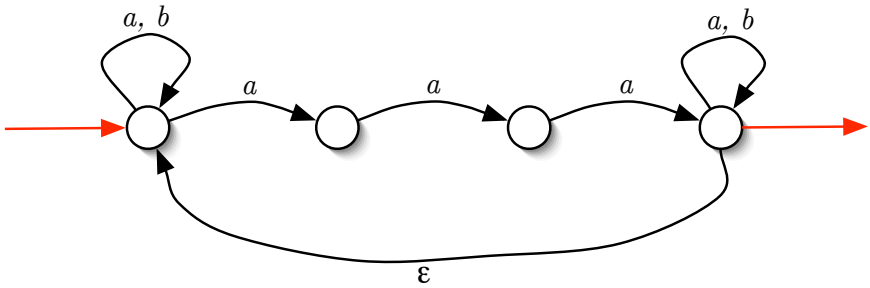
UPPSALA UNIVERSITET
Matematiska institutionen
Salling (070–6527523)

PROV I MATEMATIK
AUTOMATATEORI DV1
21 mars 2001

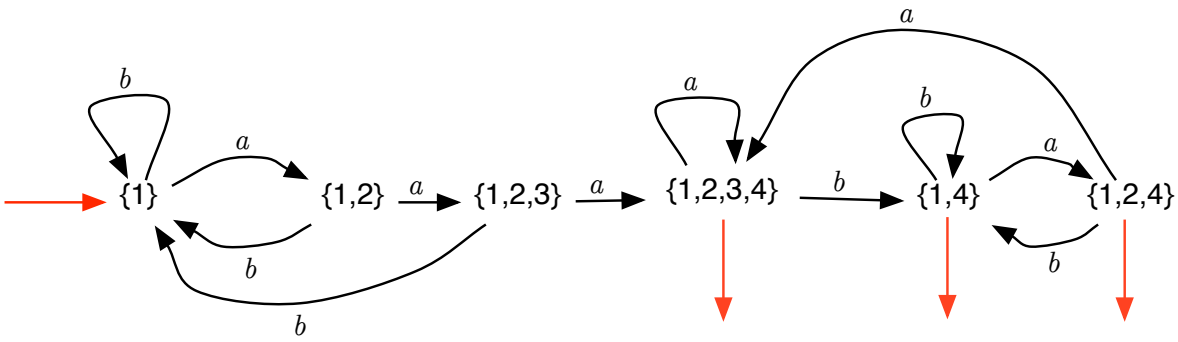
1. Betrakta nedanstående NFA.



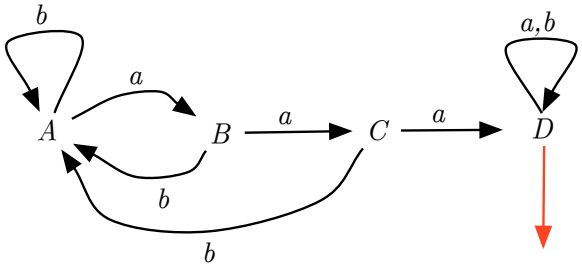
a) Konstruera med hjälp av delmängdskonstruktionen en DFA ekvivalent med NFA:n. Beskriv också informellt vilka strängar som accepteras. (3p)
b) Är den resulterande DFA:n minimal? Motivera! (2p)

LÖSNING

a) Strängarna ifråga är de (över alfabetet $\{a, b\}$) som innehåller *aaa*. Här är den DFA som delmängdskonstruktionen ger upphov till.



b) DFA:n är **inte** minimal. De tre accepterande tillstånden kan slås ihop!



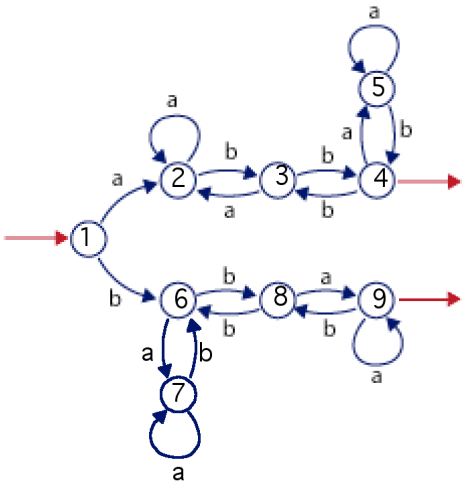
---

2. Språket L över alfabetet $\{a, b\}$ definieras av att dess strängar w uppfyller följande kriterium: $|w| > 2$, w inleds och avslutas med olika tecken och innehåller ett udda antal *bb*-förekomster. Observera att vi räknar *samtliga bb*-förekomster - även sådana som inte är åtskilda! T ex är *bbabbba* en sträng i L, men inte *bbba*.

