

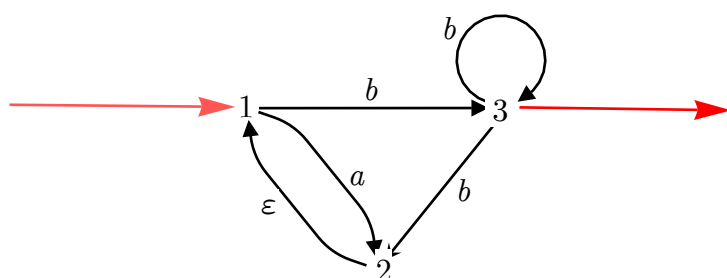
PROV I MATEMATIK AUTOMATATEORI 18 okt 2012

SKRIVTID: 8-13. HJÄLPMEDEL: Inga.

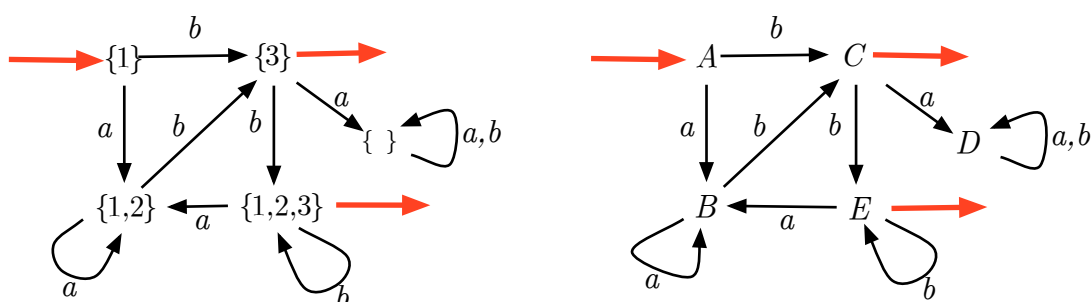
MOTIVERA ALLA LÖSNINGAR NOGGRANT.

BETYGSGRÄNSER: För betygen 3, 4 respektive 5 krävs totalt minst 18, 25 respektive 32 poäng.

1. (6p) (a) Använd delmängdskonstruktionen för att tillverka en DFA ekvivalent med nedanstående NFA. (b) Är den resulterande DFA:n minimal?

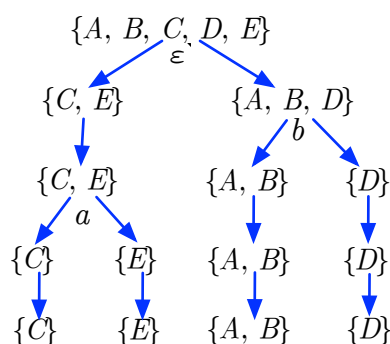


LÖSNING (a) Med delmängdskonstruktionen erhålls följande DFA.



(b) Den är *inte* minimal. Tillstånden A och B kan nämligen slås ihop. Det beror på att båda tillstånden är ickeaccepterande, och att DFA:n drivs vidare från nämnda tillstånd till ett och samma tillstånd, oavsett om DFA:n står i A eller B . Med andra ord: DFA:ns språk särskiljer inte tom sträng från strängen a .

Med känd minimeringsalgorithm ser man f.ö. (se nedanför) att inga andra tillstånd kan slås ihop:



2. (6p) Konstruera reguljära uttryck och DFA:er för

(a) $L = \{w \in \{a, b\}^* \mid w \text{ innehåller exakt två } b:n, \text{ och dessa har ett udda antal } a:n \text{ mellan sig}\}.$

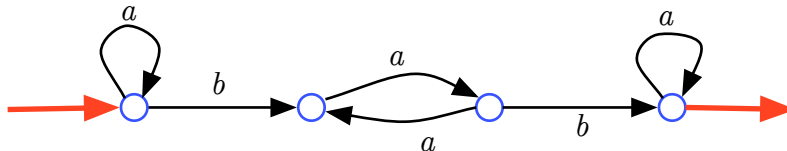
(b) L 's komplementspråk $\bar{L}$ där komplementet tas med avseende på $\{a, b\}^*$.

T.ex. är $baaabaa$ en sträng i L , medan ε , aa , $abaa$, $abaab$ och $bbba$ är strängar i $\bar{L}$.

