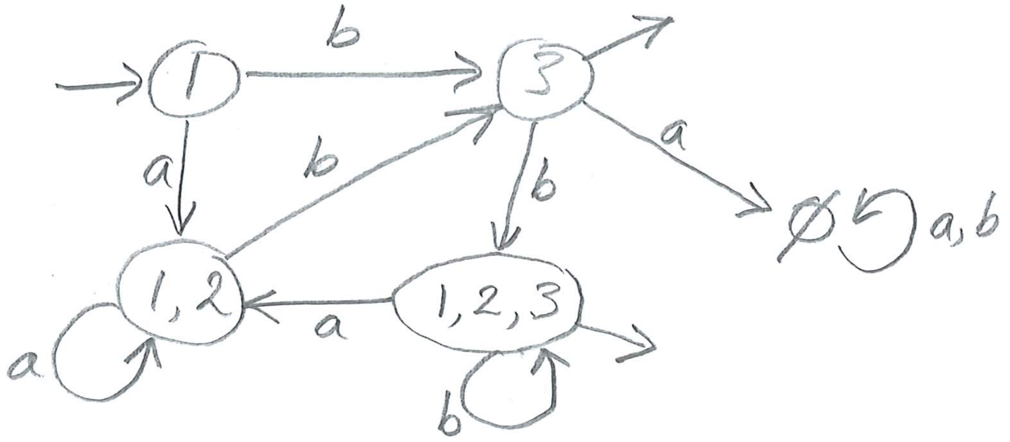
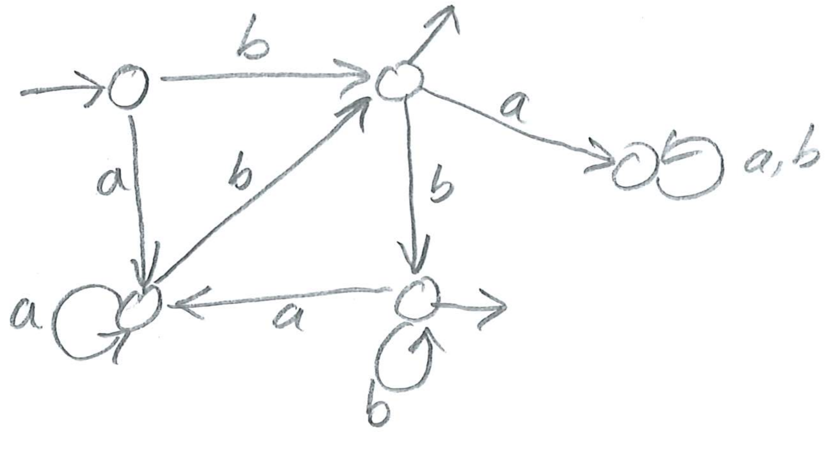
Tenta 2019-12-19, svar/lösningsförslag

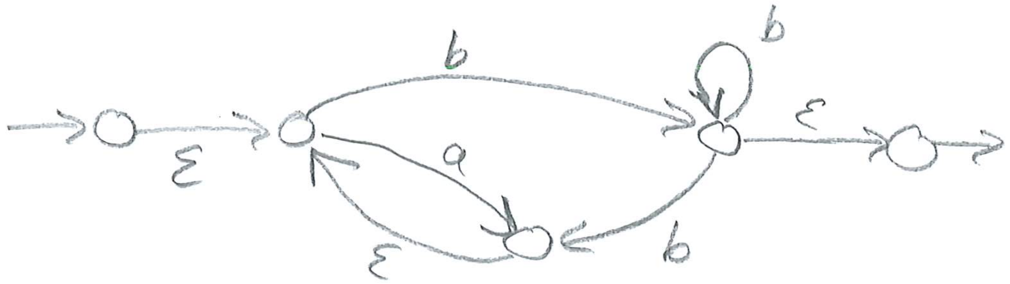
1. NFA:n är redan icke-glupsk så delmängdsalgoritmen kan användas direkt. Om tillstånden numreras som 1, 2, 3 i moturs riktning med början i starttillståndet så får man följande:



