Miroslav Ćirić
Jelena Ignjatović

TEORIJA ALGORITAMA, AUTOMATA I JEZIKA

-zbirka zadataka-

УНИВЕРЗИТЕТ У НИШУ
ПРИРОДНО-МАТЕМАТИЧКИ ФАКУЛТЕТ

DR MIROSLAV D. ĆIRIĆ, redovni profesor
Prirodno-matematičkog fakulteta Univerziteta u Nišu

DR JELENA M. IGNJATOVIĆ, docent
Prirodno-matematičkog fakulteta Univerziteta u Nišu

TEORIJA ALGORITAMA, AUTOMATA I JEZIKA
– ZBIRKA ZADATAKA
prvo izdanje, 2012

Izdavač
PRIRODNO-MATEMATIČKI FAKULTET
UNIVERZITETA U NIŠU

Za izdavača
PROF. DR DRAGAN ĐORĐEVIĆ, dekan

Urednik
PROF. DR IVAN MANČEV

Recenzenti
PROF. DR STOJAN BOGDANOVIĆ
PROF. DR PREDRAG STANIMIROVIĆ

Štampa
M KOPS CENTAR, NIŠ

Tiraž: 120 primeraka

ISBN 978-86-83481-87-3

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

519.713(075.8)(076)
519.76(075.8)(076)

ЋИРИЋ, Мирослав Д., 1964–
Teorija algoritama, automata i jezika : zbirka zadataka / Miroslav D. Ćirić, Jelena M. Ignjatović. - 1. izd. - Niš : Prirodno-matematički fakultet, 2012 (Niš : M kops centar). - VIII, 183 str. : graf. prikazi, tabele ; 24 cm

Tiraž 120. - Bibliografija: str. 177-178. - Registar.

ISBN 978-86-83481-87-3

1. Игњатовић, Јелена М., 1973– [аутор]
a) Аутомати, коначни b) Математичка лингвистика

COBISS.SR-ID 188136460

Odlukom Nastavno-naučnog veća Prirodno-matematičkog fakulteta u Nišu, broj 1072/1-01 od 21.12.2011. godine, ova knjiga je odobrena kao pomoćni udžbenik za predmet “Teorija algoritama, automata i jezika”.

Miroslav D. Ćirić
Jelena M. Ignjatović

Teorija algoritama, automata i jezika

– Zbirka zadataka –

Niš, 2012

Predgovor

Ova knjiga je zamišljena, prvenstveno, kao zbirka zadataka iz predmeta *Teorija algoritama, automata i jezika*, na prvoj godini Master akademskih studija na Departmanu za Računarske nauke Prirodno-matematičog fakulteta u Nišu. Problemi koji se obrađuju i izučavaju u okviru ovog predmeta obuhvataju formalne jezike, automate i druge modele izračunavanja. Zadatak predmeta je ne samo da se studenti upoznaju sa brojnim primenama automata i formalnih jezika, već i sa osnovnim matematičkim (prvenstveno algebarskim) pojmovima i alatima koji se koriste za opis apstraktnih objekata koji se izučavaju u računarskim naukama.

Ideja za pisanje ove zbirke rodila se nakon višegodišnjeg izvođenja vežbi i predavanja iz napred pomenutog predmeta. Na srpskom jeziku ne postoji zbirka zadataka koja prati program ovog kursa. Poznata nam je samo jedna zbirka na engleskom jeziku, ali se ona samo delimično bavi tom tematikom. Dakle, knjiga je prvenstveno namenjena potrebama pomenutog predmeta, sa naglaskom na probleme teorije jezika i automata koji su računarski rešivi, uz primenu alata iz univerzalne algebre, teorije polugrupa i teorije grafova. Trudili smo se da prikupimo mnoštvo različitih primera, počev od osnovnih pitanja vezanih za elementarne definicije i koncepte, do onih naprednih koji zahtevaju poznavanje sofisticiranijih matematičkih alata. Ideje i tehnike dokazivanja objašnjene su na konstruktivan način, uz primenu indukcije i rekurzije kad god je to moguće. Zbog ovakve koncepcije zbirku mogu koristiti i studenti doktorskih studija u oblasti računarskih nauka i matematike, za predmet *Formalni jezici, automati i izračunljivost*, kao i studenti tehničkih nauka i svi oni koji su zainteresovani za rešavanje problema iz oblasti teorijskog računarstva.

Zbirka se sastoji iz pet glava. Svaka glava najpre sadrži teorijske osnove za probleme koji se u njoj rešavaju, a nakon toga veći broj urađenih primera i zadatke predviđene za samostalni rad studenata, čije se rešavanje bazira ili neposredno sledi iz nekog od urađenih zadataka. Određeni broj primera predstavlja poznate teoreme iz ove oblasti koje su formulisane u obliku zadataka.

U prvoj glavi rešavaju se neki osnovni problemi vezani za razna uređenja na polugrupi reči, definišu se pojmovi jezika i gramatike, daje njihova klasifikacija i kroz veliki broj primera se rešavaju problemi vezani za ovu oblast, uključujući i probleme izvođenja u gramatici.

Druga glava posvećena je automatima bez izlaza i jezicima koji mogu biti raspoznati konačnim automatima. Predstavljeni su algoritmi za konstrukciju minimalnog automata datog jezika, minimizaciju datog automata, konstrukciju sintaksičkog monoida datog jezika, kao i primeri iz kojih se vidi značaj i primena Kleenijeve teorema i Leme o napumpavanju.

U trećoj glavi rešavaju se problemi vezani za predstavljanje konteksno-nezavisnih jezika pomoću konteksno-nezavisnih gramatika, obrađene su redukcije ovih gramatika, a potom i primeri konstrukcije potisnih automata kao apstraktnih mašina koje raspoznaju konteksno-nezavisne jezike.

Četvrta glava se bavi determinističkim i nedeterminističkim Turingovim mašinama, kao najopštijom matematičkom formalizacijom pojma algoritma. Turingove mašine opisuju tačno jezike generisane proizvoljnim gramatikama i precizno definišu granicu između problema koji su algoritamski rešivi i onih koji to nisu.

U poslednjoj glavi govori se o automatima sa izlazom, automatima Mealyevog i Mooreovog tipa, o transformacijama reči i funkcijama koje se mogu realizovati takvim automatima, kao i problemima ekvivalentnosti i minimizacije automata sa izlazom.

Koristimo priliku da se zahvalimo recenzentima, Prof. dr Stojanu Bogdanoviću i Prof. dr Predragu Stanimiroviću, na vrednim primedbama i savetima. Zahvalnost dugujemo i koleginicama Ivani i Zorani Jančić, koje su pažljivo čitale rukopis i dale niz korisnih sugestija. Drugi autor se posebno zahvaljuje svojoj majci, gospođi Svetlani Kovačević, za ogromno strpljenje i podršku tokom pisanja ove knjige.

Sadržaj

Glava 1
Formalni jezici i gramatike

1.1. Reči, uređenja na rečima i jezici

Neka je X neprazan skup koji nazivamo *alfabet*, a čije elemente nazivamo *slova*. *Reč* (*string*) nad alfabetom X definiše se kao konačan niz $x_1x_2\dots x_n$, gde su $x_1,x_2,\dots x_n \in X$ slova. *Dužina* reči u, koju označavamo sa $|u|$, je broj slova u toj reči. Prazan niz slova označava se sa e i naziva *prazna reč*. Jasno je da je $|e| = 0$.

Neka su $u = x_1x_2\dots x_n$ i $v = y_1y_2\dots y_m$ reči nad alfabetom X, pri čemu su $x_1,\dots x_n,y_1,\dots,y_m \in X$. Reči u i v su jednake ako su jednake kao nizovi, tj. ako je $m = n$ i $x_i = y_i$, za svaki $i \in \{1,2,\dots,n\}$.

Osnovna operacija nad rečima je operacija *nadovezivanja* ili *konkatenacije*. Konkatenacijom reči $u = x_1x_2\dots x_n$ i $v = y_1y_2\dots y_m$ dobijamo reč

$$u \cdot v = uv = x_1x_2\dots x_ny_1y_2\dots y_m.$$

Ova operacija je asocijativna i uvodimo sledeće oznake:

$$x^0 = e,\ x^1 = x \quad \text{i} \quad x^k = \underbrace{xx\dots x}_{k}.$$

Reverzna reč date reči $u = x_1x_2\dots x_n$ jeste reč $\overline{u} = x_n\dots x_2x_1$.

Jezik je proizvoljan skup reči nad datim alfabetom uključujući i prazan skup.

Skup svih reči, uključujući i praznu reč, nad proizvoljnim alfabetom X, označićemo sa X^*. Sa X^+ označavamo $X^* \setminus \{e\}$. Skup X^+ sa operacijom nadovezivanja predstavlja polugrupu koju nazivamo *polugrupa reči*, dok je X^*, u odnosu na operaciju konkatenacije *monoid* sa jedinicom e koji nazivamo *monoid reči*.

