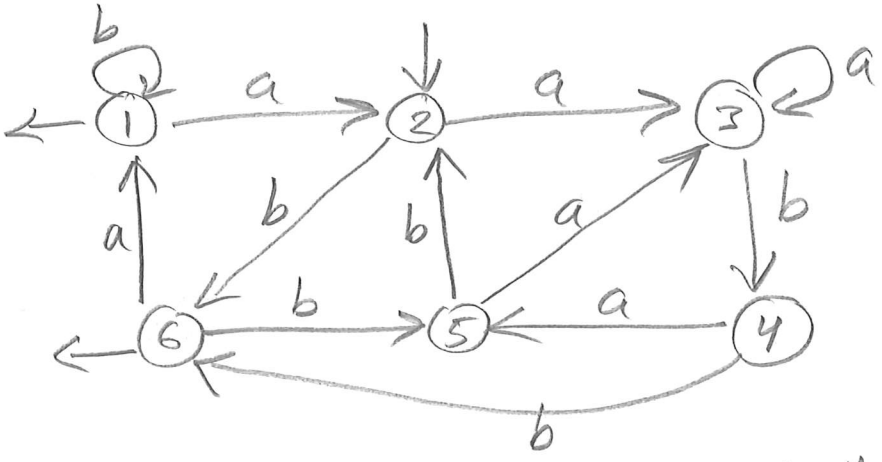
Tenta 2021-12-21
Svar eller lösningsförslag

1. Om tillstånden numreras så här



så får man övergångstabellen

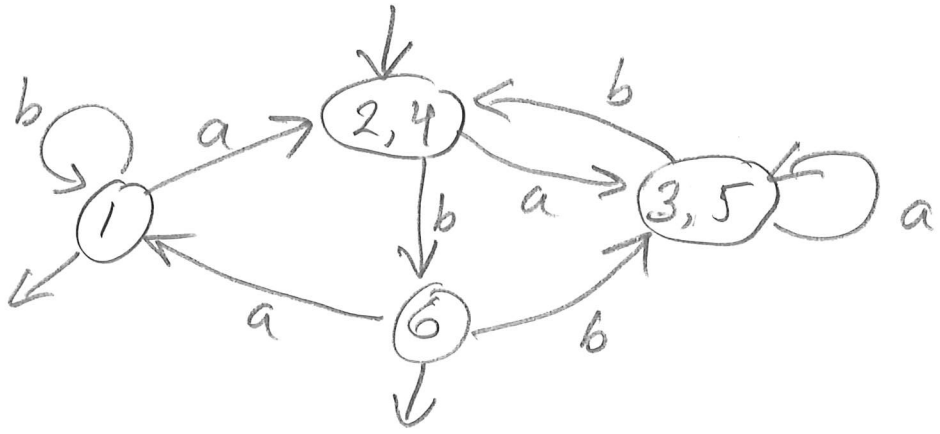
	1	2	3	4	5	6
a	2	3	3	5	3	1
b	1	6	4	6	2	5

Sedan används särskiljandealgoritmen som ger följande uppdelningar:

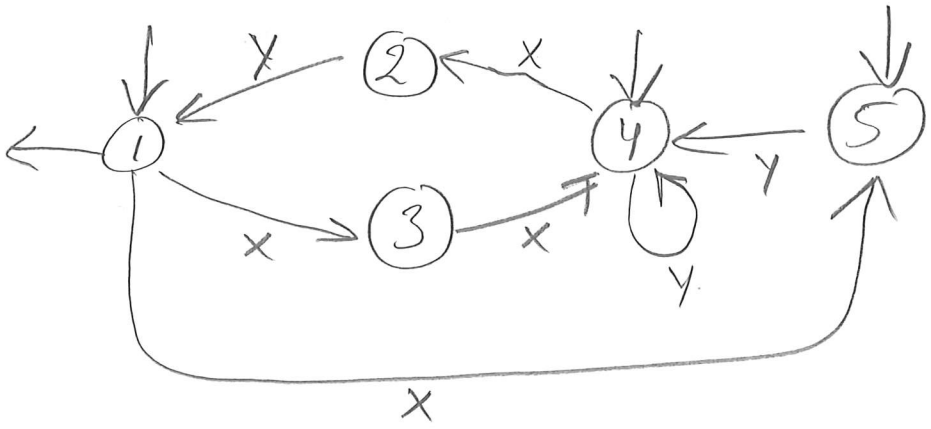
nivå	sönderdelning	
1	{1, 6} {2, 3, 4, 5}	accept. och icke-accept skiljs åt.
2	{1} {6} {2, 4} {3, 5}	1 och 6 särskiljs pga b. 3 och 4 särskiljs pga b.
3	{1} {6} {2, 4} {3, 5}	Finns inga skäl att särskilja andra tillstånd.