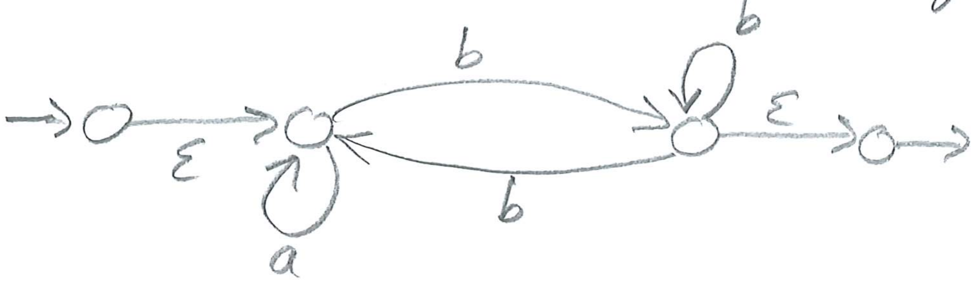
Tillsnyggat:

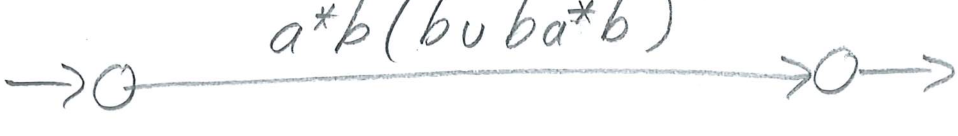


2. Nytt start- och accepterande tillstånd läggs till.



Sedan elimineras de gamla tillstånden, ett för ett, och vissa förenklingar görs på en gång.





Det sista uttrycket är ett reguljärt uttryck för den ursprungliga NFA:ns språk.

3. Om tillstånden numreras i medurs riktning och starttillstånder numreras som 1, så får vi följande övergångstabell:

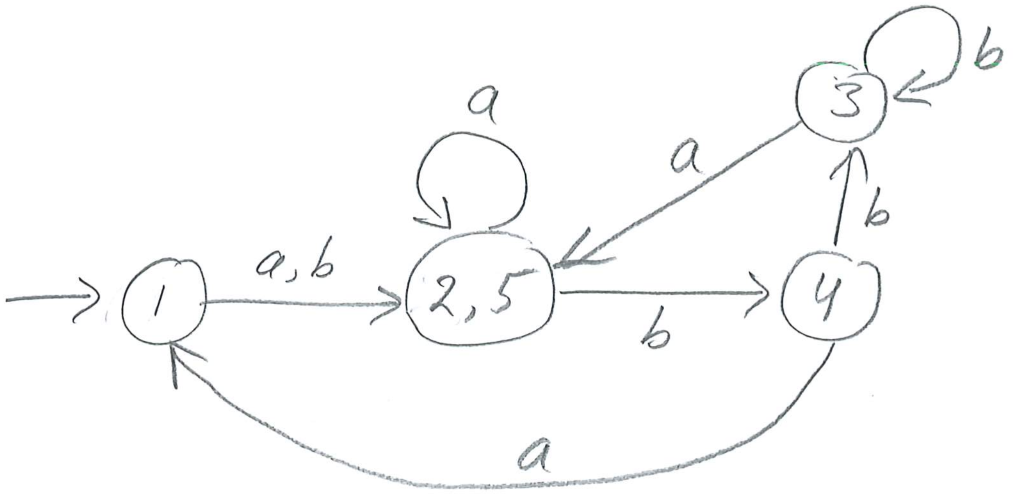
	1	2	3	4	5
a	2	2	2	1	2
b	5	4	3	3	4

Nu använder vi särskiljandealgoritmen för att sönderdela tillståndsmängden.

nivå	sönderdelningar	
1	{1} {2, 3, 4, 5}	accepterande och icke-accepterande tillst. särskiljs.
2	{1} {2, 3, 5} {4}	a driver DFA:n från 4 till 1, men b driver DFA:n från 2, 3, resp. 5, till 2.
3	{1} {2, 5} {3} {4}	b driver DFA:n från 4 till 3, men b driver DFA:n från 2 resp. 5 till 4.
4	{1} {2, 5} {3} {4}	

Eftersom ingen ny sönderdelning är möjlig så är vi klara med särskiljningen.