Kao i na svakoj algebarskoj strukturi i na polugrupi (monoidu) reči mogu se definisati pojmovi *kongruencije* i *homomorfizma*. Relacija ekvivalencije ϱ na X^+ (X^*) je *kongruencija* na polugrupi X^+ (monoidu X^*) ako za proizvoljne reči $u,v \in X^+$ ($u,v \in X^*$) i $x \in X$ važi

$$
\begin{aligned}
(u,v) \in \varrho &\Rightarrow (xu,xv) \in \varrho \qquad &\text{(leva kompatibilnost)}\\
(u,v) \in \varrho &\Rightarrow (ux,vx) \in \varrho \qquad &\text{(desna kompatibilnost).}
\end{aligned}
$$

Dakle, kongruencija na polugrupi X^+ (monoidu X^*) je ekvivalencija koja je, istovremeno levo kompatibilna (levo saglasna) i desno kompatibilna (desno saglasna). Pokazuje se da je data definicija kongrencije na polugrupi (monoidu) ekvivalentna sledećem tvrđenju:

Relacija ekvivalencije ϱ na X^+ (X^*) je kongruencija na polugrupi X^+ (monoidu X^*) ako za proizvoljne reči $u_1, u_2, v_1, v_2 \in X^+$ ($u_1, u_2, v_1, v_2 \in X^*$) iz $(u_1, u_2) \in \varrho$ i $(v_1, v_2) \in \varrho$ sledi $(u_1 v_1, u_2 v_2) \in \varrho$.

Preslikavanje $\varphi : X^+ \mapsto Y^+$ ($\varphi : X^* \mapsto Y^*$) je *homomorfizam* između dve polugrupe (monoida) reči ako važi jednakost $\varphi(u,v) = \varphi(u)\varphi(v)$, za sve reči $u, v \in X^+$ ($u, v \in X^*$).

Primetimo da je jezik L nad alfabetom X proizvoljan podskup monoida X^*. Kardinalnost datog jezika $L \subseteq X^*$, označavamo sa $|L|$.

Operacije na jezicima su skupovne operacije: *unija, presek, supstrakcija (razlika) i komplement*, kao i operacija *konkatenacije*. Uniju jezika, osim sa "$\cup$", označavamo i simbolima za sabiranje, tj. sa "+" i "Σ". Za dva jezika L_1 i L_2 njihov proizvod (konkatenacija) je $L_1 L_2 = \{uv | u \in L_1, v \in L_2\}$.

Neka je L^n skup svih reči jezika L dužine n. Jasno je da je

$$L^* = \bigcup_{n=0}^{\infty} L^n = \sum_{n=0}^{\infty} L^n. \tag{1.1}$$

Zadatak 1.1. *Koliko je reči dužine n, za $n \in \mathbb{N}^0$, moguće konstuisati nad alfabetom $X = \{x_1, x_2, \ldots x_k\}$, ?*

Rešenje: Postoji n pozicija u ovakvoj reči i svaka pozicija može da nosi jedan od k mogućih simbola. Dakle, postoji k^n reči nad alfabetom X čija je dužina tačno n. $\square$

Zadatak 1.2. *Rešiti jednačinu $u011 = 011u$ nad alfabetom $\{0,1\}$, odnosno, naći skup svih reči $u \in \{0,1\}^*$ koje zadovoljavaju datu jednačinu.*

Rešenje: Da bi jednakost važila ili je reč u prazna ili je reč 011 i prefiks i sufiks od u. (Očito da u ne može biti dužine 1 ili 2).

$$\begin{array}{rl} & 011\ \boxed{\qquad u \qquad} \\ = & \boxed{\qquad u \qquad}\ 011 \end{array}$$

Neka je $u = 011v$. Pomeranjem prvog pojavljivanja reči 011, iz uslova $011u = u011$ dobijamo da je $u = v011$. Opet smo dobili jednačinu $011v = v011$, pa rešenje u možemo da definišemo rekurzivno: $u = e$ ili $u = 011v$, pri čemu je $v \neq u$, takođe, rešenje polazne jednačine. Odavde nije teško zaključiti da su jedina netrivijalna rešenja jednačine $(011)^n$, za svako $n \geqslant 0$. $\square$

Zadatak 1.3. *Dokazati da za proizvoljan jezik L važi:*

(i) $L^* L^* = L^*$;

(ii) $LL^* = L^* L$;

(iii) $(L^*)^* = L^*$.

Rešenje: (i) Na osnovu jednakosti (1.1) dobijamo:

$$L^*L^* = (\sum_{n=0}^{\infty} L^n)(\sum_{m=0}^{\infty} L^m) = \sum_{n,m=0}^{\infty} L^n L^m = \sum_{i=0}^{\infty} L^i = L^*.$$

Tvrđenje (ii) se pokazuje jednostavno primenom (1.1).

(iii) Kako je $(L^*)^2 = L^*$, lako se pokazuje da za svaki broj $n \in \mathbb{N}$, važi $(L^*)^n = L^*$. Dakle,

$$(L^*)^* = \sum_{n=0}^{\infty} (L^*)^n = \{e\} + L^* = L^*,$$

čime smo tvrđenje dokazali. □

Zadatak 1.4. *Jednakost $L^+ = L^*$ važi ako i samo ako $e \in L$. Dokazati.*

Rešenje: Ako $e \in L$, onda je, jasno $\{e\} = L^0 \subseteq L \subseteq L^+$, što znači da je $L^* \subseteq L^+$. Kako obrat uvek važi, to imamo da je $L^* = L^+$.

Obratno, ako $e \notin L$ svaka reč jezika L ima pozitivnu dužinu. Pošto $e \in L^*$, jasno je da je $L^* \neq L^+$. □

Zadatak 1.5. ***(Ardenova lema)*** *Neka su L_1, L_2 jezici takvi da $e \notin L_1$ i L je jezik koji zadovoljava relaciju $L = L_1 L + L_2$. Dokazati da je tada $L = L_1^* L_2$.*

Rešenje: Indukcijom po dužini reči pokazaćemo da je $L \subseteq L_1^* L_2$.

Neka je $u = e$ i pretpostavimo da $e \in L = L_1 L + L_2$. Kako $e \notin L_1$ zaključujemo da $e \in L_2$, pa je $e \in L_1^* L_2$.

Pretpostavimo da, za sve reči $v \in L$ dužine $|v| \leqslant n$, važi $v \in L_1^* L_2$. Posmatramo proizvoljnu reč u dužine $n+1$. Ako $u \in L$, onda $u \in L_2 \subseteq L_1^* L_2$ ili je u oblika $u = pw$, za neku reč $p \in L_1$ i $w \in L$. U tom slučaju je $p \neq e$, pa je $|w| < |u|$, što znači da na $|w|$ možemo primeniti indukcijsku pretpostavku. Dakle, $w \in L_1^* L_2$ i $u \in L_1 L_1^* L_2 \subseteq L_1^* L_2$.

Obratno, opet koristimo indukciju po $n \in \mathbb{N}^0$ da dokažemo da je $L_1^n L_2 \subseteq L$. Za $n = 0$ imamo $L_1^0 L_2 = L_2 \subseteq L_1 L + L_2 = L$. Za $n > 0$, dobijamo, prema indukcijskoj pretpostavci, sledeću inkluziju:

$$L_1^n L_2 = L_1(L_1^{n-1} L_2) \subseteq L_1 L.$$

Prema tome, $L_1^n L_2 \subseteq L_1 L \subseteq L_1 L + L_2 = L$, za svaki $n \geqslant 0$, što znači da je $L_1^* L_2 \subseteq L$. Ovim je tvrđenje dokazano. □

Zadatak 1.6. *Neka su $L_1, L_2 \subseteq \{a,b\}^*$ jezici koji zadovoljavaju sledeće jednačine:*

$$\begin{aligned} L_1 &= \{e\} + \{a\}L_1 + \{b\}L_2, \\ L_2 &= \{e\} + \{b\}L_2. \end{aligned}$$

Naći jednostavne reprezentacije jezika L_1 i L_2.

