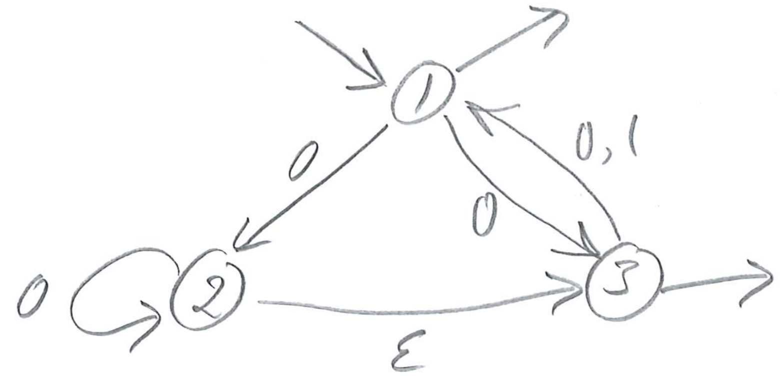
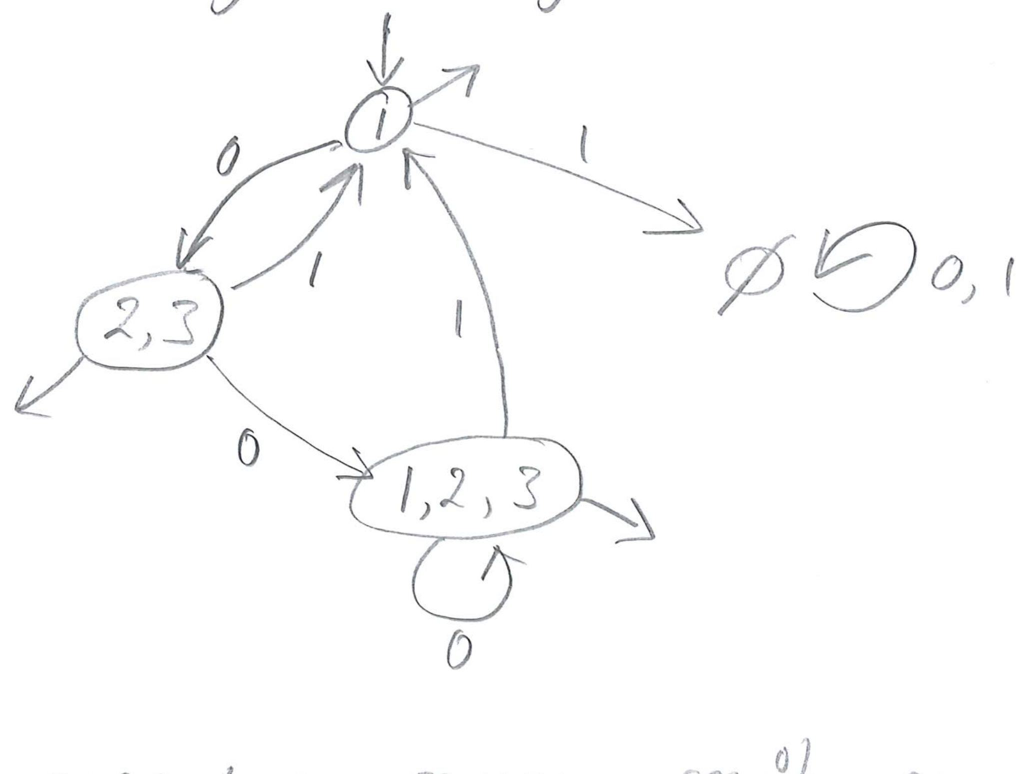
Automatateori, 2020-12-19