Enligt särskiljandealgoritmen kan tillstånden 2 och 5 "slås ihop" och vi får den minimala DFA:n:



4. L_1 är reguljärt eftersom ett reguljärt uttryck för L_1 är $a(bb)^*ab^*$.

<u>Alternativt</u>: Följande NFA accepterar L_1:

L_2 är inte reguljärt.

<u>Bevis med särskiljandesatsen</u>:

Låt $A = \{ab^{2n}a : n \in \mathbb{N}\}$ så A är oändlig.

Vi visar att A särskiljs av L_2.
Då följer att L_2 inte är reguljär.

Låt $x, y \in A$ vara olika, så $x = ab^{2^i}a$ och $y = ab^{2^j}a$ för något val av $i \neq j$.
Om $z = b^i$ så $xz = ab^{2^i}ab^i \in L_2$ men $yz = ab^{2^j}ab^i \notin L_2$.
Eftersom x och y var godtyckliga olika strängar från A, så särskiljs A av L_2.

Bevis med pumpsatsen:

Notera att L_2 är oändligt. Antag att L_2 är reguljärt och låt N vara givet av pumpsatsen. Välj t.ex. $u = ab^{2N}a$, $w = b^N$ och $v = \varepsilon$, så $uwv = ab^{2N}ab^N \in L_2$.
Antag att $w = xyz$ och $y \neq \varepsilon$. Eftersom xyz bara innehåller b:n så följer att $xy^2z = b^k$ för något $k > N$ och därmed så $uxy^2zv = ab^{2N}ab^k \notin L_2$ (det sista b-blocket har mer än hälften så många b:n som det första b-blocket).
Eftersom detta inträffar för varje

uppdelning $w = xyz$ där $y \neq \varepsilon$ så har vi en motsägelse till pumpsatsen för reguljära språk.

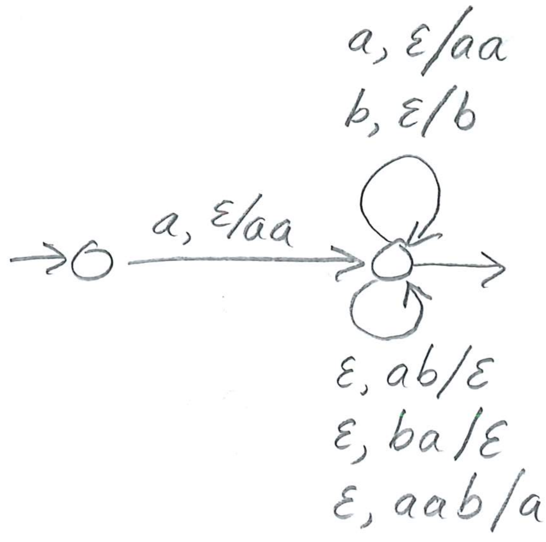
5 (a) Strängen abbabbabb tillhör $L(G)$ och här är en produktion:

$\underline{S} \Rightarrow \underline{S}ABB \Rightarrow \underline{S}ABBABB \Rightarrow$
$TBBABBABB \Rightarrow aBBABBABB \Rightarrow^*$
$abbabbabb$.

Strängen baababbbb tillhör <u>inte</u> $L(G)$ för reglerna $S \to TBB$ och $T \to a$ medför att varje sträng i $L(G)$ börjar på 'a'.