Rešenje: Primenom Ardenove leme na drugu jednačinu dobijamo

$$L_2 = \{b\}^*\{e\} = \{b\}^*.$$

Zatim primenjujemo Ardenovu lemu na prvu jednačinu:

$$L_1 = \{a\}^*(\{e\} + \{b\}L_2) = \{a\}^*(\{e\} + \{b\}\{b\}^*) = \{a\}^*\{b\}^*.$$

Dobili smo reprezentacije jezika $L_1 = \{a\}^*\{b\}^*$ i $L_2 = \{b\}^*$. □

Zadatak 1.7. *Za svaka dva jezika L_1 i L_2 važi jednakost $(L_1 + L_2)^* = (L_1^* L_2^*)^*$. Dokazati.*

Rešenje: Kako je $L_1 \subseteq L_1 + L_2$ i $L_2 \subseteq L_1 + L_2$ važi $L_1^* \subseteq (L_1 + L_2)^*$ i $L_2^* \subseteq (L_1 + L_2)^*$, pa je

$$L_1^* L_2^* \subseteq (L_1 + L_2)^*(L_1 + L_2)^* = (L_1 + L_2)^*.$$

Sa druge strane je $L_1 \subseteq L_1^* \subseteq L_1^* L_2^*$ i $L_2 \subseteq L_2^* \subseteq L_1^* L_2^*$, odakle dobijamo

$$L_1 + L_2 \subseteq L_1^* L_2^* + L_1^* L_2^* = L_1^* L_2^*.$$

Dakle, $(L_1 + L_2)^* \subseteq (L_1^* L_2^*)^*$. Ovim je dokaz kompletiran. □

Zadatak 1.8. *Za sve jezike L_1 i L_2 nad alfabetom X važi $(L_1^* L_2)^* L_1^* = (L_1 + L_2)^*$. Dokazati.*

Rešenje: Kako je $L_1 \subseteq L_1 + L_2$ i $L_2 \subseteq L_1 + L_2$ važi $L_1^* \subseteq (L_1 + L_2)^*$ i $L_2 \subseteq (L_1 + L_2)^*$, odakle je $L_1^* L_2 \subseteq (L_1 + L_2)^*$ i $(L_1^* L_2)^* L_1^* \subseteq (L_1 + L_2)^*$. Dokažimo obratnu inkluziju:

$$\begin{aligned}
((L_1^* L_2)^* L_1^*)^2 &= ((L_1^* L_2)^* L_1^*)((L_1^* L_2)^* L_1^*) \\
&= (L_1^* L_2)^* L_1^*(\{e\} + L_1^* L_2 (L_1^* L_2)^*) L_1^* \\
&= (L_1^* L_2)^* L_1^* L_1^* + (L_1^* L_2)^* L_1^* L_1^* L_2 (L_1^* L_2)^* L_1^* \\
&= (L_1^* L_2)^* L_1^* + (L_1^* L_2)^* L_1^* L_2 (L_1^* L_2)^* L_1^* \\
&= (L_1^* L_2)^*(\{e\} + L_1^* L_2 (L_1^* L_2)^*) L_1^* \\
&= (L_1^* L_2)^*(L_1^* L_2)^* L_1^* = (L_1^* L_2)^*
\end{aligned}$$

Dakle, dokazali smo da, za jezik $L = (L_1^* L_2)^* L_1^*$, važi $L = L^2$. Indukcijom pokazujemo da je $L = L^n$, za svaki $n \geqslant 1$, pa je, jasno, $L^* = \{e\} + L$, tj. $((L_1^* L_2)^* L_1^*)^* = \{e\} + (L_1^* L_2)^* L_1^* = (L_1^* L_2)^* L_1^*$. Iz ove jednakosti i prema tvrđenju prethodnog zadatka važi

$$\begin{aligned}
(L_1^* L_2)^* L_1^* &= \Big((L_1^* L_2)^* L_1^*\Big)^* = \Big((L_1^* L_2) + L_1\Big)^* = \Big((\{e\} + L_1 L_1^*) L_2 + L_1\Big)^* \\
&= (L_2 + L_1 L_1^* L_2 + L_1)^*.
\end{aligned}$$

Takođe važi $L_1 + L_2 \subseteq (L_2 + L_1 L_1^* L_2 + L_1)^*$, odnosno

$$(L_1 + L_2)^* \subseteq (L_2 + L_1 L_1^* L_2 + L_1)^* = (L_1^* L_2)^* L_1^*.$$

Ovim je tvrđenje dokazano. □

Prefiks, sufiks i faktor uređenje

Definišimo na skupu X^* sledeće relacije:

$$u \leqslant_p v \Leftrightarrow u \textit{ je prefiks od } v,$$
$$u \leqslant_s v \Leftrightarrow u \textit{ je sufiks od } v,$$
$$u \leqslant_f v \Leftrightarrow u \textit{ je faktor od } v.$$

Označimo sa $u <_p v$ (odnosno $u <_s v$, $u <_f v$), *pravi prefiks* (odnosno *pravi sufiks*, *pravi faktor*) u reči v, tj. neka je:

$$\begin{aligned} u <_p v &\Leftrightarrow u \leqslant_p v \text{ i } u \neq v, \\ u <_s v &\Leftrightarrow u \leqslant_s v \text{ i } u \neq v, \\ u <_f v &\Leftrightarrow u \leqslant_f v \text{ i } u \neq v. \end{aligned}$$

Zadatak 1.9. *Relacija $\leqslant_p$ je uređenje na X^*. Dokazati.*

Rešenje: Proverimo osobine uređenja:

(a) *Refleksivnost* sledi iz činjenice da je $u = ue$, za svaku reč $u \in X^*$.

(b) *Antisimetričnost:* Neka su $u, v \in X^*$ reči takve da važi $u \leqslant_p v$ i $v \leqslant_p u$. To znači da je $v = up$ i $u = vq$, za neke reči $p, q \in X^*$, pa je $u = vq = upq$. Na osnovu svojstva jednakosti reči, iz $u = upq$ sledi da mora biti $pq = e$, što dalje povlači da je $p = q = e$, odakle je $u = v$.

(c) *Tranzitivnost:* Neka je $u \leqslant_p v$ i $v \leqslant_p w$.

To znači da je $v = up$ i $w = vq$, za neke reči $p, q \in X^*$, odakle je $w = vq = upq$. Prema tome, $u \leqslant_p w$.

Time smo kompletirali dokaz. □

Napomena 1. *Na sličan način pokazujemo da su $\leqslant_s$, $\leqslant_f$ relacije poretka (uređenja) na X^*, i ove relacije nazivamo:*

- $\leqslant_p$ – *prefiks uređenje;*
- $\leqslant_s$ – *sufiks uređenje;*
- $\leqslant_f$ – *faktor uređenje.*

Zadatak 1.10. *Dokazati da za proizvoljne reči $u, v, w \in X^*$ važi*

$$u \leqslant_p w \wedge v \leqslant_p w \quad \Rightarrow \quad u \leqslant_p v \vee v \leqslant_p u\,.$$

Rešenje: Napišimo reč w u obliku $w = x_1 x_2 \ldots x_n$, za neki prirodan broj $n \in \mathbb{N}$ i slova $x_1, x_2, \ldots, x_n \in X$.

Tada $u \leqslant_p w$ i $v \leqslant_p w$ znači da je $u = x_1 \ldots x_i$ i $v = x_1 \ldots x_j$, za $i, j \in \{1, 2, \ldots, n\}$. Dakle, ako je $i \leqslant j$, imamo da je $u \leqslant_p v$, a ako je $j \leqslant i$, onda je $v \leqslant_p u$. □

Leksikografsko uređenje

Neka je alfabet X linearno uređen nekim uređenjem $\leqslant$.

Tada se to uređenje može proširiti do linearnog uređenja $\leqslant_l$ na X^*, koje nazivamo *leksikografsko uređenje*, na sledeći način:

$$u \leqslant_l v \;\Leftrightarrow\; u \leqslant_p v \;\text{ ili }\; \begin{array}{l} u = pxq,\ v = pyr,\ \text{za } x < y \text{ u } X, \\ \text{gde su } p,q,r \in X^* \text{ i } x,y \in X \end{array}$$

Zadatak 1.11. *Relacija $\leqslant_l$ je linearno uređenje na X^*. Dokazati.*

Rešenje: Dokažimo najpre da je $\leqslant_l$ relacija poretka (uređenje):

(1) *Refleksivnost:* Za proizvoljnu reč $u \in X^*$ je $u \leqslant_l u$, jer je $u \leqslant_p u$.

(2) *Antisimetričnost:* Za reči $u,v \in X^*$ neka je $u \leqslant_l v$ i $v \leqslant_l u$.

(2.1) Ako je $u \leqslant_p v$ i $v \leqslant_p u$, tada je $u = v$, zbog antisimetričnosti prefiks uređenja.

(2.2) Neka je $u \leqslant_p v$ i $v = pxq$, $u = pyr$, za $x < y$ u X i neke $p,q,r \in X^*$.

Kako je p najduži zajednički prefiks za reči u i v i $u \leqslant_p v$, to je $p = u$, što je u suprotnosti sa pretpostavkom da je $u = pyr$, za $y \in X$.

Dakle, zaključujemo da slučaj (2.2) nije moguć.

(2.3) Neka je $v \leqslant_p u$ i $u = pxq$, $v = pyr$, za $x < y$ u X i neke reči $p,q,r \in X^*$.

Na isti način dokazujemo da ni ovaj slučaj nije moguć.

