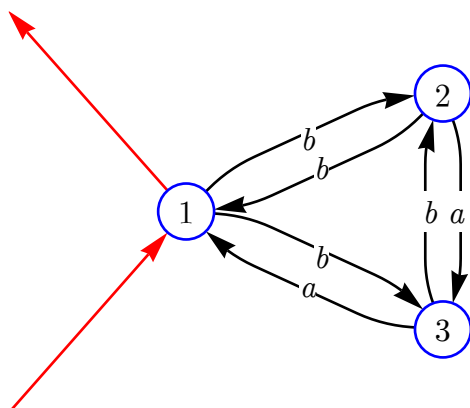


SKRIVTID: 8-13, HJÄLPMEDEL: Inga.

MOTIVERA ALLA LÖSNINGAR NOGGRANT.

BETYGSREGLER: BETYG = $Om(poäng \geq 32, 5, Om(poäng \geq 25, 4, Om(poäng \geq 18, 3, U)))$

1. (6p) Betrakta nedanstående NFA.



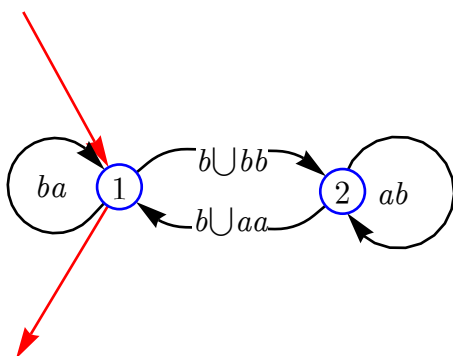
a) Beskriv NFA:s språk med ett reguljärt uttryck.

b) Konstruera med hjälp av delmängdskonstruktionen en DFA ekvivalent med NFA:n.

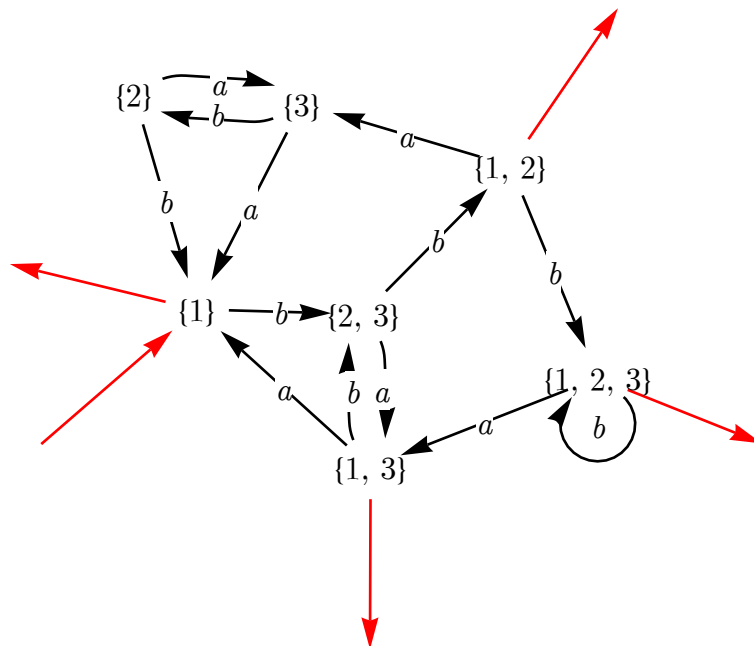
c) Är den resulterande DFA:n minimal?

LÖSNING

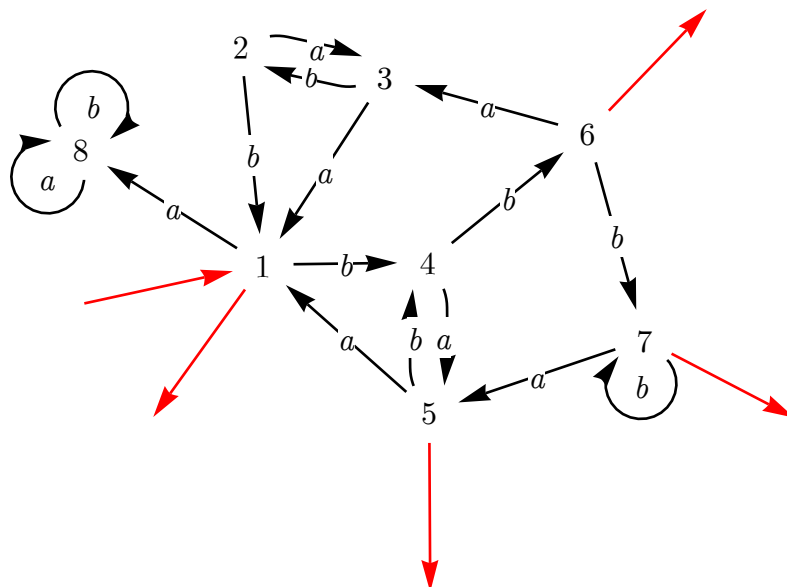
a) Efter elimination av tillståndet 3 får man nedanstående GFA, vars språk ges av $(ba \cup ((b \cup bb)(ab)^*(b \cup aa)))^*$.



