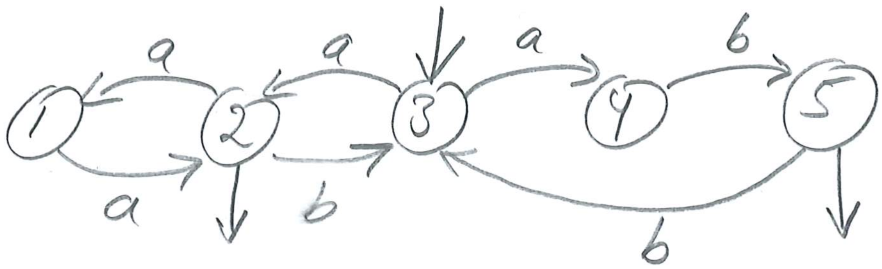
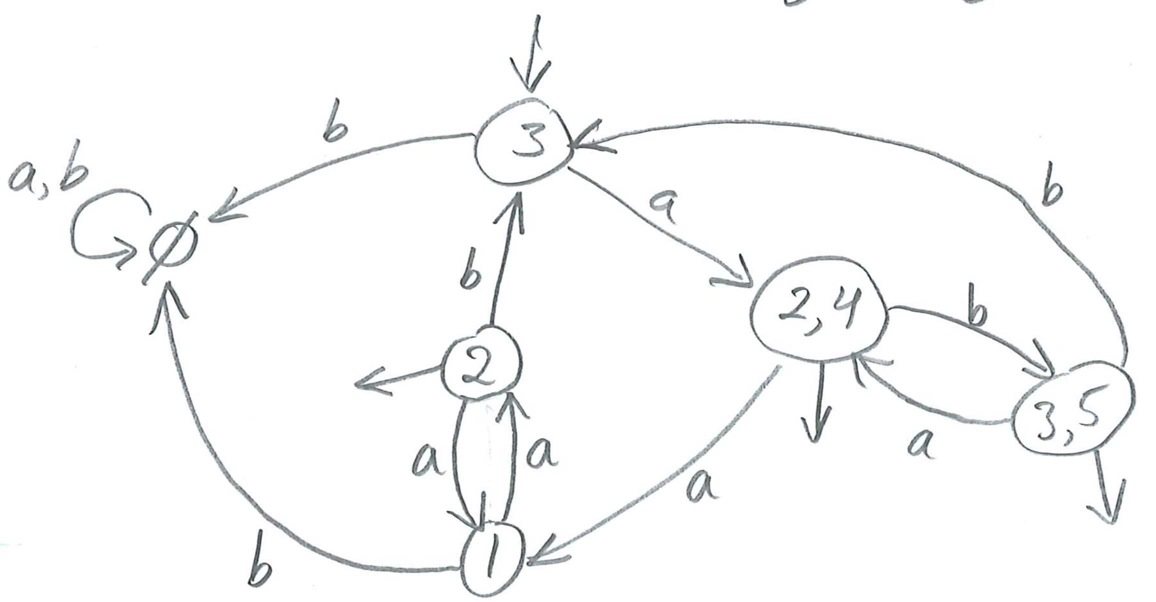
Tenta 2019-10-30

Svar/lösningsförslag

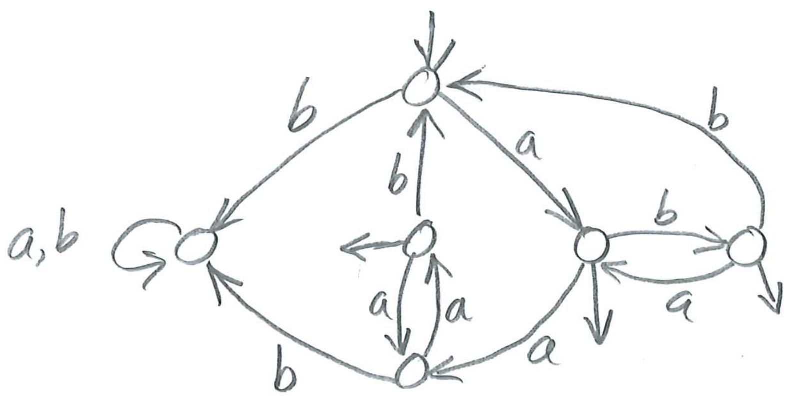
1. Först görs NFA:n icke-glupsk och tillstånden namnges:



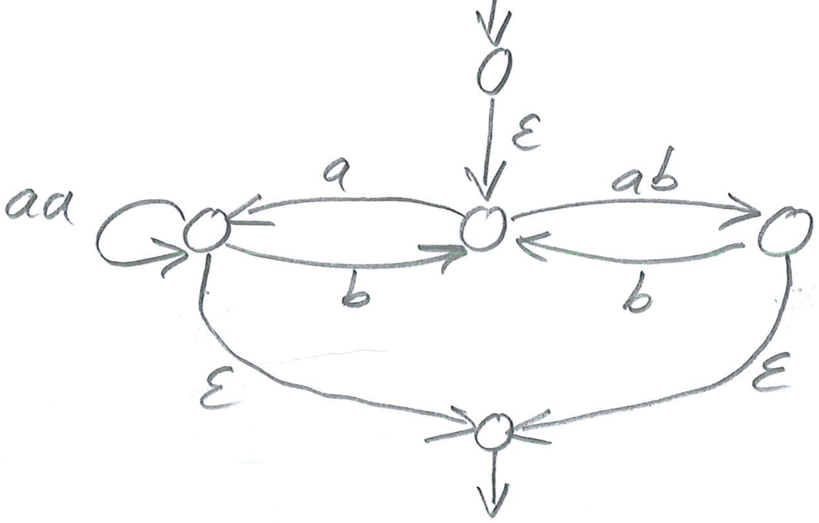
Sedan används delmängdsalgoritmen:



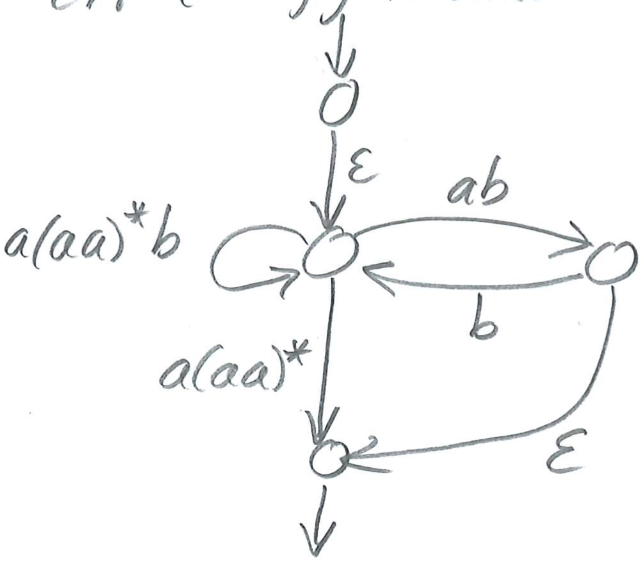
Tillsnyggat:

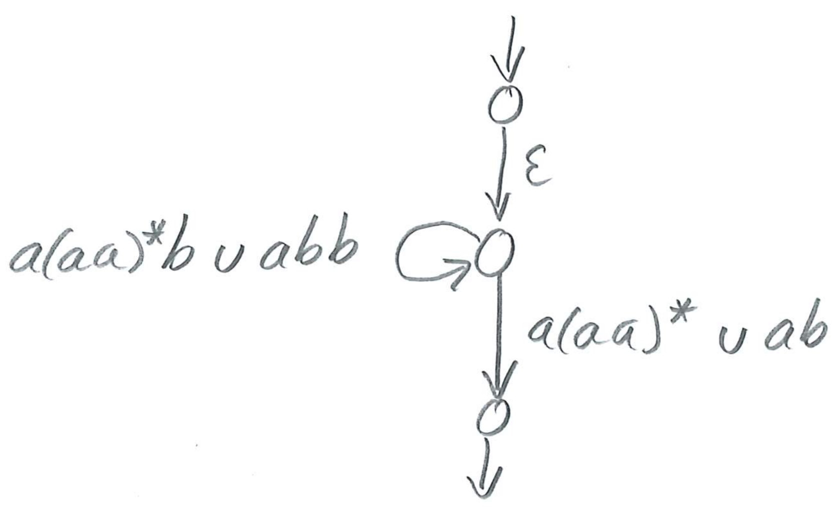


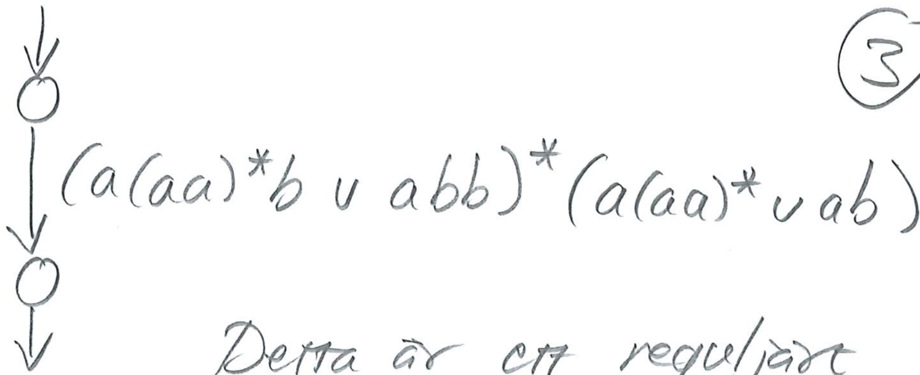
2. Först läggs nytt start- och accepterande tillstånd till:



Sedan elimineras alla gamla tillstånd, ett för ett (och jag förenklar när det är möjligt):

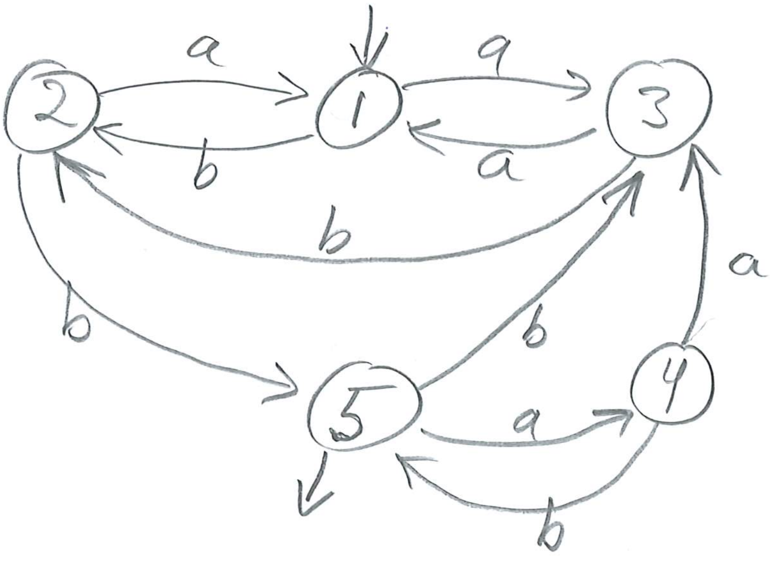






Detta är ett reguljärt uttryck för språket som den ursprungliga NFA:n accepterar.