(b) Genom att analysera reglerna så ser man att $L(G) = \{w \in \{a,b\}^* :$ w börjar på 'a' och har dubbelt så många 'b' som 'a'$\}$.

Här är en PDA som accepterar $L(G)$:



För varje 'a' som avläses läggs <u>två</u> 'a' på stacken.

Med dessa övergångar kan stacken tömmas om (och endast om) inputsträngen har dubbelt så många 'b' som 'a'.

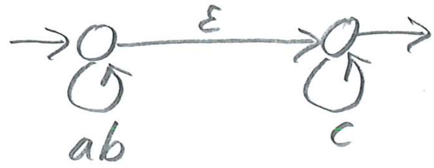
6. (a)

```
#11#1      11##       11#11      ##1111
11#1       11#1       #1#11      ⋮
111#       11#1#      ⋮          1111#
#111       11#11      1#11#      ⋮
111        11#11      ⋮          111111
#11        #11#11     1#1111     ⋮
11#                   ⋮          ##111111
                      #1#1111    #111111
                      1#1111     1111111
                                 #1111111
                                 terminerar
```

(b) $f(x,y) = 2(x+y)+1$

(Först omvandlas $1^x\#1^y$ till 1^{x+y} och sedan skrivs två ettor på slutet, för varje etta i 1^{x+y} som raderas för varje loop som körs, och till sist läggs en etta till.)

7. L_5 är reguljärt och har $(ab)^*c^*$ som reguljärt uttryck. Följande NFA accepterar L_5:



.

L_3 är sammanhangsfritt men ej reguljärt. Notera att $L_3 = \{a^n b^{n+m} c^m : n, m \in \mathbb{N}\} =$
$\{a^n b^n b^m c^m : n, m \in \mathbb{N}\}$.

Det följer att följande CFG producerar L_3:

$$S \to XY$$
$$X \to aXb \mid \varepsilon$$
$$Y \to bYc \mid \varepsilon.$$

Man kan visa (jag gör inte detaljerna) att (tex.) den oändliga mängden $A = \{a^n : n \in \mathbb{N}\}$ särskiljs av L_3 och då följer från särskiljandesatsen att L_3 inte är reguljärt.

Man kan också använda pumpsatsen för sammanhangsfria språk för att visa detta. (Om N är givet av pumpsatsen så kan man tex. välja $u = \varepsilon$, $w = a^N$, $v = b^N$, så $uwv = a^N b^N = a^N b^N b^0 a^0 \in L_3$, och sedan pumpa upp w.)

L_4 är inte sammanhangsfritt och därmed inte heller reguljärt.

Bevis med pumpsatsen för sammanhangsfria språk (CFL):

1. Vi observerar att L_4 är oändligt, för $a^n b^{n+2} c^{n+4} \in L_4$ för alla $n \in \mathbb{N}$.

2. Antag att L_4 är en CFL.

3. Låt då K vara givet av pumpsatsen för CFL.

4. Välj $w = a^K b^{K+2} c^{K+4}$, så $w \in L_4$ och $|w| \geq K$.

5. Antag att $w = uvxyz$, $|vxy| \leq K$ och $vy \neq \varepsilon$. Då är vxy en delsträng av $a^K b^{K+2}$ eller av $b^{K+2} c^{K+4}$.

Fall 1. Antag att vxy är en delsträng av $a^K b^{K+2}$. Då kommer uv^2xy^2z att ha fler än K a:n eller fler än $K+2$ b:n. I båda fallen har uv^2xy^2z exakt $K+4$ c:n och därför kan uv^2xy^2z inte ha formen $a^i b^j c^k$ där $i+2 \leq j$ och $j+2 \leq k$. Så $uv^2xy^2z \notin L_4$.