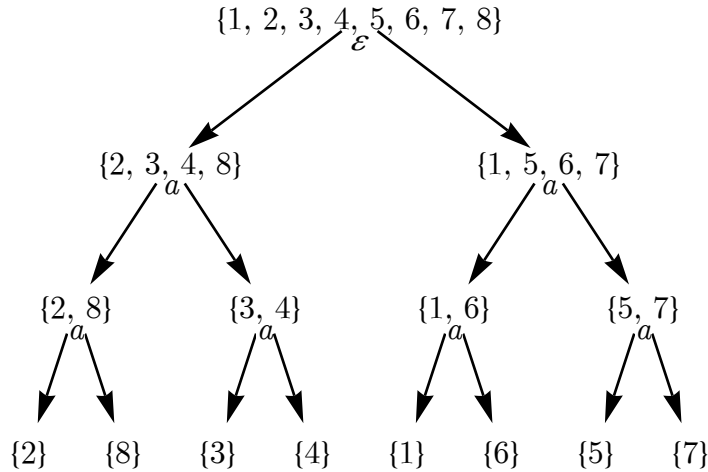
b) Delmängdskonstruktionen på den givna NFA:n ger (skräptillståndet ej utritat)



Här är DFA:n med nya tillståndsbeteckningar (och skräptillstånd utritat)



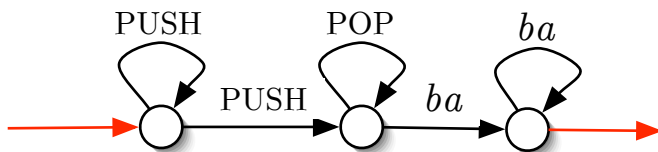
c) Ja, DFA:n ovanför är minimal. Se trädets nedanför.



2. (3p) Konstruera en PDA för det språk som beskrivs av reglerna
 $S \rightarrow AB$, $A \rightarrow ab \mid aAb$, $B \rightarrow ba \mid baB$.

LÖSNING

A och B representerar $\{a^m b^m \mid m \in \mathbb{N}_+\}$ respektive $\{(ba)^n \mid n \in \mathbb{N}_+\}$. Det beskrivna språket är därför lika med $\{a^m b^m (ba)^n \mid m, n \in \mathbb{N}_+\}$, och här är en PDA för detta språk:



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

3. (4p) Visa hur en CFG för ett språk L alltid kan modifieras till
 (a) en CFG för L^* (b) en CFG för L^{rev} ,
 samt illustrera dina konstruktionsidéer på reglerna i uppgift 2.

LÖSNING

Antag att L beskrivs med en CFG vars startsymbol är S . Eftersom strängarna i L^* fås genom att till tom sträng foga godtyckligt många strängar från L , får man en CFG för L^* om man utvidgar L 's regler med $S_0 \rightarrow \epsilon \mid S S_0$ och låter S_0 vara ny startsymbol.

En grammatik för L^{rev} fås genom att man reverserar alla högerled i en CFG för L . Ty på detta sätt kommer varje produktionsträd att spegelvändas, varvid löven (som beskriver den producerade strängen) blir "uppräknade" i omvänd ordning. Se bilden nedanför.

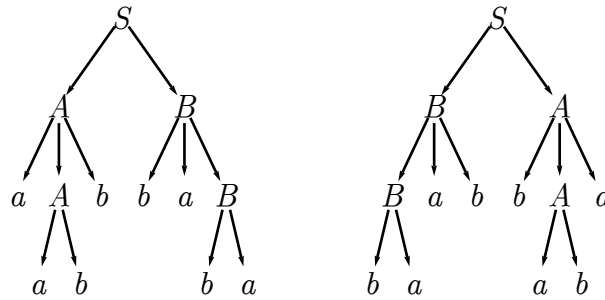


Illustration på CFG:n i uppgift 2:

$L^*: S_0 \rightarrow \varepsilon \mid S S_0, S \rightarrow A B, A \rightarrow a b \mid a A b, B \rightarrow b a \mid b a B.$

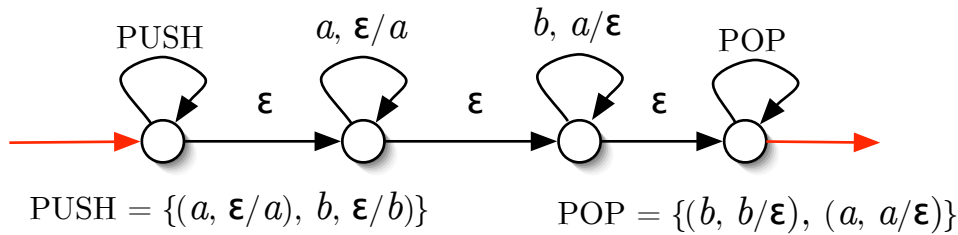
$L^{\text{rev}}: S \rightarrow B A, A \rightarrow b a \mid b A a, B \rightarrow a b \mid B a b$

4. (6p) Presentera både en CFG och en PDA för språket $\{x a^n b^n x^{\text{rev}} \mid x \in \{a, b\}^*\}$.

LÖSNING

Först en CFG: $S \rightarrow a S a \mid b S b \mid A, A \rightarrow \varepsilon \mid a A b.$

Sedan en PDA:



5. (7p) Betrakta följande tre kategorier av språk

- reguljära språk,
- inte reguljära, men sammanhangsfria,
- inte sammanhangsfria.

Placera (med motiveringar) nedanstående språk i rätt kategori.

- (a) $\{a^i a^j a^{i+j} \mid i, j \in \mathbb{N}\}$
 (b) $\{a^i b^j a^{i+j} \mid i, j \in \mathbb{N}\}$
 (c) $\{a^i b^j a^{i \cdot j} \mid i, j \in \mathbb{N}\}$
 (d) $\{x a^n y b^n x^{\text{rev}} \mid x, y \in \{a, b\}^*, n \in \mathbb{N}\}$

LÖSNING

(a) $\{a^i a^j a^{i+j} \mid i, j \in \mathbb{N}\} = \{a^{2(i+j)} \mid i, j \in \mathbb{N}\} = \{a^{2k} \mid k \in \mathbb{N}\} = (a^2)^*$ Reguljärt således.

(b) Ej reguljärt, ty $a^N b a^{N+1}$ kan inte pumpas i första N -blocket. Sammanhangsfritt, ty beskrivs av $S \rightarrow a S a \mid A, A \rightarrow \varepsilon \mid b A a.$

(c) Ej sammanhangsfritt.

Följer om vi kan visa att $w = a^N b^N a^{N^2}$ inte kan pumpas inuti något N -block på det sätt som pumpsatsen föreskriver.

Först kan vi konstatera att om pumpning av en sträng av typ $a^i b^j a^{i \cdot j}$ skall resultera i en ny sträng av samma typ, räcker det inte att pumpa i en enda av de tre delarna a^i , b^j , $a^{i \cdot j}$. Ty sådan pumpning förstör relationen mellan de tre delarnas längder.

Nej, man måste pumpa i minst en av de två första delarna och dessutom i den tredje. (1)

Eftersom ett N -block inuti ovan nämnda w inte rymmer tecken från alla tre delarna a^N , b^N , a^{N^2} följer av (1) att pumpning inuti b^N och a^{N^2} är allt vi behöver diskutera.

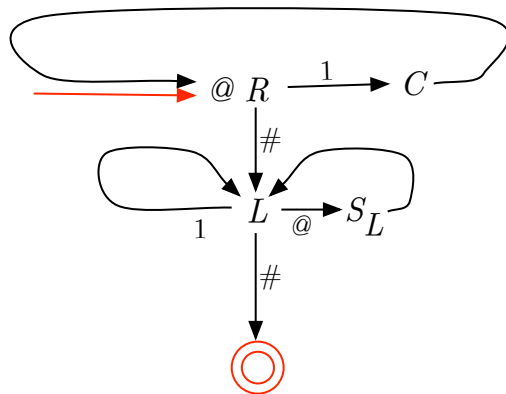
Men sådan pumpning kan omöjligen resultera i en sträng av typ $a^i b^j a^{i \cdot j}$ vid **en** upp-pumpning. Varför?

