der svarer til den uformelle beskrivelse af kommandoen. Du skal udvide den oprindelige small-step-semantik for **Bims**.

   Løsning:

   $$\langle \texttt{brexit } b_1 \ S \ b_2 \texttt{ end}, s\rangle \Rightarrow \langle \texttt{if } b_1 \texttt{ then } (S; \texttt{if } b_2 \texttt{ then brexit } b_1 \ S \ b_2 \texttt{ end else skip}) \texttt{ else skip}, s\rangle$$

3. Giv en kommando i den udvidede udgave af **Bims**, som er semantisk ækvivalent med `while` b `do` S mht. big-step-semantikken.

   Løsning:

   Kroppen af en repeat-løkke skal udføres, indtil betingelsen bliver sand, dvs. mindst én gang. Derfor har vi

   $$\texttt{while } b \texttt{ do } S \sim_{bss} \texttt{brexit } b \ S \ (0 = 0) \texttt{ end}$$

4. Giv en kommando i den udvidede udgave af **Bims**, som er semantisk ækvivalent med `repeat` S `until` b mht. big-step-semantikken.

   Løsning:

   $$\texttt{repeat } S \texttt{ until } b \sim_{bss} \texttt{brexit } (0 = 0) \ S \ \neg b \texttt{ end}$$

Opgave 7

(10 point) Herunder er en big-step-transitionsregel for procedurekald med én parameter.

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

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

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.) Vi har allerede sat én ring og givet den et nummer, så du kan se hvordan du skal gøre.

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 variabelenvironmentet, der gjaldt, da p blev erklæret
7. Her er p's krop
8. Her er variabelenvironmentet, der gjaldt, da p blev kaldt

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

Løsning: Der er tale om statiske scoperegler for variabler og dynamiske scoperegler for procedurer; de variabelbindinger som er kendt under udførelsen af p's krop er de bindinger, der gjaldt, da p blev erklæret, mens de procedurebindinger, som er kendt under udførelsen af p's krop er bindingerne på kaldstidspunktet.

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

Løsning: Der er tale om call-by-reference/referenceparameter; den formelle parameter x peger på samme lokation som den aktuelle parameter y og er dermed en reference til y.

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

Løsning: Ja, alle kald af p inde i p's krop vil være rekursive, da der er tale om dynamiske scoperegler for procedurer.

Opgave 8

(10 point) Lad alfabetet $\Sigma = \{a, b\}$. Vi definerer den binære relation $\sqsubseteq$ over strenge over Σ ved

$$u \sqsubseteq v \text{ hvis } \exists w_1, w_2 \in \Sigma^*. w_1 u w_2 = v$$

dvs. $u \sqsubseteq v$ hvis u er en delstreng af v.

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

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

   - Vi skal vise at $u \sqsubseteq u$ for alle $u \in \Sigma^*$. Det er klart at hvis $w_1 = w_2 = \varepsilon$, har vi at $w_1 u w_2 = u$.

- Vi skal vise, at hvis $u \sqsubseteq v$ og $v \sqsubseteq u$ så har vi at $u = v$.

  Da $u \sqsubseteq v$, findes der w_1, w_2 så $w_1 u w_2 = v$. Og da $v \sqsubseteq u$, findes der w_3, w_4 så $w_3 v w_4 = u$. Men så har vi at $w_1 w_3 v w_2 w_4 = v$, og det kan kun gælde, hvis $w_1 = w_2 = w_3 = w_4 = \varepsilon$. Det betyder at $u = v$.

- Vi skal vise, at hvis $u \sqsubseteq v$ og $v \sqsubseteq w$ så har vi at $u \sqsubseteq w$.

  Da $u \sqsubseteq v$, findes der w_1, w_2 så $w_1 u w_2 = v$. Og da $v \sqsubseteq w$, findes der w_3, w_4 så $w_3 v w_4 = w$. Men så har vi at $w_1 w_3 u w_2 w_4 = w$, og derfor at $u \sqsubseteq w$.

2. Lad os definere operationen w^R for en vilkårlig streng $w \in \Sigma^*$ ved

$$\varepsilon^R = \varepsilon$$
$$(a_1 \dots a_n)^R = a_n \dots a_1$$

dvs. at w^R er strengen w læst bagfra. Betragt nu funktionen $f : \Sigma^* \to \Sigma^*$ defineret ved at

$$f(x) = x^R$$

Bevis at f er en monoton funktion mht. ordningen $\sqsubseteq$.

Løsning:

Vi skal vise, at hvis $u \sqsubseteq v$, så har vi også at $f(u) \sqsubseteq f(v)$.

Hvis $u = \varepsilon$, er det klart at $u^R = \varepsilon$ også er en delstreng af v^R.

Hvis $u = a_1 \dots a_n$, har vi at $w_1 a_1 \dots a_n w_2 = v$. Men så har vi også at $(w_2)^R a_1 \dots a_n (w_1)^R = v^R$, og dermed igen at $u^R \sqsubseteq v^R$.

3. Find et fikspunkt for funktionen f, som vi definerede i delspørgsmålet lige før (du behøver ikke at angive, hvordan du fandt det).

Løsning: Et fikspunkt for f er en streng u så $f(u) = u$, dvs. en streng u så $u = u^R$. Ethvert palindrom opfylder denne betingelse, så f.eks. er ε og *abba* fikspunkter for f. Det mindste fikspunkt for f er det korteste palindrom, dvs. ε.