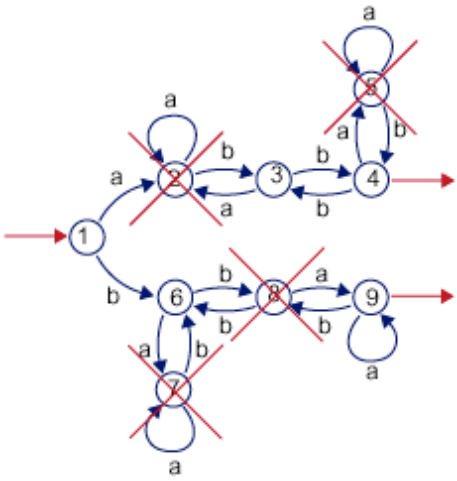
a) Konstruera en DFA för L. (3p) b) Konstruera ett reguljärt uttryck för L. (3p)

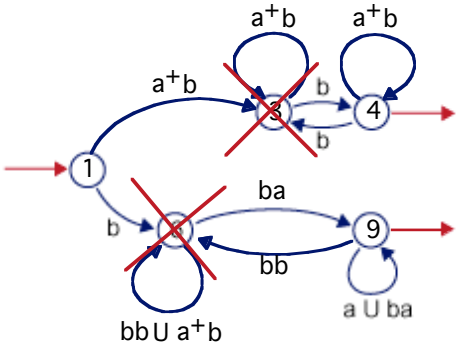
LÖSNING

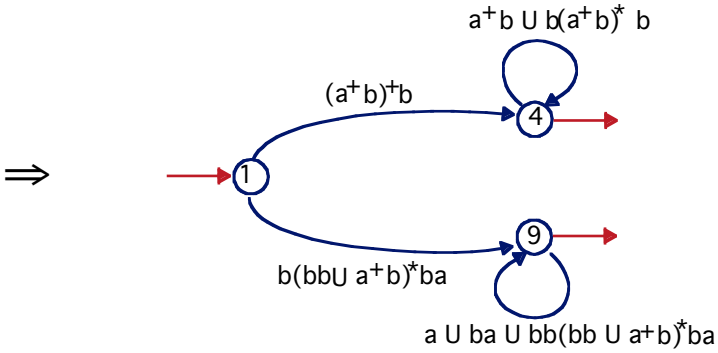
a) En DFA för L behöver ha två komponenter, en som tar hand om strängarna som inleds med a (och avslutas med b) och en för strängarna som inleds med b (och avslutas med a):



b) Medelst tillståndselimination får man på gängse sätt en GFA, och därefter ett reguljärt uttryck: $(a^+\,b)^+\,b\,(a^+\,b \cup b\,(a^+\,b)^*\,b\,)^* \;\cup\; b\,(b^2 \cup a^+\,b)^*\,b\,a\,(a \cup b\,a \cup b^2\,(b^2 \cup a^+\,b)^*\,b\,a)^*$







3. Betrakta följande klasser av språk (8p)

K_1: reguljära

K_2: inte reguljära, men sammanhangsfria

K_3: inte sammanhangsfria

Placera (med motiveringar) nedanstående språk i rätt klass.

a) $L_1 = \{w \text{ över } \{a, b\} \mid Antal(a, w) = 2 \cdot Antal(b, w)\}$,

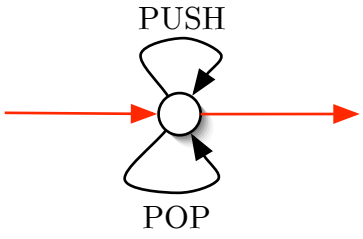
b) $L_2 = \{w \text{ över } \{a, b\} \mid Antal(a, w) = 2^{Antal(b,w)}\}$,

c) $L_3 = \{w \text{ över } \{a, b\} \mid w = x\,y \text{ och } Antal(a, x) = Antal(a, y)\}$.