(2.4) Neka je $x_1 < y_1$, gde je x_1, y_1 prvi par različitih slova koja se nalaze na istoj poziciji u u i v, i neka je $y_2 < x_2$, gde je y_2, x_2 prvi par različitih slova na istoj poziciji u v i u.

Tada je $x_1 = x_2$ i $y_1 = y_2$, što istovremeno daje $x_1 < y_1$ i $y_1 < x_1$, a to nije moguće.

Prema tome, ni slučaj (2.4) nije moguć.

(3) *Tranzitivnost:* Neka su $u,v,w \in X^*$ reči takve da je $u \leqslant_l v$ i $v \leqslant_l w$.

(3.1) Neka je $u \leqslant_p v$ i $v \leqslant_p w$. Tada je $u \leqslant_p w$, zbog tranzitivnosti prefiks uređenja, pa je $u \leqslant_l w$.

(3.2) Neka je $u \leqslant_p v$ i $v = pxq$, $w = pyr$, za $x < y$ u X i neke reči $p,q,r \in X^*$.

Kako u ovom slučaju važi da je $u \leqslant_p v$ i $p \leqslant_p v$, to dobijamo da je $u \leqslant_p p$ ili $p \leqslant_p u$.

Ako je $u \leqslant_p p$, tada, obzirom da je $p \leqslant_p w$, imamo da je $u \leqslant_p w$. Dakle $u \leqslant_l w$, što je i trebalo dokazati.

Neka je sada $p \leqslant_p u$. Kako je slučaj $p = u$ obuhvaćen prethodnim slučajem $u \leqslant_p p$, to možemo uzeti da je $p <_p u$, tj. da je p pravi prefiks od u.

U tom slučaju imamo da je $u = pxq'$, za neku reč $q \in X^*$, što zajedno sa $w = pyr$ i $x < y$ daje $u \leqslant_l w$.

(3.3) Neka je $u = pxq$ i $v = pyr$, za $x < y$ u X i $p, q, r \in X^*$ i neka je $v \leqslant_p w$.

Tada, na potpuno isti način kao u (3.2), dokazujemo da je $u \leqslant_l w$.

(3.4) Neka je $u = p_1x_1q_1$ i $v = p_1y_1r_1$, za reči $p_1, q_1, r_1 \in X^*$ i slova $x_1, y_1 \in X$, takva da je $x_1 < y_1$, i neka je $v = p_2x_2q_2$, $w = p_2y_2r_2$, za neke $p_2, q_2, r_2 \in X^*$ pri čemu je $x_2 < y_2$ u X.

Kako je $p_1 \leqslant_p v$ i $p_2 \leqslant_p v$, to imamo da je $p_1 \leqslant_p p_2$ ili $p_2 \leqslant_p p_1$.

Oba slučaja razmatraju na sličan način, pa bez umanjenja opštosti možemo pretpostaviti da je $p_1 \leqslant_p p_2$.

Pretpostavimo najpre da je $p_1 = p_2$.

Tada je $y_1 = x_2$ i $x_1 < y_1 = x_2 < y_2$. Kako je $u = p_1x_1q_1$, $w = p_1y_2r_2$ i $x_1 < y_2$ zaključujemo da je $u \leqslant_l w$.

Neka je, sada, $p_1 <_p p_2$.

Iz $v = p_1y_1r_1$, $v = p_2x_2q_2$ i $p_1 <_p p_2$ zaključujemo da je $p_2 = p_1y_1s$, za neku reč $s \in X^*$, odakle sledi da je $w = p_1y_1t$, za neku reč $t \in X^*$.

Prema tome, $u = p_1x_1q_1$ i $w = p_1y_1t$, uz uslov $x_1 < y_1$, odakle sledi da je $u \leqslant_l w$.

Ovim je dokazana tranzitivnost relacije $\leqslant_l$, a time i da je $\leqslant_l$ uređenje.

(4) *Linearnost:* Neka su date proizvoljne reči $u, v \in X^*$.

Ako u i v nemaju zajednički prefiks, to znači da im se razlikuju već prva slova. Neka je x prvo slovo reči u i y je prvo slovo reči v.

Kako je, prema pretpostavci, alfabet X linearno uređen, to je $x < y$ ili je $y < x$, što znači da je $u <_l v$ ili $v <_l u$.

Dalje, pretpostavimo da u i v imaju zajednički prefiks. Označimo sa p najduži zajednički prefiks reči u i v.

Dakle važi $u = pxq$ i $v = pyr$, za neke $q, r \in X^*$ i slova $x, y \in X$ takva da je $x \neq y$, pa opet na osnovu linearnosti uređenja na alfabetu X zaključujemo da je $x < y$, i u tom slučaju je $u <_l v$, ili je $y < x$ kada je $v <_l u$.

Ovim je dokaz završen. □

Zadatak 1.12. *Urediti leksikografski sledeće binarne reči:*

$$u = 01000001, \quad v = 00110111, \quad w = 00111111\,.$$

Rešenje: Prva pozicija na kojoj se reč u razlikuje od v i w je pozicija 2.

Pri tome, na poziciji 2 reč u ima slovo 1, a reči v i w slovo 0, pa kako je $0 < 1$, to dobijamo da je $v <_l u$ i $w <_l u$.

Dalje, prva pozicija na kojoj se razlikuju od reči v i w je pozicija 5.

Na poziciji 5 reč v ima slovo 0, a reč w slovo 1, odakle je $v <_l w$. □

Napomena 2. *Binarne reči iz prethodnog zadatka su ASCII kodovi alfanumeričkih simbola* **A**, **7** *i* **?**, *tim redom.*

Zadatak 1.13. *Urediti leksikografski sve binarne reči dužine* 4.

Rešenje: Sve binarne reči dužine 4 su:

$$\begin{array}{cccccccc} 0000 & 0001 & 0010 & 0011 & 0100 & 0101 & 0110 & 0111 \\ 1000 & 1001 & 1010 & 1011 & 1100 & 1101 & 1110 & 1111 \end{array}$$

i na ovaj način su leksikografski uređene. □

Alfabetsko uređenje

Neka je alfabet X linearno uređen nekim uređenjem $\leqslant$.

Tada se uređenje $\leqslant_a$ na X^*, koje nazivamo *alfabetsko uređenje*, definiše na sledeći način:

$$u \leqslant_a v \Leftrightarrow |u| < |v| \text{ ili } \Big(|u| = |v| \text{ i } u \leqslant_l v\Big).$$

Zadatak 1.14. *Relacija $\leqslant_a$ je linearno uređenje na X^*. Dokazati.*

Rešenje: Dokažimo da je $\leqslant_a$ linearno uređenje.

(1) Refleksivnost: Za proizvoljnu reč $u \in X^*$ je $u \leqslant_a u$, jer je $|u| = |u|$ i $u \leqslant_l u$.

(2) Antisimetričnost: Neka je $u \leqslant_a v$ i $v \leqslant_a u$, za neke reči $u, v \in X^*$. Ako je $|u| = |v|$, tada imamo da je $u \leqslant_l v$ i $v \leqslant_l u$, odakle je $u = v$, zbog antisimetričnosti leksikografskog uređenja.

Sa druge strane, slučaj $|u| \neq |v|$ nije moguć, jer bi u suprotnom dobili da je $|u| < |v|$ i $|v| < |u|$.

Prema tome, zaključujemo da je $\leqslant_a$ antisimetrična relacija.

(3) Tranzitivnost: Neka je $u \leqslant_a v$ i $v \leqslant_a w$, za neke reči $u, v, w \in X^*$.

(3.1) Ako je $|u| < |v|$ i $|v| < |w|$, tada je $|u| < |w|$, pa je $u \leqslant_a w$.

(3.2) Ako je $|u| < |v|$, $|v| = |w|$ i $v \leqslant_l w$, tada je $|u| < |w|$, odakle sledi da je $u \leqslant_a w$.

(3.3) Ako je $|u| = |v|$, $u \leqslant_l v$ i $|v| < |w|$, tada je opet $|u| < |w|$, odakle je $u \leqslant_a w$.

(3.4) Neka je $|u| = |v|$, $u \leqslant_l v$ i $|v| = |w|$ i $v \leqslant_l w$.

Tada dobijamo da je $|u| = |w|$ i $u \leqslant_l w$, zbog tranzitivnosti leksikografskog uređenja, odakle sledi da je $u \leqslant_a w$.

Ovim smo dokazali tranzitivnost relacije $\leqslant_a$.

(4)Linearnost: Neka su date proizvoljne reči $u, v \in X^*$.

Ako je $|u| \neq |v|$, tada je $|u| < |v|$i u tom slučaju je $u \leqslant_a v$, ili je $|v| < |u|$ kada je $v \leqslant_a u$.