Fall 2. Antag att vxy är en delsträng av $b^{K+2}c^{K+4}$. Då kommer uxz ha färre än $K+2$ b:n eller färre än $K+4$ c:n. I båda fallen har uxz exakt K a:n och därför kan uxz inte ha formen $a^i b^j c^k$ där $i+2 \le j$ och $j+2 \le k$. Så $uxz \notin L_4$.

Vi har visat att oavsett fall så finns $n \in \mathbb{N}$ så att $uv^n xy^n z \notin L_4$.

6. Vår slutsats i punkt 5 motsäger pumpsatsen för CFL och därför kan L_4 inte vara en CFL.

---

8. (a) L är inte TM-avgörbar.
Bevis med Rices sats:
Låt $\Omega = \{L :$ L är TM-accepterbar och innehåller någon sträng av längd 5 eller 12$\}$.

Vi verifierar tre saker:

- Alla språk i Ω är TM-accepterbara. Detta följer från definitionen.
- Ω är inte tom. Språket $\{a^5\}$ är reguljärt (med reg. uttr. aaaaa) och därmed TM-accepterbart, så $\{a^5\} \in \Omega$.
- Något TM-accepterbart språk tillhör <u>inte</u> Ω. Språket $\emptyset$ är reguljärt och därmed TM-accepterbart och $\emptyset$ tillhör inte Ω.

Från Rices sats följer nu att ingen TM kan avgöra, för godtycklig TM M, om $L(M) \in \Omega$ eller inte.
Detta innebär (med annan formulering) att L <u>inte</u> är TM-avgörbar.

(b) L är TM-accepterbar och här är en informell algoritm som terminerar på input K_M om och endast om $K_M \in L$:

Givet koden K_M till en TM M, generera alla (ändligt många) strängar över M:s inputalfabet som har längd 5 eller 12. Kör sedan M "parallellt" på alla dessa strängar. (Med detta menar jag att vi har ett antal kopior av M och varje sträng körs på "sin" kopia.)

Om M stannar på någon av dessa strängar så terminerar algoritmen och $K_M \in L$.

Om M inte stannar på någon av strängarna så terminerar inte algoritmen och M accepterar ingen sträng av längd 5 eller 12, så $K_M \notin L$.

(c) En sats från kursen säger att om L och $\bar{L}$ är TM-accepterbara så är L TM-avgörbar. Från (a) och (b) följer att $\bar{L}$ <u>inte</u> är TM-accepterbar.

9. (a) Påståendet stämmer.
Antag att G_1 och G_2 är CFG:er för L_1 och L_2, respektive.
Vi kan anta att S_1 och S_2 är startsymbolerna för G_1 och G_2, respektive, och att varje icke-terminerande symbol i G_1 är olik varje icke-terminerande symbol i G_2.

Låt G vara CFG:n som innehåller alla G_1:s regler och alla G_2:s regler samt regeln

$$S \to S_1 S_2$$

där S är en ny startsymbol.
Då gäller att $L(G) = L_1 L_2$ så $L_1 L_2$ är sammanhangsfritt.

(b) Påståendet stämmer <u>inte</u>.
Genom att resonera som i (a) men ersätta $S \to S_1 S_2$ med $S \to S_1$ och $S \to S_2$ så följer att om L_1 och L_2 är sammanhangsfria så är även $L_1 \cup L_2$ sammanhangsfritt.

Låt $L_1 = \{a^n b^n c^k : n, k \in \mathbb{N}\}$ och $L_2 = \{a^n b^k c^k : n, k \in \mathbb{N}\}$.

Då är L_1 och L_2 CFL:er vilket lämnas som övning att visa.

Men $L_1 \cap L_2 = \{a^n b^n c^n : n \in \mathbb{N}\}$ är <u>inte</u> en CFL vilket kan visas med liknande resonemang som för L_4 i uppgift 7. Eftersom $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ så gäller för minst ett av språken L_1, L_2 eller $\overline{L_1} \cup \overline{L_2}$ att det är en CFL men dess komplement är <u>inte</u> en CFL.