ANM. $Antal(a, w)$ betecknar antalet a-förekomster i w. Om du menar att ett språk ligger i K_1 räcker det att presentera ett reguljärt uttryck eller en finit automat för språket ifråga. Om du menar att det ligger i K_2 måste du bevisa bristen på reguljäritet och dessutom presentera en CFG eller en PDA för språket. Om du menar att det ligger i K_3 måste du bevisa bristen på sammanhangsfrihet.

LÖSNING

a) L_1 tillhör K_2. Bristen på reguljäritet följer tex av att $a^{2\,N}\,b^N$ inte kan pumpas på något sätt inuti a-blocket utan att man faller ur språket. (En urpumpning av a:n leder ju till att det blir för få a:n.) Men L_1 är sammanhangsfritt. Här är en PDA för språket:



PUSH $= \{(a,\varepsilon/a),\ (b,\varepsilon/bb)\}$

POP $= \{(a,\ b/\varepsilon),\ \ (b,\ aa/\varepsilon),\ \ (b,\ a/b)\}$

Och här är en CFG: $S \to aSaSbS \mid aSbSaS \mid bSaSaS \mid \varepsilon$

b) L_2 tillhör K_3. Bristen på sammanhangsfrihet följer tex av att $w = a^{2^K}\,b^K$ inte kan pumpas på något sätt i något enda block av längd K utan att man faller ur språket. Det senare förklaras av att *ingen* upp-pumpning kan bibehålla det givna förhållande mellan antalet a:n och antalet b:n, om pumpblocket *inte* är längre än K.

Ty en upp-pumpning i ett K-block kan högst ge K nya tecken. Och det räcker inte. Den närmast längre strängen (än w) är nämligen $a^{2^{K+1}}\,b^{K+1} = a^{2\cdot 2^K}\,b^{K+1} = a^{2^K+\mathbf{2^K}}\,b^{K+\mathbf{1}}$. (Det behövs således minst $2^K + 1$ nya tecken.)

c) L_3 tillhör K_1. Det givna villkoret uttrycker nämligen bara att antal a:n i w skall vara jämnt. Här är ett reguljärt uttryck för sådana w: $b^* \cup (b^*\,a\,b^*\,a)^*\,b^*$

---

4. Ge informella beskrivningar av nedanstående språk och presentera sedan om möjligt både en CFG och en PDA för vart och ett av språken. Ifall det senare är omöjligt, motivera det!

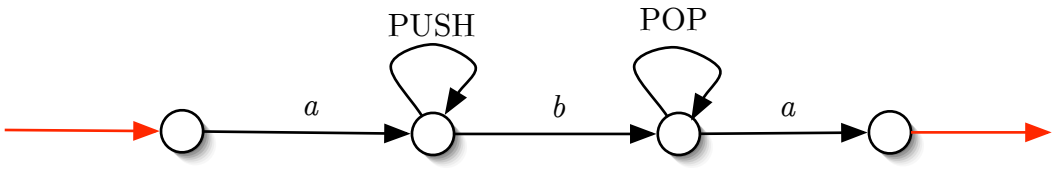
a) $L_1 = \{a\,w\,b\,w^{\text{rev}}\,a \mid w \in \{a, b\}^*\}$ (3p)

b) $L_2 = \{a\,x\,b\,y\,a \mid x, y \in \{a, b\}^* \text{ och } y \neq x^{\text{rev}}\}$ (3p)

LÖSNING

a) L_1:s strängar börjar och slutar på a och däremellan finns en udda palindromsträng med b i mitten.

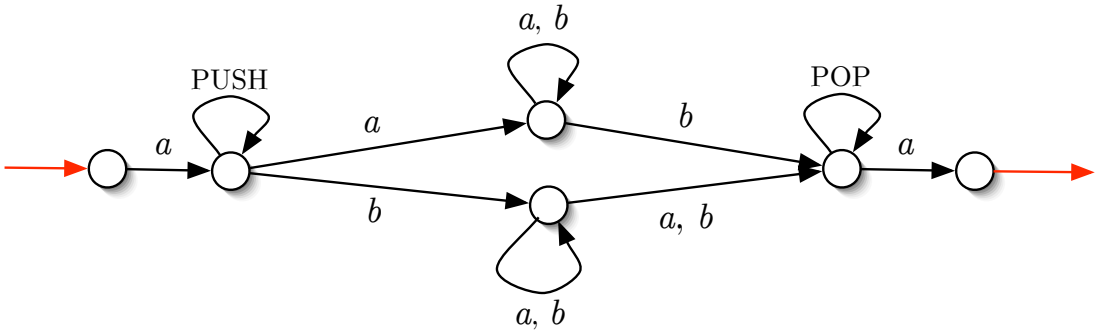
$S \to a\ W\ a, \quad W \to a\ W\ a \mid b\ W\ b \mid b$



