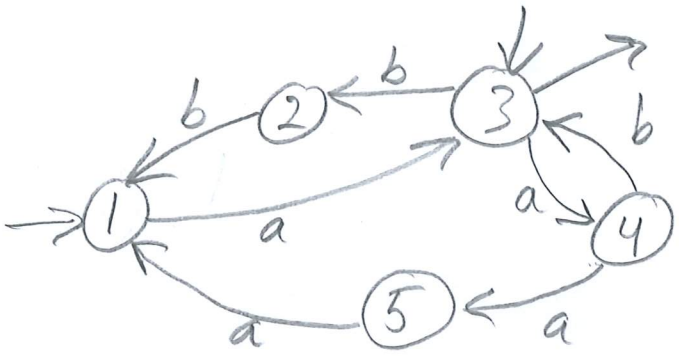
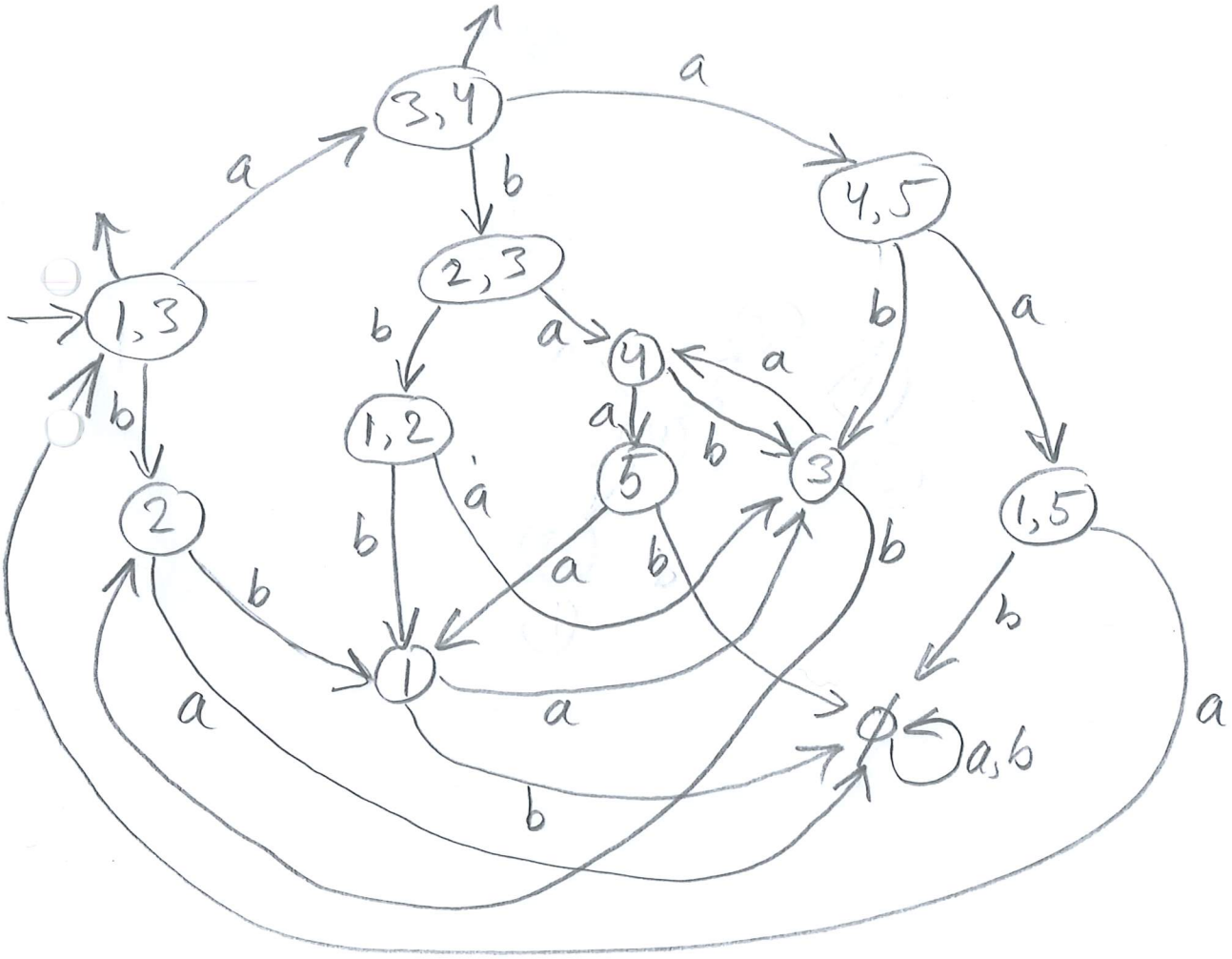
Automatateori 2021-08-20
Lösningsförslag