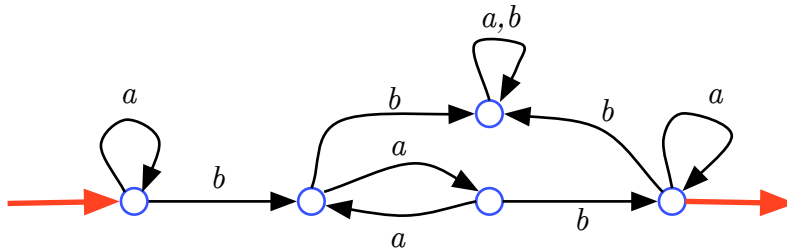
LÖSNING

(a) Följande reguljärt uttryck talar för sig själv: $a^* b a (a a)^* b a^*$.

Med det reguljära uttrycket som stöd är det lätt att rita en DFA utan skräptillstånd:



Samma DFA med skräptillstånd utritad blir något klottrigare:



men kommer till användning i (b)

(b) Strängarna i $\bar{L}$ kännetecknas av att de tillhör någon av följande fyra kategorier:

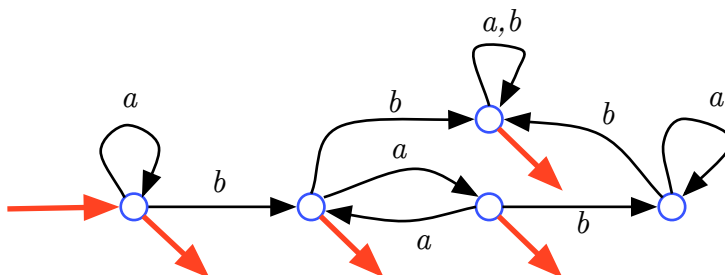
- (1) de som saknar b -förekomster,
- (2) de som har exakt en b -förekomst,
- (3) de som har exakt två b -förekomster, och där dessa har ett jämnt antal $a:n$ mellan sig,
- (4) de som har tre eller flera b -förekomster.

Härav följande reguljära uttryck

$$a^* \cup a^* b a^* \cup a^* b (a a)^* b a^* \cup (a^* b)^3 (a \cup b)^*$$

Strängarna i $\bar{L}$ är precis de strängar som driver en DFA för L till ickeaccepterande tillstånd.

Härav följande DFA för $\bar{L}$.



3. (8p) Låt L_0 vara språket som ges av $a^* b b a^*$, och betrakta

- (a) $L_1 = \{u u \mid u \in L_0\}$, (b) $L_2 = L_0 L_0$,
 (c) $L_3 = L_1 L_1$, (d) $L_4 = \{x x \mid x \in L_1\}$.

Avgör för vart och ett av L_1, L_2, L_3, L_4 om det är reguljärt, inte reguljärt men sammanhangsfritt, eller om det inte ens är sammanhangsfritt.

LÖSNING

- a) L_1 är inte reguljärt men sammanhangsfritt. Bristen på reguljäritet följer t.ex. av att $a^N b b a^0 a^N b b a^0$ inte kan pumpas på något sätt inuti första N -blocket utan att falla ur språket. Att L_1 är sammanhangsfritt beror på att varje sträng $a^m b b a^n a^m b b a^n$ i L_1 kan skrivas som $a^m b b a^m a^n b b a^n$, och således är en sammanfogning av två strängar som var och en ges av reglerna $X \rightarrow a X a \mid b b$. Därför kan en CFG för L_1 se ut så här: $S \rightarrow X X, X \rightarrow a X a \mid b b$.
- b) $L_2 = L_0 L_0$ är reguljärt. Ty L_0 är det, och de reguljära språken är slutna under sammanfogning (ihopslagning). L_2 kan t.ex. beskrivas av det reguljära uttrycket $a^* b b a^* b b a^*$.
- c) $L_3 = L_1 L_1$ är inte reguljärt men sammanhangsfritt. Bristen på reguljäritet följer t.ex. av att $w = u_1 u_1 u_2 u_2 = a^N b b a^0 a^N b b a^0 a^0 b b a^0 a^0 b b a^0$ (där $u_1 = a^N b b a^0$ och $u_2 = a^0 b b a^0$) inte kan pumpas på något sätt i det första N -blocket utan att falla ur L_3 . Att $L_3 = L_1 L_1$ är sammanhangsfritt kan förklaras av att L_1 är sammanhangsfritt, och att de sammanhangsfria språken är slutna under sammanfogning.
- d) L_4 är inte sammanhangsfritt. Ett pumpbevis för detta kan se ut som följer: $w = x x = u u u u = a^N b^2 a^N b^2 a^N b^2 a^N b^2$ (där $u = a^N b^2 \in L_0$) kan inte pumpas inuti något enda N -block utan att falla ur L_4 . Om w efter pumpning skall bli en ny sträng i L_4 , måste nämligen alla fyra u :na påverkas. Och det senare låter sig inte göras vid pumpning inuti ett N -block, eftersom avståndet från det första u :et till det sista är större än N .