$\text{PUSH} = \{(a,\varepsilon/a),\ (b,\varepsilon/b)\} \quad \text{POP} = \{(a,\ a/\varepsilon),\quad (b,b/\varepsilon)\}$

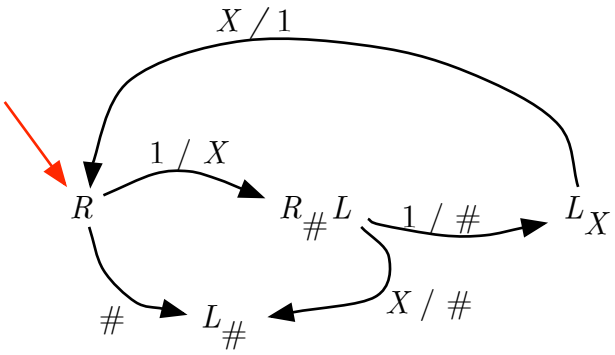
b) L_2:s strängar börjar och slutar på a. Däremellan finns en sträng w som innehåller minst ett b, som *inte* omgjärdas av en jämn palindromsträngs två halvor.
(Notera att strängar kan ligga både i L_1 *och* i L_2. Ex.vis ligger strängen $ababaaba$ i $L_1 \cap L_2$. Den ligger i L_2 tack vare att den har b-förekomster utanför mittpunkten: $ababaa = axbya$ med $x = baba$, och $y = \varepsilon$.)
Strängarna i L_2 kännetecknas av att de är av typen $a\,w\,a$, med någon b-förekomst utanför mittpunkten (se icketerminalen W nedanför). Att varje sådan sträng $a\,w\,a$ faktiskt ligger i L_2 följer av att den del (låt oss kalla den för x) av $a\,w\,a$ som ligger mellan det inledande a:et och den nämnda b-förekomsten har *annan* längd än den del (låt oss kalla den för y) som ligger mellan b-förekomsten och det avslutande a:et, och om $|y| \neq |x|$ följer att $y \neq x^{\text{rev}}$. Omvänt gäller också att varje sträng $a\,x\,b\,y\,a$ i L_2 är av typen $a\,w\,a$, med någon b-förekomst utanför mittpunkten. Ty endera ligger den understrukna b-förekomsten i $a\,x\,\underline{b}\,y\,a$ utanför mitten, och då är saken klar, eller så ligger den understrukna b-förekomsten i mitten, och i så fall är x och y lika långa, och kan därmed inte bestå av enbart a:n (VARFÖR?). Således finns det i detta fall någon b-förekomst inuti x eller y, vilken uppenbart ligger utanför mitten.

$S \to a\ W\ a$
$W \to a\ W\ a \mid b\ C\ b \mid a\ C\ b \mid b\ C\ a$
$C \to \varepsilon \mid a\ C \mid b\ C$



$\text{PUSH} = \{(a,\ \varepsilon/c),\ \ (b,\ \varepsilon/c)\}$ $\qquad$ $\text{POP} = \{(a,\ c/\varepsilon),\quad (b,\ c/\varepsilon)\}$

5. Turingmaskinen i figuren är en funktionsberäknande typ med inputalfabet {1} och tapealfabet {1, X, ⌗}. Vilken funktion beräknar den? (3p)



LÖSNING

Maskinen beräknar funktionen Kvot(x, 2), för godtyckliga naturliga tal x, där input och output representeras unärt.