Jo, strängen $a^N b^{N+1} a^{N \cdot (N+1)}$ är den närmast längre sträng än w av typ $a^i b^j a^{i \cdot j}$ som möjligen kan uppstå genom sådan pumpning. Den är emellertid $N + 1$ tecken längre än w , och kan därför inte uppstå vid **en** upp-pumpning i ett N -block.

Det följer att man – vid **en** upp-pumpning inuti ett N -block – faller ur språket oavsett hur N -blocket är placerat i w , och oavsett vilka pumpdelar man väljer.

(d) Reguljärt. Ty $L_4 = \{x a^n y b^n x^{\text{rev}} \mid x, y \in \{a, b\}^*, n \in \mathbb{N}\} = \{a, b\}^*$, vilket följer av att L_4 innehåller $\{\varepsilon a^0 y b^0 \varepsilon^{\text{rev}} \mid y \in \{a, b\}^*\}$, som är lika med $\{a, b\}^*$.

6. (6p) Följande Turingmaskin – vars tapealfabet är $\{1, @, \#\}$ – beräknar en funktion från $\{1\}^*$ till $\{1\}^*$. Vilken är denna funktion? Du måste förklara hur de olika delarna i maskinen medverkar till att maskinens beräkning blir just det som du föreslår.



ANM. Kopiatorn C omvandlar $\dots \underset{\wedge}{\sigma} w \# \dots$ till $\dots \sigma w \# w \# \dots$, och vänstershiftaren S_L omvandlar $\dots \sigma_1 \sigma_2 \sigma_3 w \# \dots$ till $\sigma_1 \sigma_3 w \# \dots$.

LÖSNING

Maskinen beräknar funktionen $1^n \mapsto 1^{n+(n-1)+\dots+0}$, $n \in \mathbb{N}$, dvs. den omvandlar en sträng av n ettor till en med $\sum_{k=0}^n k$ stycken dito.

MOTIVERING:

För blank tape kommer den övre delen av maskinen att omvandla $\underset{\wedge}{\# \#} \dots$ till $\underset{\wedge}{@ \#} \dots$. Den undre delen går sedan ett steg till vänster på tapen och shiftar bort $@$. Den stannar (efter ha tagit ett steg till vänster) i konfigurationen $\underset{\wedge}{\# \#} \dots$.

För ickeblank tape omvandlar översta delen av maskinen $\underset{\wedge}{\# 1^n \#} \dots$ till $\underset{\wedge}{@ 1^n @ 1^{n-1} @ \dots @ 1 @ \#} \dots$.

Den undre shiftar sedan bort samtliga $@$, och stannar när resultatet blir lika med $\underset{\wedge}{\# 1^n 1^{n-1} \dots 1 \#} \dots$, dvs. $\underset{\wedge}{\# 1^{n+(n-1)+\dots+1} \#} \dots$.

7. (8p) Diskutera för vart och ett av följande problem om det finns någon TM som kan avgöra problemet ifråga.

- a) Givet en godtycklig TM M , accepterar M fler än 20 strängar?
- b) Givet en godtycklig DFA M , accepterar M fler än 20 strängar?
- c) Givet en godtycklig TM M , finns det någon inputsträng för vilken M stannar inom 20 steg?

ANM Om Du menar att ett problem är avgörbart, måste Du motivera denna åsikt genom att skissera åtminstone en informellt beskriven algoritm. Annars (vid oavgörbarhet) måste Du motivera det oavgörbara t.ex. med hjälp av Rices sats eller på något annat sätt.

LÖSNING

a) Oavgörbart. Följer av RICES sats om man sätter $\Omega = \{L \mid L \text{ är Turingaccepterbart och } |L| > 20\}$

b) Avgörbart. Ty en DFA M accepterar fler än 20 strängar omm

(i) $L(M)$ är ändlig och $|L(M)| > 20$ eller

(ii) $L(M)$ är oändlig.

Därför kan följande procedur avgöra problemet:

Kontrollera först om M innehåller någon slinga på väg från start till acceptans. Om så är fallet är $L(M)$ oändlig och svaret på den givna frågan blir JA. Annars är $L(M)$ ändlig, och då kan M inte acceptera några andra strängar än de som är kortare än N , där N betecknar antalet tillstånd i M . Därmed räcker det att provköra M på de strängar över M 's inputalfabet vars längd är $< N$, och därefter kontrollera om de som accepterades var flera än 20.

c) Avgörbart. Följer av att inga andra strängar än de av längd ≤ 20 hinner undersökas vid körning i 20 steg, och att det bara finns ändligt många strängar sådana strängar (över M 's inputalfabet). Således behöver man bara provköra ändligt många strängar, var och en i 20 steg.