3. Man ser att ett av tillstånden aldrig kan nås, så det tillståndet tar vi bort på en gång och sedan numreras tillstånden



så att vi kan skriva upp en övergångstabell:

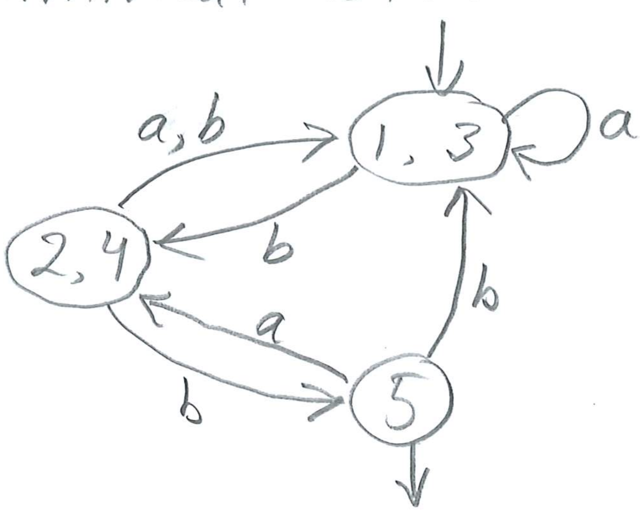
	1	2	3	4	5
a	3	1	1	3	4
b	2	5	2	5	3

Nu används särskiljandealgoritmen:

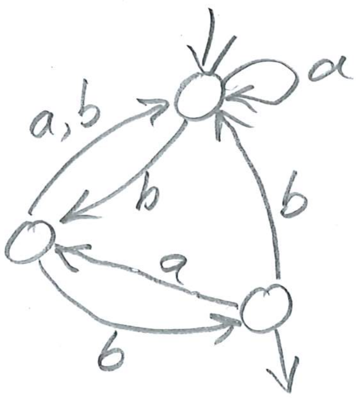
nivå	sönderdelningar
1	{1, 2, 3, 4} {5}
2	{1, 3} {2, 4} {5}
3	{1, 3} {2, 4} {5}

Proceduren terminerar när de två sista nivåerna är likadana.

En minimal DFA:



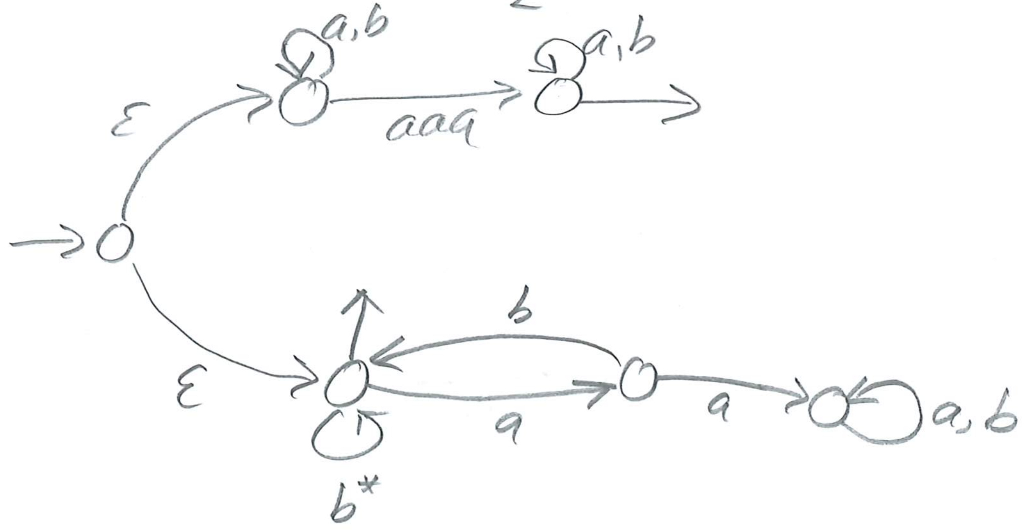
Tillsnyggat:



4. L_2 är reguljärt och har följande reguljära uttryck:

$$(a \cup b)^* aaa (a \cup b)^* \cup b^*(abb^*)^*$$

En NFA för L_2 är (t.ex.)



L_1 är <u>inte</u> reguljärt.