Ako je $|u| = |v|$, tada iz linearnosti leksikografskog uređenja sledi da je $u \leqslant_l v$ ili $v \leqslant_l u$, što zajedno sa $|u| = |v|$ daje $u \leqslant_a v$ ili $v \leqslant_a u$. □

Zadatak 1.15. *Urediti leksikografski i alfabetski sve binarne reči dužine manje ili jednake* 3.

Rešenje: Leksikografski poredak je:

$$\begin{array}{ccccccc} 0 & 00 & 000 & 001 & 01 & 010 & 011 \\ 1 & 10 & 100 & 101 & 11 & 110 & 111 \end{array}$$

Iste reči mogu se urediti i na sledeći način:

$$\begin{array}{cccccccc} 0 & 1 & 00 & 01 & 10 & 11 & & \\ 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \end{array}$$

i to je *alfabetski poredak*. □

Zadatak 1.16. *Počev od najmanjeg, pa do najvećeg, leksikografski urediti sledeće binarne reči:*

$$A = 01001011,\ B = 00101010,\ C = 01100100,$$
$$D = 01101111,\ E = 01000101.$$

Rešenje: Kako sve ove reči imaju isto prvo slovo, to razmatramo drugo slovo.

Jedino reč B ima drigo slovo 0, dok sve ostale imaju drugo slovo 1. Prema tome, B je najmanji element u ovom skupu.

Od preostalih reči, A i E imaju treće slovo 0, pa su manje od C i D, koje kao treće slovo imaju 1.

Ako dalje uporedimo $A = 01001011$ i $E = 01000101$, videćemo da se prvo slovo po kome se razlikuju na petoj poziciji, gde kod A stoji 1, a kod E stoji 0.

Dakle, reč E je druga, a A treća po veličini.

Konačno, prvo slovo po kome se reč $C = 01100100$ razlikuje od reči $D = 01101111$ je na petoj poziciji, gde kod C stoji 0, a kod D stoji 1.

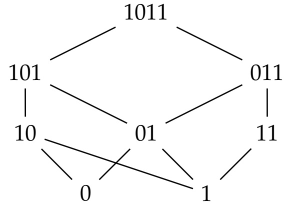
Prema tome, reč C je četvrta a D peta po veličini.

Dakle, traženi poredak je $BEACD$. □

Zadatak 1.17. *Neka je* $\preccurlyeq$ *uređenje na skupu binarnih reči*

$$X = \{0, 1, 10, 01, 11, 101, 011, 1011\}$$

zadato sledećim Haseovim dijagramom.



Koje od sledećih uređenja ima $\preccurlyeq$ *kao svoju restrikciju na skupu X:*

(a) *prefiks uređenje*
(b) *leksikografsko uređenje*
(c) *faktor uređenje*
(d) *alfabetsko uređenje*
(e) *sufiks uređenje*

Rešenje: Primetimo najpre da uređenje $\preccurlyeq$ nije linearno, jer, na primer, elementi 0 i 1 nisu uporedivi.

Kako znamo da leksikografsko i alfabetsko uređenje jesu linearna uređenja, to zaključujemo da $\preccurlyeq$ nije jedno od njih.

Ako bi $\preccurlyeq$ bilo prefiks uređenje, onda ne bi moglo da bude $0 \preccurlyeq 10$, a ako bi to bilo sufiks uređenje, onda ne bi moglo da bude $1 \preccurlyeq 01$.

Odavde zaključijemo da $\preccurlyeq$ ne može biti ni jedno od ta dva uređenja. Dakle, ostaje samo mogućnost da $\preccurlyeq$ jeste faktor uređenje.

To zaista važi, jer je bilo koja reč iz datog skupa manja od neke druge ako i samo ako je njen faktor. Prema tome, $\preccurlyeq$ je faktor uređenje. □

Zadatak 1.18. *U odnosu na koje uređenje su poređani sledeći nizovi:*

$$11,\ 101,\ 011,\ 01,\ 0.$$

(a) *prefiks uređenje*
(b) *leksikografsko uređenje*
(c) *faktor uređenje*
(d) *alfabetsko uređenje*
(e) *simetrično uređenje*

Rešenje: U zavisnosti od toga da li su ove reči date u rastućem poretku (od najmanjeg ka najvećem) ili opadajućem poretku (od najvećeg ka najmanjem), imamo da je ili $011 \preccurlyeq 101$ ili $101 \preccurlyeq 011$.

Odatle zaključujemo da se ne radi o prefiks uređenju, jer nijedna od te dve reči nije prefiks one druge.

Na isti način zaključujemo i da se ne radi o faktor uređenju, jer nijedna od te dve reči nije faktor druge.

Simetrično uređenje ne postoji, pa i tu mogućnost isključujemo.

Prema tome, preostaju mogućnosti da je $\preccurlyeq$ alfabetsko ili leksikografsko uređenje.

Kako se kod alfabetskog uređenja reči uređuju najpre po dužini, a potom se reči iste dužine uređuju leksikografski, to zaključujemo da $\preccurlyeq$ nije ni alfabetsko uređenje, jer dati poredak ne uvažava dužinu reči.

Preostaje, dakle, da $\preccurlyeq$ jeste leksikografsko uređenje. Ako date nizove poređamo po leksikografskom poretku, od najvećeg ka najmanjem, videćemo da je to upravo dati poredak.

To znači da je rešenje (b) - leksikografsko uređenje. □

1.2. Formalne gramatike

Pod *formalnom gramatikom*, ili kraće *gramatikom*, podrazumeva se trojka $G = (V,X,\pi)$ za koju važi:

- V je konačan skup koji nazivamo *rečnikom gramatike* G;
- $X \subseteq V$ je neprazan skup koji nazivamo *terminalnim alfabetom*;
- $\pi \subseteq (V \setminus X)^+ \times V^*$ je konačan skup koji nazivamo *pravilima gramatike* G.

Skup $V \setminus X$ nazivamo *pomoćnim alfabetom*, a njegove elemente *pomoćnim simbolima*.

Znači, rečnik V se sastoji iz dva disjunktna dela: $V = X \cup (V \setminus X)$.

Da bi pojednostavili pisanje, kao zamenu za izraz $(u,v) \in \pi$ koristićemo izraz $u \to v$.

Za reč $w' \in V$ kažemo da je neposredno izvodljiva iz reči $w \in V$, što označavamo sa $w \Rightarrow w'$, ako postoje $p,q \in V^*$ i pravilo $u \to v$ iz π, tako da je $w = puq$ i $w' = pvq$.

Dakle, reč w' je *neposredno izvedena* iz reči w ako postoji pravilo $u \to v$ iz π takvo da je u podreč od w, a reč w' je dobijena iz w tako što smo podreč u u w zamenili sa v.

Reč $w' \in V^*$ je *izvodljiva* iz reči $w \in V^*$, što označavamo sa $w \overset{*}{\Rightarrow} w'$ ako je ili $w = w'$ ili postoji niz $w_1, w_2, \ldots, w_n \in V^*$, gde je $n \geqslant 2$, takav da važi

$$w = w_1 \Rightarrow w_2 \Rightarrow \ldots \Rightarrow w_n = w'.$$

U tom slučaju, niz $w_1, w_2, \ldots, w_n$ nazivamo *izvođenjem* reči w' iz w. Za pomoćni simbol $\sigma \in V \setminus X$, skup $L(G,\sigma) = \{w \in X^* \mid \sigma \overset{*}{\Rightarrow} w\}$ nazivamo jezikom generisanim gramatikom G polazeći od simbola σ.

Za jezik $L \subseteq X^*$ kažemo da je *generisan gramatikom*, ili da je jezik *tipa* 0, ako postoji gramatika $G = (V,X,\pi)$ i pomoćni simbol $\sigma \in V \setminus X$ tako da je $L = L(G,\sigma)$.

Zadatak 1.19. *Neka je data gramatika $G = (V,X,\pi)$. Dokazati da tada važi:*

(i) $\Rightarrow$ *je saglasno zatvorenje od π (najmanja saglasna relacija na V^* koja sadrži π);*

(ii) $\overset{*}{\Rightarrow}$ *je polu-kongruencija na V^* generisana sa π (najmanje saglasno kvazi-uređenje na V^* koje sadrži π).*

Rešenje: Neka su reči $u,v \in X^*$ i pravilo $u \to v$ iz π (što možemo zapisati kao $(u,v) \in \pi$). Po definiciji relacije izvođenja, iz $u = eue$ i $v = eve$ zaključujemo da je $u \Rightarrow v$, tj. da relacija $\Rightarrow$ sadrži π. Po definiciji je očigledno da je relacija izvodjenja saglasna, pa za $u,v \in X^*$, uslov $u \Rightarrow v$ povlači $uw \Rightarrow vw$ i $wu \Rightarrow wv$, za proizvoljnu reč $w \in X^*$.