— blir ingen skillnad på nivå 3 så vi är klara

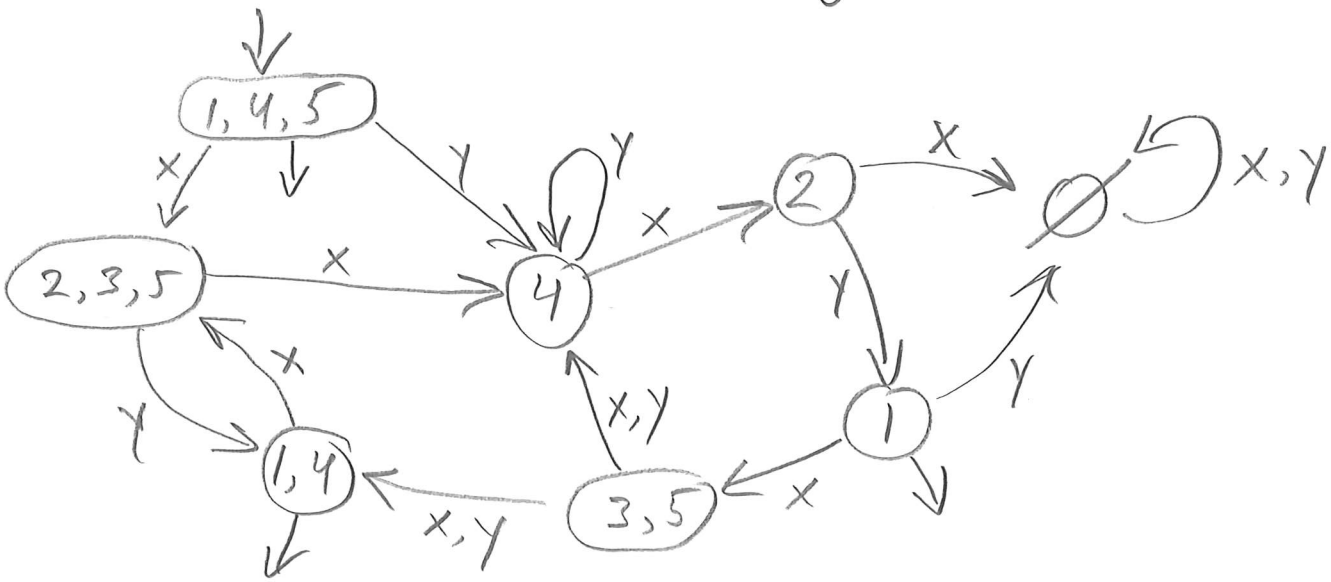
En minimal DFA fås genom att "slå ihop" 2 och 3 samt 4 och 5: (2)



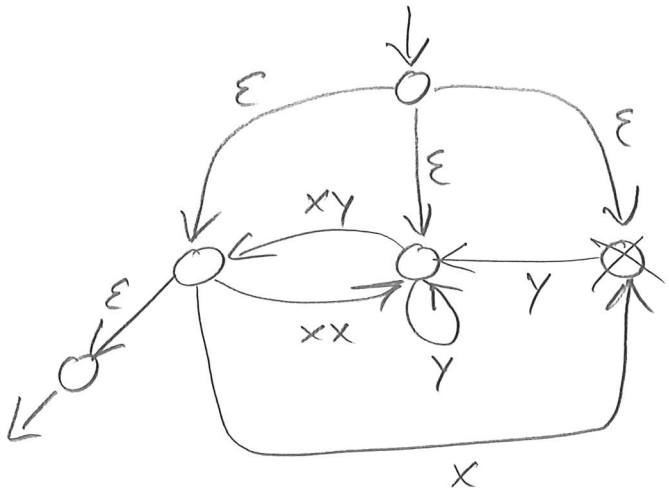
2. Först görs NFA:n icke-glupsk och tillstånden namnges:



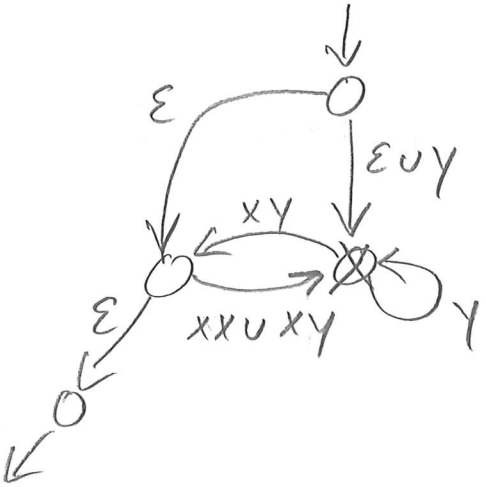
Sedan används delmängdsalgoritmen:

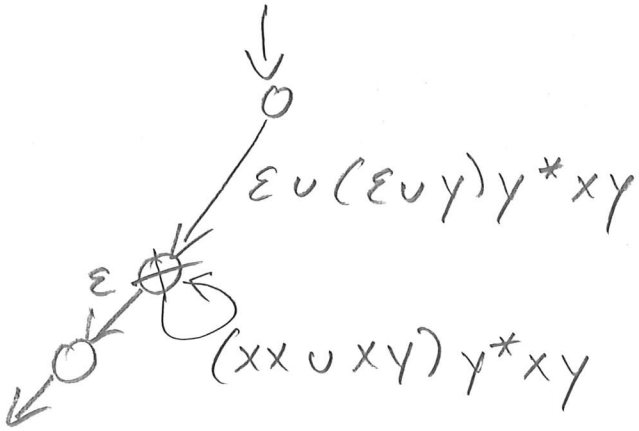


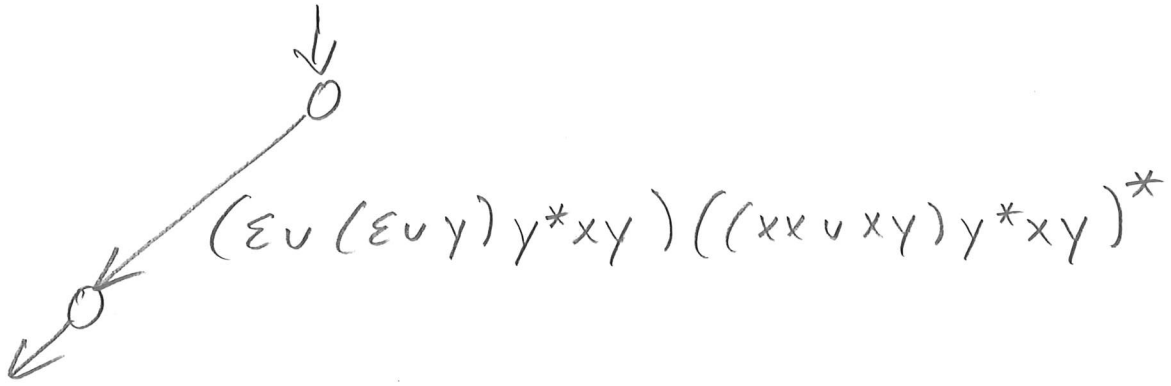
3. Nytt start- och accept. tillst. läggs till.



Sedan elimineras de gamla tillstånden, ett för ett, och jag förenklar också när det är möjligt







Det sista reguljära uttrycket beskriver NFA:ns språk.

4. L_2 är reguljärt och har följande reguljära uttryck:

$$aa(a\cup b)^*aa \cup bb(a\cup b)^*bb \cup$$
$$ab(a\cup b)^*ba \cup ba(a\cup b)^*ab$$

L_1 är inte reguljärt, och här är ett pumpsatsbevis av detta:

1. L_1 är vändlig för $a^nbba^n \in L_1$ för alla $n \in \mathbb{N}$.
2. Antag att L_1 är reguljärt.
3. Låt N vara givet av pumpsatsen för L_1.
4. Välj (tex.) $u=\varepsilon$, $w=b^{N+1}$, $v=aab^{N+1}$ så $uwv = b^{N+1}aab^{N+1} \in L_1$ och $|w| \geq N$.