Bevis med särskiljandesatsen:
Låt $A = \{b^n : n = 1,2,3,4,\dots\}$, så A är oändlig. Låt $x, y \in A$ vara olika, så $x = b^i$ och $y = b^j$ där $i,j > 0$ och $i \neq j$. Låt k vara den minsta av i och j ($k = \min\{i,j\}$).
Välj $z = a^{k+2}$. Då kommer precis en av $xz = b^i a^{k+2}$ och $yz = b^j a^{k+2}$ att tillhöra L_1.

Det följer att A särskiljs av L_1 och särskiljandesatsen säger nu att L_1 <u>inte</u> är reguljärt.

Bevis med pumpsatsen:

1. L_1 är oändligt för $b^n a^{n+2} \in L_1$ för alla $n > 0$.
2. Antag att L_1 är reguljärt.
3. Låt N vara givet av pumpsatsen för reguljära språk.
4. Välj $u = b^{N+1}$, $w = a^{N+3}$, $v = \varepsilon$, så $uwv = b^{N+1} a^{N+3} \in L_1$ och $|w| \geq N$.
5. Antag att $w = xyz$ och $y \neq \varepsilon$. Då gäller att $xy^0 z = xz = a^m$ där $m \leq N+2$, så $uxzv = b^{N+1} a^m \notin L_1$.
6. Slutsatsen i punkt 5 motsäger pumpsatsen för reg. språk, så L_1 är <u>inte</u> reguljärt.

5 (a) Strängen abaabbba accepteras och här är en körning som leder till acceptans:

tape	stack	tillstånd
abaabbba	ε	v
abaabbba	a	h
abaabbba	ε	h
abaabbba	a	v
abaabbba	aa	h
abaabbba	a	h
abaabbba	ε	h
abaabbba	b	h
abaabbba	ε	v

(som jag kallar v = vänster och h = höger)

vänster tillstånd är accepterande.

Strängen abaabb accepteras inte för varje sätt att läsa av en sträng med tre a:n slutar i det högra tillståndet som inte är accepterande.

(b) PDA:ns språk består av alla strängar (över {a,b}) som har ett jämnt antal 'a' och lika många 'a' som 'b'.

6. (a) Körning för ababa:

#ababa
ababa
#baba
baba#
baba
bab#
#bab
bab
#ab
ab#
ab
a#
#a
a
#
#
#
nej

Jag gör inte körningen för abaaba utan säger bara att den körningen slutar med att TM:en skriver 'ja'.

(b) TM:en besvarar frågan "Är inputsträngen ett palindrom av jämn längd?"
(Alternativt: Har inputsträngen formen ww^{rev}?)

7. L_0 är reguljärt eftersom det beskrivs av ett reguljärt uttryck. Det följer att $L_3 = L_0 L_0$ är reguljärt (för sammanfogning av reguljära språk bildar ett reguljärt språk), och därmed också sammanhangsfritt

Vi har $L_2 = \{uu : u \in L_0\} =$

$$= \{a^n bba^m a^n bba^m : n, m \in \mathbb{N}\} =$$

$$= \{a^n bba^n a^m bba^m : n, m \in \mathbb{N}\},$$

så en CFG för L_2 ges av

$$S \to TT$$
$$T \to aTa \mid bb$$

och därmed är L_2 sammanhangsfritt.

L_2 är inte reguljär, vilket jag visar med särskiljandesatsen. Låt

$$A = \{a^n bb : n \in \mathbb{N}\},$$ så A är oändlig.

Låt $x, y \in A$ vara olika så $x = a^i bb$ och $y = a^j bb$ där $i \neq j$. Kom ihåg att

$$L_2 = \{a^n bba^n a^m bba^m : n, m \in \mathbb{N}\}.$$

Låt $z = a^i bb$. Då gäller att
$xz = a^i bb a^i bb = a^i bb a^i a^0 bb a^0 \in L_2$
men $yz = a^j bb a^i bb \notin L_2$.
Alltså särskiljs A av L_2.

Från definitionerna följer att $L_2 \subseteq L_3$
så $L_5 = L_2 \cup L_3 = L_3$ och därmed
är L_5 reguljärt (och sammanhangsfritt).