Posmatrajmo proizvoljnu saglasnu relaciju ϱ na X^* koja sadrži π i neka važi $w \Rightarrow w'$. To znači da postoje $p,q \in X^*$ takvi da je $w = puq$ i $w' = pvq$ za neke $(u,v) \in \pi \subseteq \varrho$. Zbog saglasnosti relacije ϱ važi da su $(w,w') \in \varrho$, odakle je očito $\Rightarrow$ sadržana u ϱ, čime je tvrđenje (i) dokazano.

Da bi dokazali tvrđenje (ii) dovoljno je da pokažemo tranzitivnost relacije $\overset{*}{\Rightarrow}$. Posmatrajmo reči $w, w', w'' \in X^*$ takve da je $w \overset{*}{\Rightarrow} w'$ i $w' \overset{*}{\Rightarrow} w''$. To znači da važi jedan od slučajeva:

(1) $w = w'$ i $w' = w''$, odakle je $w = w''$, što, prema definiciji relacije $\overset{*}{\Rightarrow}$, znači da $w \overset{*}{\Rightarrow} w''$.

(2) $w = w'$ i postoji niz $w_1, w_2, \dots, w_n \in V^*$, gde je $n \geqslant 2$, takav da važi

$$w = w' = w_1 \Rightarrow w_2 \Rightarrow \dots \Rightarrow w_n = w'', \text{ tj. } w \overset{*}{\Rightarrow} w''.$$

(3) $w' = w''$ i postoji niz $w_1, w_2, \dots, w_n \in V^*$, gde je $n \geqslant 2$, takav da važi

$$w = w_1 \Rightarrow w_2 \Rightarrow \dots \Rightarrow w_n = w' = w'', \text{ pa je } w \overset{*}{\Rightarrow} w''.$$

(4) Postoje nizovi $w_1, w_2, \dots, w_n \in V^*$, $w'_1, w'_2, \dots, w'_m \in V^*$ gde su $n, m \geqslant 2$, takvi da važi

$$w = w_1 \Rightarrow w_2 \Rightarrow \dots \Rightarrow w_n = w' \text{ i } w' = w'_1 \Rightarrow w'_2 \Rightarrow \dots \Rightarrow w'_m = w'',$$

odnosno

$$w \overset{*}{\Rightarrow} w' \overset{*}{\Rightarrow} w'' \qquad \Leftrightarrow \qquad w \overset{*}{\Rightarrow} w''.$$

Jasno je da je relacija izvođenja tranzitivna, te je ona najmanja polukongruencija na V^* generisana sa π. Ovim je dokaz završen. □

Zadatak 1.20. *Neka je data gramatika $G = (V, X, \pi)$, neka su $u, v, w \in V^*$ i neka je $u \overset{*}{\Rightarrow} v$. Dokazati da tada postoje izvođenja*

$$uw \overset{*}{\Rightarrow} vw \quad i \quad wu \overset{*}{\Rightarrow} wv \tag{1.2}$$

za koja važi

(i) *) dužine izvođenja (1.2) nisu veće od dužine izvođenja $u \overset{*}{\Rightarrow} v$;*
(ii) *sva pravila koja se koriste u izvođenjima (1.2) nalaze se među pravilima koja se koriste u izvođenju $u \overset{*}{\Rightarrow} v$.*

Rešenje: Tvrđenja ćemo dokazati indukcijom po dužini izvođenja $u \overset{*}{\Rightarrow} v$.

Pretpostavimo najpre da je $u \overset{*}{\Rightarrow} v$ izvođenje dužine 1, tj. neposredno izvođenje. To znači da je $u = pu'q$ i $v = pv'q$, za neke $p, q \in V^*$ i neko pravilo $u' \to v'$ iz π. Tada imamo da je $uw = pu'(qw)$, $vw = pv'(qw)$, $wu = (wp)u'q$ i $wv = (wp)v'q$, odakle dobijamo da $uw \Rightarrow vw$ i $wu \Rightarrow wv$.

Uzmimo dalje da je $u \overset{*}{\Rightarrow} v$ izvođenje dužine $n > 1$ i da tvrđenje važi za sva izvođenja dužine manje od n. Tada imamo da je $u \overset{*}{\Rightarrow} u' \overset{*}{\Rightarrow} v$ za neki $u' \in V^*$, pri čemu je $u' \overset{*}{\Rightarrow} v$ izvođenje dužine $n-1$, pa prema napred dokazanom i induktivnoj pretpostavci imamo da je

$$uw \Rightarrow u'w \overset{*}{\Rightarrow} vw \qquad \text{i} \qquad wu \Rightarrow wu' \overset{*}{\Rightarrow} wv,$$

pri čemu su izvođenja $u'w \overset{*}{\Rightarrow} vw$ i $wu' \overset{*}{\Rightarrow} wv$ dužine ne veće od $n-1$ i, takođe, izvođenja $uw \Rightarrow u'w$ i $wu \Rightarrow wu'$ su zasnovana na istom pravilu kao i $u \Rightarrow u'$, a izvođenja $uw \overset{*}{\Rightarrow} u'w$ i $wu \overset{*}{\Rightarrow} wu'$ su zasnovana na istim pravilima kao i kao i izvođenja $u' \overset{*}{\Rightarrow} v$. Ovim je dokaz upotpunjen. □

Zadatak 1.21. *Neka je data gramatika* $G = (V, X, \pi)$ *i neka za reči* $u_1, u_2, \dots u_n$; $v_1, v_2, \dots v_n \in V^*$, $n \in N$, *važi*

$$u_i \overset{*}{\Rightarrow} v_i \ \ \textit{za svaki} \ \ i \in \{1, 2, \dots n\}. \tag{1.3}$$

Dokazati da tada postoji izvođenje

$$u_1 u_2 \cdots u_n \overset{*}{\Rightarrow} v_1 v_2 \cdots v_n \tag{1.4}$$

za koje važi

(i) *dužina izvođenja* (1.4) *nije veća od zbira dužina izvođenja* (1.3);
(ii) *sva pravila koja se koriste u izvođenju* (1.4) *nalaze se među pravilima koja se koriste u izvođenjima* (1.3).

Rešenje: Tvrđenje zadatka biće dokazano indukcijom po n. Označimo sa l_i dužinu izvođenja $u_i \overset{*}{\Rightarrow} v_i$, $1 \leqslant i \leqslant n$.

Jasno je da tvrđenje zadatka važi za $n = 1$. Pretpostavimo da je $n > 1$ i da tvrđenje važi za sva izvođenja dužine $n-1$. Tada prema indukcijskoj pretpostavci dobijamo da postoji izvođenje

$$u_1 u_2 \cdots u_{n-1} \overset{*}{\Rightarrow} v_1 v_2 \cdots v_{n-1}, \tag{1.5}$$

čija dužina nije veća od $l_1 + l_2 + \cdots l_{n-1}$ i u kome se koriste samo pravila koja se koriste u izvođenjima $u_i \overset{*}{\Rightarrow} v_i$, $1 \leqslant i \leqslant n-1$. Sa druge strane, prema prethodnom zadatku, imamo da postoje izvođenja

$$u_1 \cdots u_{n-1} u_n \overset{*}{\Rightarrow} v_1 \cdots v_{n-1} u_n \quad \text{i} \quad v_1 \cdots v_{n-1} u_n \overset{*}{\Rightarrow} v_1 \cdots v_{n-1} v_n, \tag{1.6}$$

pri čemu dužina prvog ne prelazi dužinu izvođenja (1.5), odnosno ne prelazi $l_1 + l_2 + \cdots l_{n-1}$, a dužina drugog ne prelazi l_n, i takođe, među pravilima koja se koriste u prvom su samo pravila koja se koriste u (1.5), a među pravilima koja se koriste u drugom od izvođenja (1.6) se koriste samo pravila koja se koriste u izvođenju $u_n \overset{*}{\Rightarrow} v_n$. Dakle, iz (1.6) sledi da postoji izvođenje oblika (1.4) koje zadovoljava uslove (i) i (ii) zadatka. Ovim je dokaz završen. □

Zadatak 1.22. *Prazan jezik* $L = \emptyset$ *je jezik tipa 0. Dokazati.*

Rešenje: Gramatika koja generiše prazan jezik definiše se vrlo jednostavno. Ako je $X = \emptyset$, $V \setminus X = \{\sigma\}$ i jedino pravilo iz π dato sa $\sigma \to \sigma$, jednostavno se pokazuje da je $L(G, \sigma) = L = \emptyset$. □

Zadatak 1.23. *Jezik* $L = X$ *(*X *je proizvoljan alfabet) je jezik tipa 0. Dokazati.*

Rešenje: Jednostavno se pokazuje da je, za neki alfabet X, jezik $L = X$, generisan gramatikom $G = (V, X, \pi)$, u kojoj je $V \setminus X = \{\sigma\}$ i skup pravila izvođenja je $\pi = \{\sigma \to x \mid x \in X\}$. $\square$

Zadatak 1.24. *Dokazati da je jezik $L = X^*$ jezik tipa 0.*

Rešenje: Definišimo formalnu gramatiku $G = \{V, X, \pi)$, u kojoj je X dati alfabet, $V \setminus X = \{\sigma\}$, a pravila su data sa $\pi = \{\sigma \to \sigma\lambda\} \cup \{\lambda \to x \mid x \in X\} \cup \{\sigma \to e\}$. Dokazaćemo da je $X^* = L(G, \sigma)$.