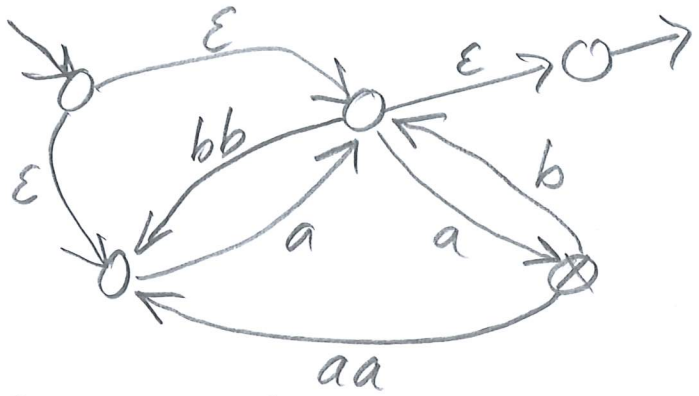
1. Först skapas en icke-glupsk NFA med samma språk och tillstånden namnges:



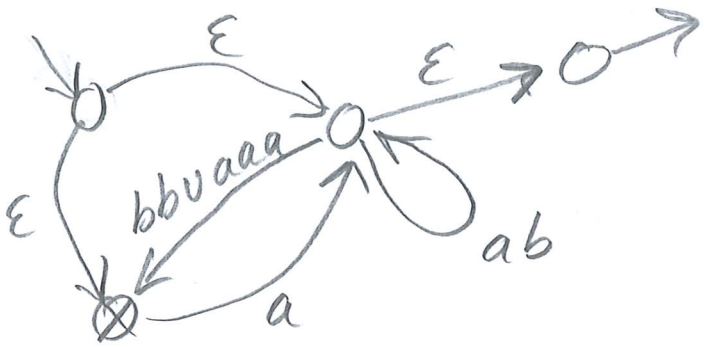
Sedan används delmängdskonstruktionen:

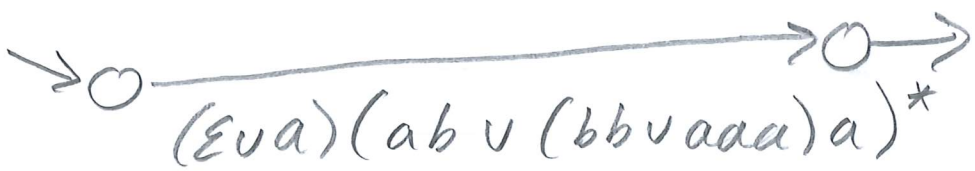


2. Nytt läggs nytt start- och accepterande tillstånd till:



Sedan elimineras de gamla tillstånden, ett efter ett och förenklingar görs:





De sista uttrycket beskriver NFA:ns språk.

3. Om tillstånden numreras medurs med början i starttillståndet får man följande tillståndsövergångstabell:

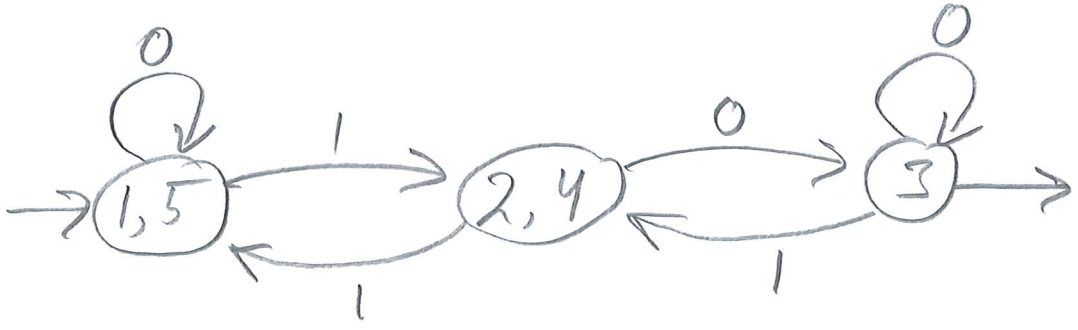
	1	2	3	4	5
0	5	3	3	3	5
1	2	5	2	5	4

Sedan används särskiljandealgoritmen:

nivå	uppdelning	uppdelning: accept. resp. icke-accept. tillstånd.
1	{3} {1, 2, 4, 5}	
2	{3} {1, 5} {2, 4}	
3	{3} {1, 5} {2, 4}	

Algoritmen terminerar när de två sista nivåerna är likadana.