Lösningsförslag

1. NFA:n är redan icke-glupsk så delmängdsalgoritmen kan tillämpas direkt, efter att tillstånden har namngivits:

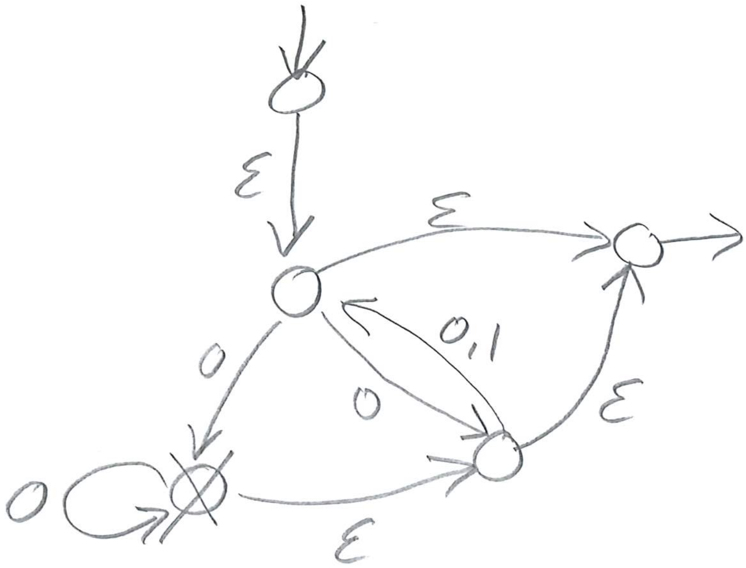


Tillståndsalgoritmen ger DFA:n

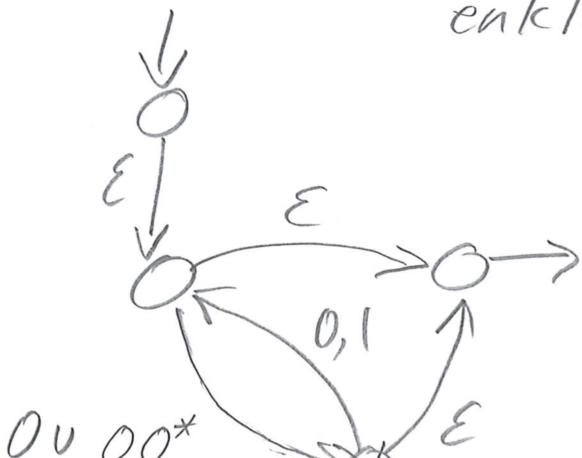


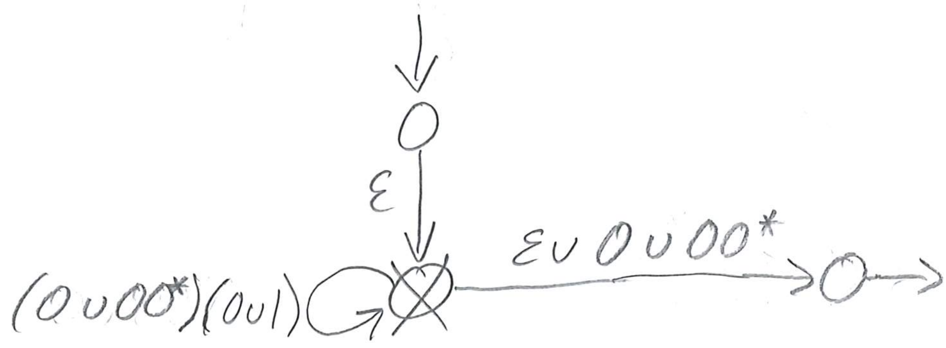
som accepterar samma språk som den givna NFA:n.

2. Nytt start- och accepterande tillstånd läggs till:



Sedan elimineras alla de gamla tillstånden, ett för ett, och vissa förenklingar gör jag på en gång:





$((0 \cup 00^*)(0 \cup 1))^*(\varepsilon \cup 0 \cup 00^*)$

(Detta kan förenklas till $(00^*(0 \cup 1))^*(\varepsilon \cup 00^*)$.)

Ovanstående reguljära uttryck beskriver språket som NFA:n accepterar.

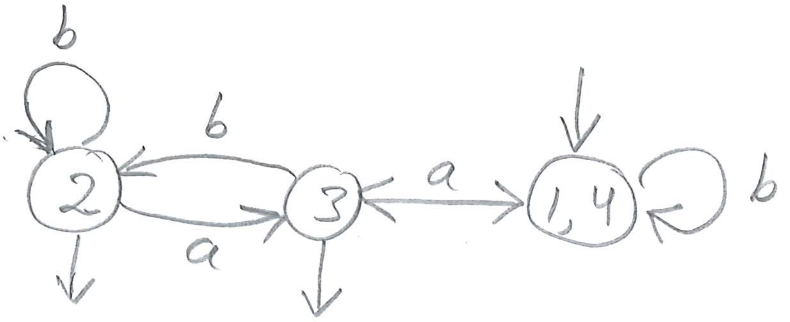
3. Jag kallar starttillståndet '1' och sedan numreras de övriga tillstånden som 2, 3, 4 från vänster till höger. Då får jag följande övergångstabell som gör det enklare att använda särskiljandealgoritmen:

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

Sedan särskiljandealgoritmen:

Nivå	Uppdelning			
1	{2, 3}		{1, 4}	(accepterande resp. icke-accept.)
2	{2}	{3}	{1, 4}	
3	{2}	{3}	{1, 4}	

När två nivåer ser likadana ut så terminerar algoritmen och tillstånd i samma del "identifieras (slås ihop)".



Denna DFA är minimal och accepterar samma språk som den ursprungliga DFA:n.

4. L_1 är reguljärt och detta kan inses på följande sätt. Antag att x innehåller exakt tre gånger så många 'a' som y gör. Om y innehåller n stycken 'a' så innehåller x 3n stycken 'a' och xy innehåller $3n + n = 4n$ stycken 'a'. Det är rakt på sak att visa att varje $w \in \{a,b\}^*$ så att antalet 'a' i w är jämnt delbart med 4 tillhör L_1. Alltså gäller att

$$L_1 = \{w \in \{a,b\}^* : \text{antalet 'a' i } w \text{ är jämnt delbart med } 4\}.$$

Det följer att

$$(b^*ab^*ab^*ab^*ab^*)^*$$

är ett reguljärt uttryck för L_1.

L_2 är inte reguljärt och jag bevisar det med pumpsatsen.

1. L_2 är oändligt för $a^{3n}b^n \in L_2$ för alla $n = 0, 1, 2, 3, \ldots$

2. Antag att L_2 är reguljärt.

3. Låt N vara givet av pumpsatsen för L_2.

4. Välj (t.ex.) $u = a^{3N}$, $w = b^N$, $v = \varepsilon$ så $|w| \geq N$ och $uwv = a^{3N}b^N \in L_2$.

5. Antag att $w = xyz$ där $y \neq \varepsilon$. Då gäller att $xy^2z = b^m$ där $m > N$ så $uxy^2zv = a^{3N}b^m \notin L_2$.

6. Punkt 5 motsäger pumpsatsen för reguljära språk, så L_2 kan inte vara reguljärt.

(Man kan också använda särskiljandesatsen och visa att (t.ex.) mängden $\{a^{3n} : n \in \mathbb{N}\}$ särskiljs av L_2.)