Posmatrajmo proizvoljnu reč $w \in X^*$. Indukcijom po dužini reči w pokazaćemo da $w \in L(G, \sigma)$.

Ako je $w = e$, na osnovu pravila $\sigma \to e$, direktno sledi da $w \in L(G, \sigma)$. Pretpostavimo da za svaku reč dužune $|w| = n-1$ važi da $w \in L(G, \sigma)$. Dokažimo da tvrđenje važi za reč w dužine n. Tada je reč w oblika $w = w'x$, za neki $x \in X$ i $|w'| = n-1$.

Prema induktivnoj pretpostavci i Zadatku 1.20. postoji izvođenje $\sigma \Rightarrow^* w'$ i važi

$$\sigma \Rightarrow \sigma\lambda \overset{*}{\Rightarrow} w'\lambda \overset{*}{\Rightarrow} w'x = w,$$

pa je $w \in L(G, \sigma)$, odnosno $X^* \subseteq L(G, \sigma)$.

Indukcijom po dužini izvođenja dokazaćemo obratnu inkluziju. Jedino izvođenje dužine jedan je $\sigma \to e$ i $e \in X^*$, pa tvrđenje važi. Reči dužine jedan (slova iz X) dobijamo samo u sledećem izvođenju dužine tri

$$\sigma \Rightarrow \sigma\lambda \Rightarrow e\lambda \Rightarrow ex = x,$$

za proizvoljan $x \in X$. Pretostavimo da tvrđenje važi za sva izvođenja dužine $n > 3$ (kada dobijamo reči iz X^* dužine $n-2$) i dokažimo da važi ako je izvođenje dužine $n+2$. Neka je $w \in L(G, \sigma)$. Tada je u izvođenju

$$\sigma \Rightarrow w_1 \Rightarrow w_2 \cdots \Rightarrow w_n \Rightarrow w_{n+1} \Rightarrow w_{n+2} = w.$$

prvi korak je sigurno $\sigma \Rightarrow \sigma\lambda$, te je $w_1 = \sigma\lambda$. Kako se niz ne zaustavlja, to u narednom koraku imamo izvođenje $\lambda \Rightarrow x$ i $w_2 = \sigma x$, pri čemu je, očigledno, $x \in X$ poslednje slovo reči w, tj. možemo pisati $w = w'x$ i $w' \in L(G, \sigma)$ se može dobiti u $n-2$ koraka polazeći od σ. Primenom indukcijske pretpostavke dobijamo $w' \in X^*$, pa je jasno da i $w = w'x \in X^*$, tj. $L(G, \sigma) \subseteq X^*$. Dakle, X^* je jezik tipa 0 generisan gramatikom $L(G, \sigma)$. $\square$

Zadatak 1.25. *Neka je data gramatika $G = (V, X, \pi)$, gde je $X = \{x\}$, $V \setminus X = \{\sigma\}$ i pravila su data sa $\sigma \to x\sigma$, $\sigma \to e$. Tada je $L(G, \sigma) = \{x^n \mid n \in \mathbb{N}^0\}$. Dokazati.*

Rešenje: Indukcijom po $n \in \mathbb{N}^0$ dokazuje se da je

$$L = \{x^n \mid n \in \mathbb{N}^0\} \subseteq L(G, \sigma).$$

Za $n = 0$ direktno iz pravila $\sigma \to e = x^0$ sledi da $x^0 \in L(G, \sigma)$. Za $n = 1$ imamo izvođenje

$$\sigma \Rightarrow x\sigma \Rightarrow x,$$

te tvrđenje i u ovom slučaju važi. Pretpostavimo da važi za reči dužine n i dokažimo za reči dužine $n+1$. Neka je $w = x^{n+1} \in L$, odnosno $w = xx^n$. Prema indukcijskoj pretpostavci postoji izvođenje $\sigma \overset{*}{\Rightarrow} x^n$, odakle dobijamo

$$\sigma \Rightarrow x\sigma \overset{*}{\Rightarrow} xx^n = x^{n+1},$$

pa je jasno da je $L = \{x^n \mid n \in \mathbb{N}^0\} \subseteq L(G,\sigma)$.

Indukcijom po dužini n izvođenja dokazaćemo da je $L(G,\sigma) \subseteq L$. Za $n = 1$ imamo izvođenje $\sigma \to e = x^0 = x^{1-1}$ i $x^0 \in L$. Pretpostavimo da naše tvrđenje važi za neko izvođenje dužine n (kada dobijamo reč oblika X^{n-1}) i dokažimo da važi za izvođenja dužine $n+1$. Uzmimo da je $w \in L(G,\sigma)$. To znači da postoji niz $w_1, w_2, \cdots w_{n+1} \in V^*$, za $n \geqslant 1$, takav da

$$\sigma \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n \Rightarrow w_{n+1} = w.$$

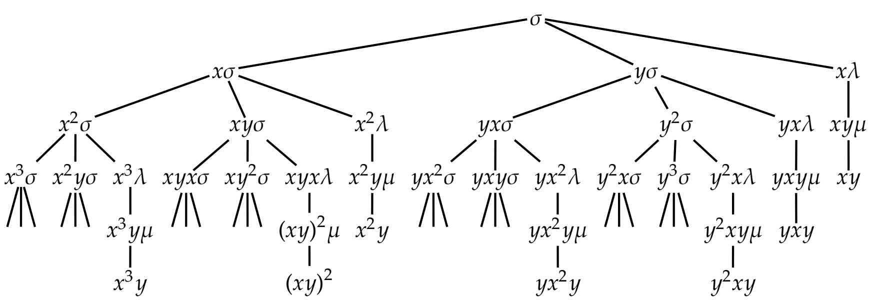
Prvi korak u ovom izvođenju svakako je $\sigma \Rightarrow x\sigma$, tj. $w_1 = x\sigma$. Prema Zadatku 1.20. postoji izvođenje $\sigma \Rightarrow x\sigma \overset{*}{\Rightarrow} xw' = w$ i kako je $\sigma \overset{*}{\Rightarrow} w'$ je dužine n, pa prema pretpostavci $w' \in L$, odnosno $w' = x^{n-1}$, odakle diretno dobijamo da $w = xw' = x^n \in L$. Ovim je dokaz kompletiran. □

Zadatak 1.26. *Neka je data gramatika $G = (V, X, \pi)$, gde je $X = \{x, y\}$, skup pomoćnih simbola $V \setminus X = \{\sigma, \lambda, \mu\}$ i pravila su data sa*

$$\sigma \to x\sigma, \ \ \sigma \to y\sigma, \ \ \sigma \to x\lambda, \ \ \lambda \to y\mu, \ \ \mu \to e.$$

Dokazati da je $L(G,\sigma) = X^*xy$.

Rešenje: Skupu pravila izvođenja pridružićemo korensko stablo čiji su čvorovi označeni terminalnim i neterminalnim simbolima. Koren stabla označićemo početnom simbolom σ.



Vidimo da za svaku reč $w \in X^*$ postoji izvođenje

$$\sigma \overset{*}{\Rightarrow} w\sigma \Rightarrow wx\lambda \Rightarrow wxy\mu \Rightarrow wxy,$$

pa je $X^*xy \subseteq L(G,\sigma)$. Dokazaćemo da važi i obratna inkluzija, tj. da je jezik generisan datom gramatikom $L(G,\sigma) = X^*xy$.