4. (6p) Konstruera både en CFG och en PDA för språket $\{a^n b^n \mid n \text{ är inte delbart med } 3\}$.

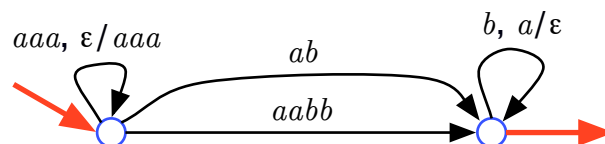
LÖSNING Naturliga tal som inte är delbara med 3 ges av $3k+1$ eller $3k+2$, $k \in \mathbb{N}$ varför det givna språket beskrivs av unionen

$$\{a^{3k+1} b^{3k+1} \mid k \in \mathbb{N}\} \cup \{a^{3k+2} b^{3k+2} \mid k \in \mathbb{N}\} = \{a^{3k} a b b^{3k} \mid k \in \mathbb{N}\} \cup \{a^{3k} a^2 b^2 b^{3k} \mid k \in \mathbb{N}\}.$$

Man ser att kortaste strängarna i det aktuella språket är ab och $a^2 b^2$, och att varje *längre* sträng inleds med tre a :n, åtföljs av en kortare sträng ur språket, och avslutas med tre b :n. Det följer att språket beskrivs av CFG:n

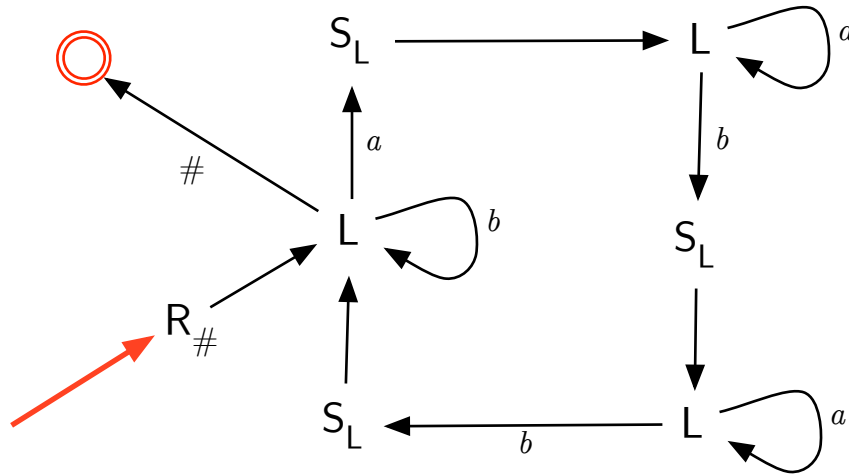
$$S \rightarrow a b \mid a^2 b^2 \mid a^3 S b^3$$

Så här kan en PDA se ut. Motivering av densamma återfinns under figuren.



Med hjälp av de två övergångarna som saknar stackagerande, tar PDA:n hand om de två kortaste strängarna. Slingan på det första tillståndet ser till att strängar med ytterligare inledande block av tre a :n får a :na bokförda på stacken. Slingan på det sista tillståndet kontrollerar till sist att ytterligare avslutande b :n stämmer i antal med nämnda inledande a :n.