L_4 är inte sammanhangsfritt och därmed
inte reguljärt. Pumpsatsbevis:

1. L_4 är oändlig för $a^n bb a^n a^n bb a^n \in L_4$ för alla $n \in \mathbb{N}$.
2. Antag att L_4 är en CFL.
3. Låt K vara givet av pumpsatsen för CFL.
4. Låt $w = a^{K+1} bb a^{2(K+1)} bb a^{K+1}$, så $w \in L_4 = \{ a^n bb a^{2n} bb a^n : n \in \mathbb{N} \}$ och $|w| \geq K$.

5. Antag att $w = uvxyz$ där $vy \neq \varepsilon$ och $|vxy| \leq K$.

Då gäller att $w = uv^0xy^0z = uxz = a^i b^j a^k b^l a^m$ där

$i < K+1$ eller
$j < 2$ eller
$k < 2(K+1)$ eller
$l < 2$ eller
$m < K+1$.

I fallen $j<2$ och $l<2$ så $uxz \notin L_4$ pga att alla strängar i L_4 har fyra b:n, men uxz har högst tre b:n i dessa fall.
Om $i < K+1$ så $m = K+1$ och därmed $uxz \notin L_4$. Om $m < K+1$ så $i = K+1$ och på samma sätt $uxz \notin L_4$.
Om $k < 2(K+1)$ så $i = K+1$ eller $m = K+1$ och det följer att $uxz \notin L_4$.

6. Slutsatsen i punkt 5 motsäger pumpsatsen för CFL, så L_4 är <u>inte</u> sammanhangsfritt (dvs. inte en CFL).

8. (a) Svaret är ja och här följer en informell algoritm som avgör problemet. (Enligt Church-Turings tes finns då en TM som avgör problemet.)

<u>Input</u>: Godtycklig NFA M.
Gör om (med delmängdsalgoritmen) M till en DFA N så att $L(N) = L(M)$.
Kör N på alla (ändligt många) strängar w över N:s alfabet sådana att $5 \leq |w| \leq 100$ och de 5 första tecknen i w är likadana. Om N accepterar någon av dessa strängar så svara 'ja'. I annat fall, svara 'nej'.

(b) Svaret är nej och jag använder Rices sats för att visa detta.
Låt $\Omega = \{L(M) :$ M är en TM som accepterar någon sträng w sådan att $5 \leq |w| \leq 100$ och de 5 första tecknen i w är likadana$\}$.

Från definitionen av Ω följer direkt att alla språk i Ω är TM-accepterbara.

Språket $\{a^5\}$ är reguljärt och därmed TM-accepterbart, så $\{a^5\} = L(M)$ för någon TM M och därmed så

$\{a^5\} \in \Omega$ och $\Omega \neq \emptyset$.

Språket $\emptyset$ är reguljärt och därmed TM-accepterbart. Eftersom $\emptyset \notin \Omega$ så innehåller <u>inte</u> Ω alla TM-accepterbara språk.

Från Rices sats följer nu att ingen TM kan avgöra, för godtycklig TM M, om $L(M) \in \Omega$. Så svaret till (b) är nej.

9. Antag att L är TM-avgörbar så då finns en TM M som avgör L. Här är en algoritm som, givet input w, svarar 'ja' om $w \in L^*$ och 'nej' annars:

Om $w = \varepsilon$ så svara 'ja', eftersom $\varepsilon \in L^*$ (oavsett vad L är).

Om $w \neq \varepsilon$ så dela upp w i icketomma delar $w = v_1 v_2 \ldots v_n$ på alla möjliga sätt (för alla möjliga $1 \leq n \leq |w|$).

För varje sådan uppdelning $w_1 = v_1 \ldots v_n$, kör M på v_i för varje $i = 1, \ldots, n$.

Om, för någon uppdelning $w = v_1 \ldots v_n$, M svarar 'ja' vid körning på v_i <u>för varje</u> $i = 1, \ldots, n$, så svara 'ja'.

I annat fall, svara 'nej'.