Den minimala DFA:n blir:



4. De två språken kan också beskrivas så här

$L_1 = \{ w \in \{0,1\}^* : w = 0$ eller så börjar w med 1 och slutar med $0 \}$.

$L_2 = \{ 1^n 0^{n+1} : n = 2, 3, 4, \ldots \}$

L_1 är reguljärt med följande reguljära uttryck: $0 \cup 1(0 \cup 1)^* 0$.

L_2 är inte reguljärt.
Bevis med särskiljandesatsen.
Låt $A = \{ 1^n : n = 2, 3, 4, \ldots \}$ så A är oändlig. Räcker nu att visa att L_2 särskiljer A. Låt $x, y \in A$ vara olika. Så det finns $i \neq j$ så att $x = 1^i$ och $y = 1^j$. Låt $z = 0^{i+1}$. Då gäller att $xz = 1^i 0^{i+1} \in L_2$ men $1^i 0^{j+1} \notin L_2$ eftersom $j \neq i$.
Så L_2 särskiljer A.

Om man använder pumpsatsen kan man välja (tex) $u = 1^N$, $w = 0^{N+1}$, $v = \varepsilon$, där N ges av pumpsatsen.

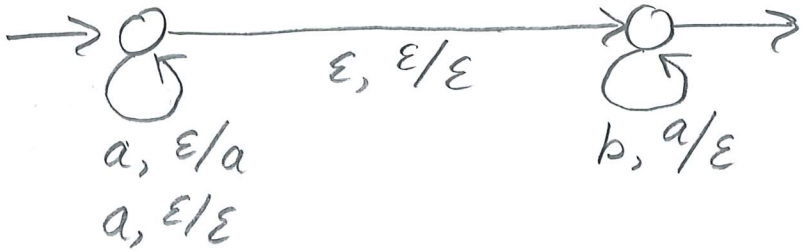
5 (a) $S \Rightarrow ASb \Rightarrow AASbb \Rightarrow$ (5)

$AAASbbb \Rightarrow AAAAbbb \Rightarrow$
$aAAAbbb \Rightarrow aaAAbbb \Rightarrow aaaAbbb$
$\Rightarrow aaaabbb$

Strängen aabab kan inte produceras för bara regeln $S \to ASb$ skapar nya A och b. Övriga regler kan bara skapa fler A intill tidigare A:n och göra 'a' från 'A' (samt ta bort 'S'). Sålunda kommer alla 'a' att ligga till vänster om alla 'b' vilket inte är fallet i aabab.

(b) Språket som produceras är
$\{ a^m b^n : m \geq n \geq 0 \}$.

(c) Man kan utgå från grammatiken (som är sammanhangsfri) och konstruera en top-down eller bottom-up parser. Eller så kan en PDA för språket konstrueras direkt, t.ex. så här:



6 (a) Strängen aababb accepteras och här visas det med en körning:

(6)

#aababb	aa#bb	ab#
aabab#b	aabb#	ab
aababb	aabb	ab
aabab#b	aabb	#b
aabab#b	aabb	b#
aab#bb	a#bb	b
aabbb#	abb#	#
#aabbb	#abb	#
aabbb	abb	stannar
	a#b	

(b) Språket som accepteras består av alla strängar över $\{a, b\}$ som innehåller minst lika många 'b' som 'a'. Om inputsträngen innehåller fler 'a' än 'b' så kommer, efter ett antal loopar, sökningen efter 'b' som görs av R_b inte att terminera eftersom inga 'b' är kvar.

7. Om $w \in L_4$ så måste första och sista tecknet i w vara detsamma och det näst första och näst sista tecknet måste vara detsamma.
Därför kan L_4 beskrivas så här:

$$L_4 = \{u^{rev} v u : u, v \in \{a,b\}^*, |u| = 2, |v| \geq 2\}.$$

och man ser att L_4 är reguljärt med reguljärt uttryck

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

Därmed är L_4 också sammanhangsfritt.
L_3 är inte sammanhangsfritt och därmed inte heller reguljärt.
Vi visar detta med pumpsatsen för sammanhangsfria språk.