MOTIVERING: Efter n varv i slingan från R och tillbaka till R suddar maskinen (i höger ände) bort **hälften** av 1:orna i inputsträngen. Om antalet 1:or i nämnda sträng var **jämnt** från början, säg lika med $2\,n$, så ser tapekonfiguration ut på följande sätt när R aktiveras i varv nummer $n+1$:

⌗1^n⌗ med pekaren placerad på den sista 1:an

Som du förstår kommer detta att leda till att maskinen (efter att R har agerat) drivs via #-övergången till $L_{⌗}$ som därefter stannar med pekaren omedelbart till vänster om de n 1:orna.

Om istället antalet 1:or var **udda** från början, säg lika med $2\,n+1$, så ser det ut så här efter n varv:

⌗1^{n+1}⌗ med pekaren placerad på den näst sista 1:an

Detta leder till att maskinen (efter att R har agerat) drivs via övergången $1\,/\,X$, och sedan drar igång seriekopplingen $R_{⌗}\,L$ för att slutligen gå via övergången $X\,/\,⌗$ till $L_{⌗}$ som stannar med pekaren till vänster om 1:orna. Dessa är nu n stycken till antalet.

---

6. Konstruera en alternativ men enklare grammatik än nedanstående dito. (6p)

$$
\begin{aligned}
S &\to A\,B \\
A &\to A\,C \mid a\,C \\
B &\to C\,B \mid C\,a \\
C &\to \varepsilon \mid a\,C\,a \mid a\,C\,b \mid b\,C\,a \mid b\,C\,b \\
b\,C\,b &\to C
\end{aligned}
$$

ANM. Inom hierarkin reguljär $\prec$ sammanhangsfri $\prec$ restriktionsfri finns de enklaste grammatikerna till vänster. Din grammatik skall beskriva samma språk som den givna grammatiken.

Full poäng erhålls enbart om en enklaste grammatik presenteras.

LÖSNING

Med A-reglerna produceras a följt av ett eller flera C:n, och B-reglerna ger på motsvarande sätt ett eller flera C:n följt av ett a.

Så S-regeln tillsammans med ovanstående regler säger att S är av typ $a\,C^n\,a$, där $n \geq 2$.

Med C-reglerna kan man producera *alla* strängar (över $\{a, b\}$) av *jämn* längd, även tom sträng. Den sista radens regel ($b\,C\,b \rightarrow C$) är värdelös. Ty med dess hjälp kan man inte åstadkomma något som man inte klarar utan densamma.

Sammantaget beskriver den givna grammatiken således språket (över $\{a, b\}$) av strängar som har jämn längd och som börjar och slutar på a - ett reguljärt språk.

Här är en reguljär grammatik för språket: $S \rightarrow a\,A$, $A \rightarrow a \mid a\,a\,A \mid a\,b\,A \mid b\,a\,A \mid b\,b\,A$

---

7. Är följande problem

"Innehåller $L(M)$ enbart strängar av jämn längd?"

a) avgörbart om M är en godtycklig finit automat? (3p)

b) avgörbart om M är en godtycklig Turingmaskin? (3p)

Om du menar att problemet är avgörbart, ska du beskriva (informellt) en algoritm som avgör problemet. Om du menar att problemet inte är avgörbart, förväntar jag mig en motivering som visar att ett antagande om avgörbarhet leder till något motsägelsefullt, eller en motivering som använder sig av RICES' SATS. I det senare fallet måste du ange mängden Ω på ett korrekt sätt för att få full poäng på uppgiften.

LÖSNING

a) Avgörbart. För given finit automat M med inputalfabet Σ behöver man bara konstruera en DFA för $L(M) \cap \Sigma(\Sigma^2)^*$ och undersöka ifall den saknar accepterandetillstånd. (En sådan DFA *saknar* accepterandetillstånd omm svaret på det givna problemet är JA.)

b) Oavgörbart. Följer av RICES' sats. Ty det givna problemet handlar om en icketrivial egenskap för Turingaccepterbara språk - nämligen egenskapen att innehålla enbart strängar av jämn längd. FORMELLT: Låt Ω vara mängden av Turingaccepterbara språk som innehåller enbart strängar av jämn längd. Det finns Turingaccepterbara språk inuti Ω (ex.vis $\{\varepsilon\}$) och det finns Turingaccepterbara språk utanför Ω (ex.vis $\{a\}$). Då följer enligt RICES' sats att ingen TM kan avgöra för godtycklig TM M ifall $L(M) \in \Omega$.