5. (7p) Beskriv med en så enkel grammatik som möjligt (se ANM. 2 nedan) det språk över $\{a, b\}$ som Turingmaskinen nedanför accepterar. Argumentera för att det inte finns någon enklare grammatik. Om du bara kan beskriva informellt vilket språk som maskinen accepterar kan du bara bli sparsamt belönad. Vidare måste du förklara hur de olika delarna i maskinen medverkar till att maskinens språk blir just det som du föreslår. Annars riskerar du att bli helt utan poäng.



ANM. 1 Vänstershiftaren S_L omvandlar $\dots \# x \underset{\Delta}{\sigma} y \# \dots$ till $\dots \# x y \underset{\Delta}{\#} \dots$.

ANM. 2 Inom hierarkin reguljär < sammanhangsfri < restriktionsfri finns de enklaste grammatikerna till vänster.

LÖSNING Turingmaskinens arbete efter den inledande körningen av $R_{\#}$ kan beskrivas som att maskinen under varje lyckad (stor) rundtur från det första L:et och tillbaka skannar av den ickeblanka delen av tapen tre gånger från höger till vänster, och därvid hittar först ett a och sedan två b :n, samt shiftar bort nämnda fynd från tapen. Efter k lyckade rundturer har således k stycken a :n och $2k$ stycken b :n shiftats bort.

En inputsträng som innehåller exakt k stycken a :n och minst $2k$ stycken b :n driver således maskinen att göra k rundturer av nämnt slag. Vid nästkommande rundturs inledning påträffas inte några a :n, med följd att maskinen drivs till stopptillståndet.

För en sträng som innehåller $k > 0$ stycken a :n men färre än $2k$ stycken b :n kommer TM:en under den k :te rundan att hänga sig under någon av de två jakterna efter ett b .

Resonemanget visar att TM:en stannar för sådana inputsträngar som innehåller minst dubbelt så många b :n som a :n.

Här är en CFG för TM:ens språk: $S \rightarrow \varepsilon \mid bS \mid aSbSbS \mid bSaSbS \mid bSbSaS$.

Att inga felaktiga strängar kan produceras med dessa regler följer av att varje regel som innehåller a innehåller två b :n. (Regeln $S \rightarrow bS$ fogar bara in flera b :n.) Att verkligen *alla* korrekta strängar kan produceras beror på att de tre avslutande reglerna beskriver *alla* sätt att infoga ett a och två b :n inuti en kortare sträng i språket, och att alla strängar med flera b :n än dubbla antalet a :n kan åstadkommas genom att dessutom använda regeln $S \rightarrow bS$ på lämpliga sätt.

Språket är förstås *inte* reguljärt, eftersom t.ex. $a^N b^N b^N$ *inte* kan pumpas på något sätt inuti det inledande N -blocket utan att falla ur språket. Därför finns det ingen reguljär grammatik för maskinens språk.

6. (7p) Är problemet

“Innehåller $L(M)$ minst 100 strängar?”

- (a) avgörbart för godtyckliga finita automater M ?
- (b) avgörbart för godtyckliga Turingmaskiner M ?

ANM Om Du hävdar att ett visst problem är avgörbart, måste Du motivera ståndpunkten genom att skissera åtminstone en informellt beskriven algoritm. (Om Du menar att samma problem är oavgörbart, måste Du givetvis motivera denna åsikt också.)

LÖSNING

a) AVGÖRBART. Följande procedur avgör problemet:

Kontrollera först om M innehåller någon slinga på väg från start till acceptans. (Man kan ta reda på om en DFA M med N tillstånd har en sådan slinga genom att köra M på strängarna av längd $\geq N$ men $< 2N$, och se om någon sträng accepteras.) Om så är fallet är $L(M)$ oändlig och svaret på den givna frågan blir JA.

I annat fall 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 behöver man (i detta fall) bara provköra M på de strängar över M 's inputalfabet vars längd är $< N$, samt räkna de strängar som accepteras. Är de senare minst 100 stycken, så är svaret JA, annars NEJ.

b) OAVGÖRBART. Eftersom det finns Turingaccepterbara språk som innehåller 100 strängar (t.ex. varje oändligt språk), och sådana som inte gör det (t.ex. det tomma språket), så följer oavgörbarheten av RICES sats om man sätter $\Omega = \{L \mid L \text{ är Turingaccepterbart och } |L| \geq 100\}$.