1. Notera att L_1 är oändligt, för ex. $a^{3n} \in L_1$ för alla $n = 2, 3, 4, \ldots$

2. Antag att L_1 är sammanhangsfritt.

3. Låt k vara given av pumpsatsen för sammanhangsfria språk.

4. Låt $w = a^k b^k aaa^k b^k$, så $w \in L_1$ och $|w| \geq k$.

5. Antag att $w = uvxyz$ där $|vxy| \leq k$ och $vy \neq \varepsilon$. En fallanalys kommer visa att oavsett hur uppdelningen av w ser ut så finns n så att $uv^n xy^n z \notin L_1$.

Notera att eftersom $|vxy| \leq k$ så kan inte vy innehålla a:n från båda "a-blocken" och vy kan inte innehålla b:n från båda "b-blocken".

Om vy innehåller a:n från det första a-blocket så kommer $uv^2xy^2z \notin L_1$, för strängen har för få 'a' i det andra a-blocket.

Om vy innehåller a:n från det andra a-blocket (aaa^N) så $uxz \notin L_1$, för strängen har för få 'a' i det andra a-blocket.

Om vy innehåller b:n från det första b-blocket så kommer $uxy \notin L_3$ för strängen uxy har för få b:n i det första b-blocket.

Om vy innehåller b:n från det andra b-blocket så kommer $uv^2xy^2z \notin L_3$ för strängen har för få b:n i det första b-blocket.

6. Punkt 5 motsäger pumpsatsen för sammanhangsfria språk så L_3 är inte sammanhangsfritt.

8. (a) L_5 är inte TM-avgörbart och detta kan visas med Rices sats.

Låt $\Omega = \{L : L = L_1 \cup L_2$ där L_1 och L_2 är sammanhangsfria$\}$.

Eftersom unionen av två sammanhangsfria språk är sammanhangsfritt och varje sammanhangsfritt språk är TM-accepterbart så följer att

alla språk i Ω är TM-accepterbara. Ω är icke-tom för $\emptyset \cup \emptyset = \emptyset$ och $\emptyset$ är sammanhangsfritt så $\emptyset \in \Omega$.

Språket L_{stopp} är TM-accepterbart men inte TM-avgörbart och därmed inte sammanhangsfritt. Eftersom unionen av två sammanhangsfria språk är sammanhangsfritt så kan L_{stopp} inte tillhöra Ω, så Ω innehåller alltså inte alla TM-accepterbara språk.

Nu följer av Rices sats att ingen TM existerar som givet K_M avgör om $L(M) \in \Omega$ (dvs. om $L(M)$ är unionen av två sammanhangsfria språk), så L_5 är inte TM-avgörbart.

(b) L_6 är TM-avgörbart.
Här är en informell algoritm som avgör problemet och enligt Church-Turings tes kan algoritmen "översättas" till en TM som avgör språket.

Algoritm:
Givet $K_M \# K_w$, beräkna antalet konfigurationer $(p, u\sigma v)$ som M kan ha där $|u\sigma v| \leq 2|w|$. Kalla detta antal för $n_{M,w}$. Kör nu M på w i $n_{M,w}+1$ steg eller tills M stannar om den gör det innan $n_{M,w}+1$ steg. Om M har besökt mer än $2|w|$ rutor så svara 'nej'. Annars måste M ha befunnit sig i samma konfiguration minst två gånger och eftersom M är deterministisk så kommer M aldrig att besöka mer än $2|w|$ rutor, oavsett hur många steg som körs och oavsett om M någonsin stannar.
Så svaret 'ja' ska ges i detta fall.

Låt $w = abaabbbabaaaab$.

9. Antalet kombinationer av tillstånd och tecken som finns för M_1 är $5 \cdot 2 = 10$. Eftersom $|w| = 13 > 10$ och $w \in L(M_1)$ så måste samma kombination av tillstånd och tecken som avläses ha uppstått minst två gånger under avläsningen av w som leder till acceptans. Då kan w delas upp som $w = xyz$ där y är den sträng som avläses mellan den första och andra uppkomsten av denna kombination. Eftersom M_1 är deterministisk så kommer alla strängar $xyz, xy^2z, xy^3z, \ldots$ accepteras av M_1 så den accepterar oändligt många strängar. Alltså var informationen tillräcklig för att besvara frågan om M_1 accepterar oändligt många strängar och svaret på frågan är 'ja'.

(12)

Informationen som ges är däremot inte tillräcklig för att besvara frågan om M_2 accepterar oändligt många strängar, för det finns en DFA med 15 tillstånd som accepterar w och oändligt många strängar och det finns en DFA med 15 tillstånd som bara accepterar w. Ge exempel själv.