5. Antag att $w=xyz$ och $y \neq \varepsilon$. Då finns $m>0$ så att $y=b^m$, och vi får $uxy^3zv = b^{N+1+m}aab^N$.
Observera att de två mittersta tecknen i $b^{N+1+m}aab^N$ är 'b'. Om det skulle finnas s, t så att $|t|=2$ och $b^{N+1+m}aab^N = s^{rev}ts$ så skulle s innehålla något 'a' samtidigt som s^{rev} bara innehåller b:n, vilket är omöjligt. Så vi drar slutsatsen att

$b^{N+1+m}aab^N$ inte kan skrivas på formen $s^{rev}ts$ där $|t|=2$ och det följer att $uxy^3zv \notin L_1$.

6. Punkt 5 motsäger pumpsatsen för reguljära språk så L_1 kan inte vara reguljärt.

Man kan också använda särskiljandesatsen och visa att den oändliga mängden

$$A = \{ b^n aa : n = 1, 2, 3, \ldots \}$$

särskiljs av L_1.

---

5. (a) Strängen aabaaaab accepteras men inte aababa. En körning på aabaaaab, där jag bara skriver ut symbolerna som inte har avlästs:

tape	stack
aabaaaab	ε
abaaaab	ε
baaaab	ε
aaaab	b
aaab	ab
aab	aab

tape	stack
ab	ab
b	b
#	ε

hela strängen har avlästs, stacken är tom och vi är i det accepterande tillståndet.

PDA:n kan börja att läsa in a:n utan att göra något på stacken. Sedan kan den gå in i det andra tillståndet där den samtidigt som den läser tecken (a:n eller b:n) lägger samma tecken på stacken i <u>omvänd</u> ordning. För att stacken ska vara tom när man har läst hela strängen måste man därför kunna övergå till det tredje tillståndet (vid någon lämplig tidpunkt) och på tapen ha kvar reverseringen av den delsträng man tidigare har avläst i det andra tillståndet.

Det följer att alla strängar i PDA:ns språk har formen $a^n w w^{rev}$ för något $n \in \mathbb{N}$ och vilken $w \in \{a,b\}^*$ som helst. Eftersom aababa ej har denna form så accepteras inte aababa

(b) En CFG för PDA:ns språk, som motiveras av resonemanget i del (a):

$S \to UV$, $U \to Ua \mid \varepsilon$, $V \to aVa \mid bVb \mid \varepsilon$.

(S, U, V är icke-terminerande)

6. (a)

#abaab
xabaab
xabaab#
xabaaby
xabaaby
xabaaby
x#baaby
x#baaby
#x#baaby
1x#baaby

1x#baaby
1x#baaby
1x##aaby
1x##aaby
1x##aby
1x##aby
#1x###aby
11x##aby
11x##aby
11x###aby

11x####by
11x####by
#11x####by
111x####by
111x####by
111x####by
111x####by
111x####by
111x#####y
111x#####y
111x#######
111x
111#
#111 terminerar

(b) Om TM:en startas med w på tapen så stannar den med 1^n på tapen där n är antalet a i w.

7. (a) Problemet är avgörbart. Enligt Church-Turings tes räcker det att informellt beskriva en algoritm som avgör problemet.

Algoritm: Input: en NFA M.
Omvandla M till en DFA N (med tex. delmängdsalgoritmen) med samma språk som M. Låt s vara antalet tecken i N:s alfabet och låt t vara antalet tillstånd som N har.
Undersök om man kan följa övergångar från starttillståndet till något accepterande tillstånd och passera något tillstånd två gånger. Om det går så är $L(N)$ (och $L(M)$) oändlig och då accepterar N (och M) någon sträng av längd minst 1000, så algoritmen svarar ja. Annars så accepterar N (och M) bara strängar av längd högst s·t. Så vi kör N på alla strängar (över N:s alfabet) med längd högst s·t, och denna procedur terminerar

eftersom det finns högst $s^{s \cdot t + 2}$ sådana strängar. Om N accepterar någon av dem som har längd minst 1000 så svarar algoritmen ja och i annat fall nej.