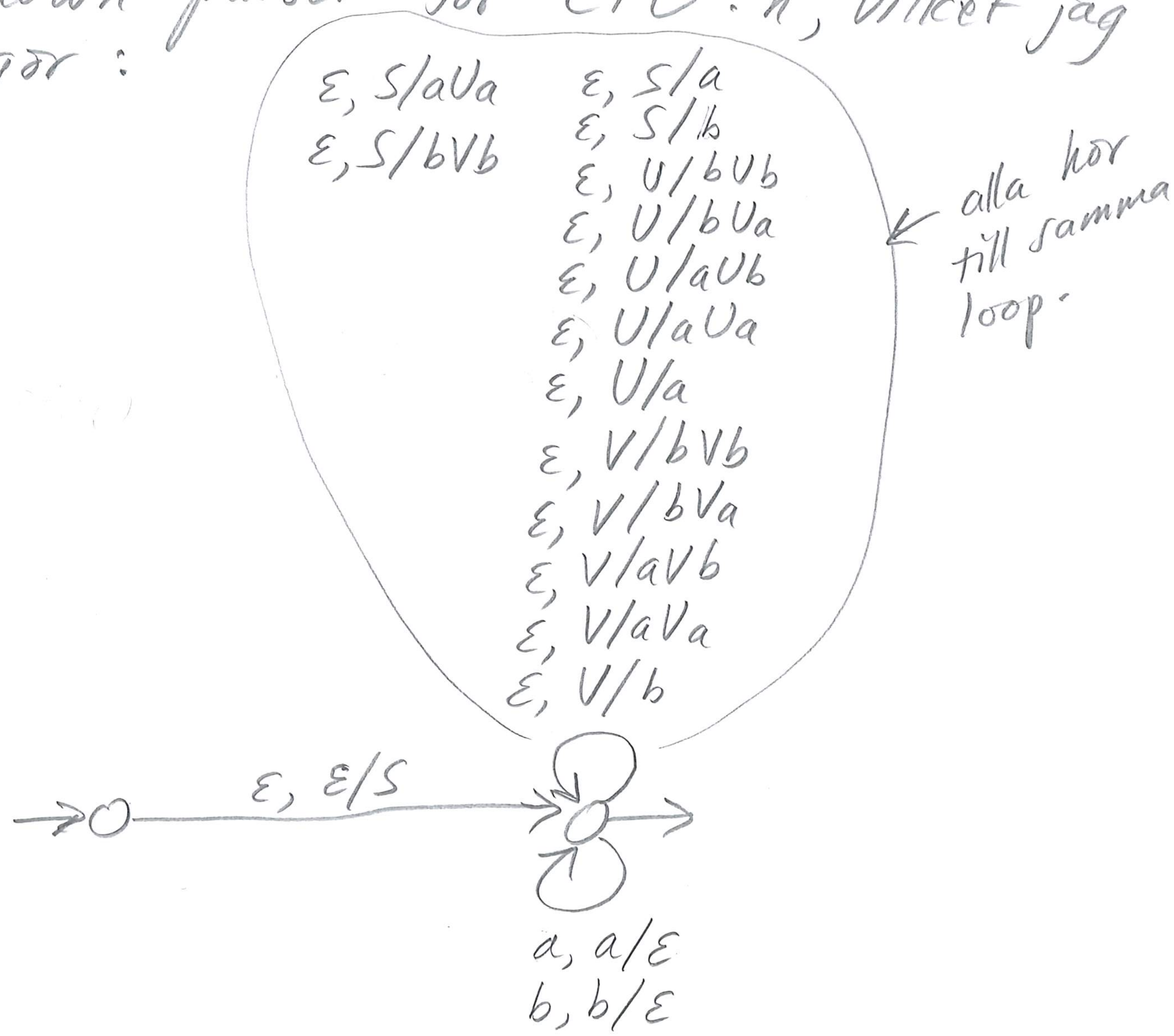
5. En CFG för språket:

$S \to aUa \mid bVb \mid a \mid b$

$U \to aUa \mid aUb \mid bUa \mid bUb \mid a$

$V \to aVa \mid aVb \mid bVa \mid bVb \mid b$

En PDA kan konstrueras direkt, men man kan också göra en (t.ex.) top-down parser för CFG:n, vilket jag gör:



6. (a) Körning:

#ababa

ababa#

ababa

ababa

abaa# (S_L kördes)

abaa##

abaa#b

abaa#b

⋮

abaa#b

aaa##b (S_L kördes)

aaa##b

aaa#bb

aaa#bb

⋮

#aaa#bb (går nu ut ur huvudslingan)

⋮

aaa#bb

aaabb#

⋮

#aaabb

terminerar.

(b) Vid start på #w (där $w \in \{a,b\}^*$) så stannar TM:en i konfigurationen $\#a^n b^m$ där n är antalet 'a' i w och m är antalet 'b' i w.

Så man kan säga att M beräknar funktionen som ordnar alla tecken i inputsträngen i bokstavsordning.

7. L_3 är reguljärt (till och med ändligt) med följande reguljära uttryck:

$a(a \cup b \cup c)a \cup b(a \cup b \cup c)b$.

Därmed är L_3 även sammanhangsfritt.

L_4 är inte sammanhangsfritt och därmed inte reguljärt. Bevis med pumpsatsen för sammanhangsfria språk:

1. L_4 är oändligt för $a^n b^n c a^n b^n \in L_4$ för alla $n = 1, 2, 3, \ldots$
2. Antag att L_4 är en CFL.
3. Låt K vara givet av pumpsatsen.
4. Välj (t.ex.) $w = a^{K+1} b^{K+1} c a^{K+1} b^{K+1}$ så $|w| \geq K$ och $w \in L_4$.
5. Antag att $w = uvxyz$, $|vxy| \leq K$ och $vy \neq \varepsilon$. Vi har två fall:

   a) vxy är en delsträng av $a^{K+1} b^{K+1}$. Då kommer uv^0xy^0z ha formen $a^n b^m c a^{K+1} b^{K+1}$ eller $a^{K+1} b^{K+1} c a^n b^m$ där $n < K+1$ eller