(b) Problemet är <u>oavgörbart</u> och vi visar det med Rices sats.

Låt $\Omega = \{ L : L$ är ett TM accepterbart språk och innehåller en sträng vars längd är minst $1000 \}$

Det följer direkt att alla språk i Ω är TM-accepterbara. Språket med reguljärt uttryck a^* är reguljärt och därmed TM-accepterbart och innehåller strängar med längd minst 1000, så Ω är icketom. Språket $\emptyset$ är reguljärt och därmed TM-accepterbart, men innehåller ingen sträng av längd minst 1000. Så det finns något TM-accepterbart språk som ej tillhör Ω. Från Rices sats följer nu att ingen TM kan avgöra

för godtycklig TM M om $L(M) \in \Omega$, dvs. om $L(M)$ innehåller någon sträng av längd minst 1000.

8. L_4 är sammanhangsfri och produceras av följande CFG (där stora bokstäver är icketerminerande symboler):

$$S \to UV$$
$$U \to aUa \mid bUb \mid aaaa \mid bbbb \mid abba \mid baab$$
$$V \to aV \mid bV \mid \varepsilon$$

L_4 är <u>inte</u> regulärt och det följer från särskiljandesatsen om man visar att L_4 särskiljer den oändliga mängden

$$A = \{(ab)^n : n \in \mathbb{N}, n \geq 1\}$$

Låt $x, y \in A$ och $x \neq y$. Då finns $i, j \geq 1$ så att $i \neq j$, $x = (ab)^i$ och $y = (ab)^j$.

Vi kan anta att $i < j$ (om $i > j$ så resonerar vi på liknande sätt). Välj $z = (ba)^i$.

Vi får $xz = (ab)^i (ba)^i \in L_4$ för om $u = (ab)^i$ och $v = \varepsilon$ så $uu^{rev}v = (ab)^i (ba)^i$ och $|u| \geq 2$. Men det går <u>inte</u> att

skriva $(ab)^j(ba)^i$ på formen $uu^{rev}v$ där $|u| \geq 2$ så $yz \notin L_4$. (Detta beror på att om $uu^{rev}v = (ab)^j(ba)^i$ så måste u sluta på samma tecken som u^{rev} börjar med så vi får $u = (ab)^j$ och $u^{rev} = (ba)^j$ men då kan vi inte ha $uu^{rev}v = (ab)^j(ba)^i$ eftersom $j > i$.)

L_3 är inte sammanhangsfritt och därmed inte heller reguljärt. Vi använder pumpsatsen för CFL.

1. L_3 är oändligt för $a^{2n} \in L_3$ för alla $n \geq 2$.
2. Antag att L_3 är en CFL.
3. Låt K vara given av pumpsatsen.
4. Välj $w = a^K b^K b^4 a^K b^K$ så $w \in L_3$ och $|w| \geq K$.
5. Antag att $w = uvxyz$, $|vxy| \leq K$ och $vy \neq \varepsilon$. Då är vxy en delsträng av $a^K b^K$ eller av $b^K b^4 a^K$. Vi har $uxz = a^i b^j b^4 a^s b^t$ och ett av följande fall gäller:
   (a) $i < K$ eller $j < K$, och $s = t = K$.
   (b) $j < K$ eller $s < K$, och $i = t = K$.
   (c) $s < K$ eller $t < K$, och $i = j = K$.

Om $a^i b^j b^4 a^s b^t = \alpha\beta\beta^{rev}\alpha$ och $|\beta| \geq 2$ så måste $\alpha = a^i b^j$ och $\alpha = a^s b^t$, vilket är omöjligt i samtliga fall (a) – (c). Så $uxz \notin L_3$.

6. Punkt 5 motsäger pumpsatsen för CFL, så L_3 är inte en CFL.

---

10. Se resonemanget på sidorna 79-80 i kursboken. Notera också att grammatiken G i denna uppgift kan producera alla strängar som G_3 på sidan 79 kan producera, för (tex.) produktionen $S \Rightarrow^* aSbS$ kan fås med reglerna $S \to SS$ och $S \to aSb$.