$m < K+1$. Oavsett vilket som är fallet så kan <u>inte</u> uv^0xy^0z ha formen sts där $s \in \{a,b\}^+$ och $t \in \{a,b,c\}$, så $uv^0xy^0z \notin L_4$.

b) vxy är en delsträng av $b^{K+1}ca^{K+1}$. Då kommer uv^0xy^0z ha formen

$$a^{K+1}b^n c^m a^l b^{K+1}$$

där $n < K+1$ eller $m = 0$ eller $l < K+1$. I samtliga fall så kan <u>inte</u> uv^0xy^0z ha formen sts där $s \in \{a,b\}^+$ och $t \in \{a,b,c\}$, så $uv^0xy^0z \notin L_4$.

6. Punkt 5 motsäger pumpsatsen för CFL så L_4 kan inte vara en CFL.

Språket L_5 är <u>inte</u> sammanhangsfritt och därmed <u>inte</u> reguljärt. Detta kan bevisas på samma sätt som för L_4. För om $w = a^{K+1}b^{K+1}ca^{K+1}b^{K+1}$ i steg 4 så $|w| \geq K$ och $w \in L_5$. I steg 5 får vi $uv^0xy^0z \notin L_5$ i båda fallen.

8. Både problemet i (a) och i (b) är oavgörbart. Vi visar detta med motsägelsebevis kombinerat med Rices sats.

(a) Antag att en TM M kan avgöra, för godtyckliga TM:ar M_1 och M_2 om M_1 och M_2 accepterar samma strängar av längd högst 1000.

Fixera nu M_1 till att vara en TM sådan att $L(M_1) = \emptyset$ (så M_1 accepterar ingen sträng).

Då kan M avgöra, givet en godtycklig TM M_2, om $L(M_2)$ innehåller någon sträng av längd högst 1000.

Men detta problem är oavgörbart, enligt Rices sats. För låt

$\Omega = \{ L : L$ är TM-accepterbart och innehåller någon sträng av längd $\leq 1000 \}$.

Uppenbarligen är alla språk i Ω TM-accepterbara. Dessutom är Ω icketom, för tex. språket som beskrivs av a^* tillhör Ω. Och det finns TM-accepterbara språk som inte tillhör Ω, som tex. språket $\{a^{1001}\}$.

Det följer av Rices sats att ingen TM kan avgöra, för en godtycklig TM M_2, om $L(M_2) \in \Omega$.

(b) Antag att en TM M kan avgöra, för godtyckliga TM:ar M_1 och M_2, om $L(M_1) \subseteq L(M_2)$. Fixera en TM M_1 sådan att $L(M_1) = \{\varepsilon\}$. (Så M_1 ska bara acceptera den tomma strängen.)

Då kan M avgöra, för en godtycklig TM M_2, om $\varepsilon \in L(M_2)$. Men detta problem är oavgörbart, vilket kan visas med Rices sats. Låt $\Omega =$
$\{L : L$ är TM-accepterbart och $\varepsilon \in L\}$.

Att avsluta beviset överlåts åt läsaren.

9. (a) Påståendet är <u>falskt</u>. Motexempel: $L_1 = \emptyset$ och $L_2 = \{a^n b^n : n \in \mathbb{N}\}$. Då är L_1 reguljärt (med reg. uttryck '$\emptyset$'), L_2 är ett icke-reguljärt språk och $L_1 \subseteq L_2$.

(b) Påståendet är <u>falskt</u>. Motexempel: $L_1 = \{a^n b^n : n \in \mathbb{N}\}$. Då är L_1 icke-reguljärt och $L_1 \subseteq L_2$ där L_2 beskrivs av a^*b^*.

(c) Påståendet är <u>falskt</u>. Motexempel: $L_1 = \{a^n b^n : n \in \mathbb{N}\}$, $L_2 =$ språket som beskrivs av a^*b^* och $L_3 = L_2$. Eftersom $L_1 \subseteq L_3$ så $L_3 \cup L_1 = L_3 = L_2$ men L_1 är <u>inte</u> reguljärt trots att L_2 och L_3 är reguljära.