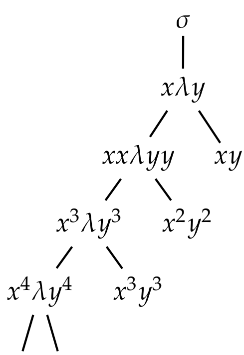
Uzmimo da je $w \in L(G,\sigma)$, tj. da je $\sigma \overset{*}{\Rightarrow} w$. To znači da postoji niz $w_1, w_2, \cdots w_n \in V^*$, gde je $n \geqslant 2$, takav da

$$\sigma \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_{n-1} \Rightarrow w_n = w.$$

Kako je pravilo $\mu \to e$ jedino koje ne sadrži pomoćni simbol sa desne strane, to dobijamo da je $w_{n-1} = w\mu$. Dalje, $\lambda \to y\mu$ je jedino pravilo u kome se μ javlja na desnoj strani, pa je $w = w'y$, za neki $w' \in X^*$, i $w_{n-2} = w'\lambda$. Slično, pravilo $\sigma \to x\lambda$ je jedino u kome se λ javlja sa desne strane, pa je $w' = w''x$, za neki $w'' \in X^*$ i $w_{n-3} = w''\sigma$. Prema tome, $w = w'y = w''xy \in X^*xy$, što znači $L(G,\sigma) \subseteq X^*xy$. Dakle, $L(G,\sigma) = X^*xy$. □

Zadatak 1.27. *Neka je $G = (V, X, \pi)$, gde je $X = \{x, y\}$, $V \setminus X = \{\sigma, \lambda\}$ i neka su pravila iz π data sa $\sigma \to x\lambda y$, $\lambda \to x\lambda y$, $\lambda \to e$. Tada je $L(G,\sigma) = \{x^n y^n \mid n \in N\}$. Dokazati.*

Rešenje: Izvođenja za datu gramatiku možemo predstaviti stablom



Za proizvoljnu reč $w = x^n y^n$ postoji izvođenje

$$\sigma \Rightarrow x\lambda y \Rightarrow xx\lambda yy \overset{*}{\Rightarrow} x^n\lambda y^n \Rightarrow x^n y^n,$$

što znači da je $\{x^n y^n \mid n \in N\} \subseteq L(G,\sigma)$.

Sa druge strane, neka je reč $w \in L(G,\sigma)$, tj. neka postoji izvođenje $\sigma \overset{*}{\Rightarrow} w$. To znači da postoji niz izvođenja

$$\sigma \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_{n-1} \Rightarrow w_n \Rightarrow w_{n+1} = w,$$

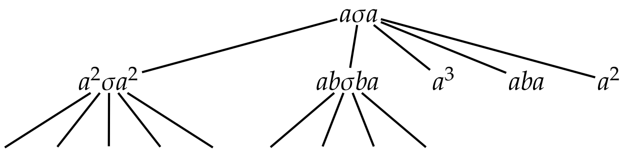
za neke $w_1, w_2, \cdots w_{n+1} \in V^*$ i $n \geqslant 2$. Kako je pravilo $\sigma \to x\lambda y$ jedino koje sadrži σ sa leve strane, to dobijamo da je $w_1 = x\lambda y$. Dalje, pravilo $\lambda \to x\lambda y$ je jedino previlo u kome se λ javlja sa leve strane i sadrži pomomoćni simbol sa desne strane, pa je $w_2 = x^2\lambda y^2$. Nastavljajući postupak zaključujemo da je $w_{n-1} = x^{n-1}\lambda y^{n-1}$. Jasno da je, odatle, $w_n = x^n\lambda y^n$, odnosno $w_{n+1} = x^n y^n$, jer je $\lambda \to e$ jedino pravilo koje ne sadrži pomoćni simbol sa desne strane.

Time smo dokazali da važi $L(G,\sigma) = \{x^n y^n \mid n \in N\}$. □

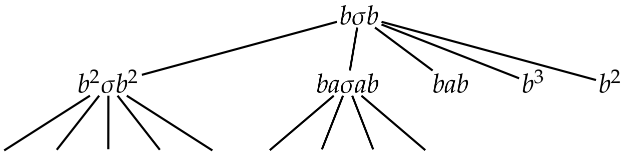
Zadatak 1.28. *Neka je data gramatika $G = (V, X, \pi)$, gde je $X = \{a, b\}$, $V \setminus X = \{\sigma\}$ i pravila iz π su data sa $\sigma \to a\sigma a$, $\sigma \to b\sigma b$, $\sigma \to a$, $\sigma \to b$, $\sigma \to e$. Dokazati da je tada $L(G,\sigma)$ skup svih palindroma (reči koje su jednake svojim reverznim rečima).*

Rešenje: Ako sa $\overline{w}$ označimo reč w čitanu sa kraja i sa $C = \{w \in X^* | w = \overline{w}\}$ označimo skup svih palindroma, pokazaćemo da je $L(G,\sigma) = C$.

Primetimo da sve reči koje nisu slova ulaznog alfabeta dobijamo izvođenjem iz $a\sigma a$ ili iz $b\sigma b$. Tako dobijamo dva podstabla:



i, dualno:



Indukcijom po dužini izvođenja dokazaćemo inkluziju $L(G,\sigma) \subseteq C$. Posmatrajmo proizvoljnu reč $w \in L(G,\sigma)$. Ako važi $\sigma \to w$ onda je $w \in \{a, b, e\} \subseteq C$. Pretpostavimo da tvrđenje važi za svako izvođenje $\sigma \overset{*}{\Rightarrow} w$ dužine n i dokažimo da važi ako se reč w može dobiti iz σ izvođenjem dužine $n+1$.

To znači da postoji izvođenje $\sigma \Rightarrow w_1 \overset{*}{\Rightarrow} w_{n+1} = w$. Jasno je da je, tada, $w_1 = a\sigma a$ ili $w_1 = b\sigma b$ i $w = w_{n+1} = aua$ (ili $w = w_{n+1} = bub$). Pri tome je izvođenje $\sigma \overset{*}{\Rightarrow} u$ dužine n te, na osnovu indukcijske pretpostavke, $u \in C$, tj. $u = \overline{u}$. Znači da je $\overline{w} = \overline{aua} = a\overline{u}a = aua = w$ (i analogno za $w = bub$), odnosno $w \in C$.

Sa druge strane, ako je $w \in C$ prazna reč ili slovo, onda postoje pravila izvođenja po kojima se w direktno dobija. Pretpostavimo da za proizvoljnu reč $w \in C$ dužine $|w| = n$ postoji izvođenje $\sigma \overset{*}{\Rightarrow} w$. Posmatrajmo reč $u \in C$ čija je dužina $n+2$. Kako je $u = \overline{u}$ prvo i poslednje slovo reči u moraju biti jednaki, pa je $u = awa$ ili $u = bwb$, za neku reč $w = \overline{w}$ za koju postoji izvođenje $\sigma \overset{*}{\Rightarrow} w$. Dakle imamo $\sigma \Rightarrow a\sigma a \overset{*}{\Rightarrow} awa = u$ (i analogno, $\sigma \Rightarrow b\sigma b \overset{*}{\Rightarrow} bwb = u$), tj. $u \in L(G,\sigma)$. Ovim je dokaz završen. □

Zadatak 1.29. *Neka je data gramatika $G = (V, X, \pi)$, gde je $X = \{a, b\}$, $V \setminus X = \{\sigma, \alpha, \beta\}$ i pravila iz π su data sa $\sigma \to a\beta$, $\sigma \to b\alpha$, $\alpha \to b\alpha\alpha$, $\alpha \to a\sigma$, $\alpha \to a$, $\beta \to a\beta\beta$, $\beta \to b\sigma$, $\beta \to b$. Tada je jezik generisan ovom gramatikom $L(G,\sigma)$ skup svih nepraznih reči nad azbukom $\{a, b\}$ u kojima se broj slova a poklapa sa brojem slova b. Dokazati.*

Rešenje: Označimo sa $|w|_x$ broj slova x u reči w ($x \in \{a, b, \alpha, \beta\}$). Pokazaćemo da je $L(G,\sigma) = \{w \in \{a, b\}^+ \mid |w|_a = |w|_b\}$.

Neka je $w \in L(G,\sigma)$ i neka je

$$\sigma \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_{n-1} \Rightarrow w_n = w$$

izvođenje reči w u gramatici G. Jasno je da je $w \neq e$. Indukcijom po n lako se pokazuje da je $|w_n|_a + |w_n|_\alpha = |w_n|_b + |w_n|_\beta$. Pošto $w \in L(G,\sigma)$ imamo da je $|w|_\alpha = |w|_\beta = 0$, pa je $|w|_a = |w|_b$.

Obratno, neka je reč $w \in \{a,b\}^+$ takva da je $|w|_a = |w|_b$ i $|w| = 2n$. Indukcijom po n pokazujemo da je $w \in L(G,\sigma)$. Ako je $n = 1$ onda je $w = ab$ ili $w = ba$. Kako u oba slučaja postoje izvođenja

$$\sigma \Rightarrow a\beta \Rightarrow ab \quad \text{i} \quad \sigma \Rightarrow b\alpha \Rightarrow ba,$$

jasno da $w \in L(G,\sigma)$. Pretpostavimo da je naše tvrđenje istinito za sve reči dužine ne veće od $2n$ i dokažimo da važi za reči dužine $2n+2$. Primetimo da je reč w oblika $a^i bu_1$ ili $b^i au_1$, za neki $i \geqslant 1$ i $u_1 \in \{a,b\}^*$. Razmatraćemo, bez umanjenja opštosti, slučaj kada je $w = a^i bu_1$, jer je dokaz u drugom slučaju analogan.

Ako je $i = 1$ onda važi $|u_1|_a = |u_1|_b$, pa prema indukcijskoj hipotezi postoji izvođenje $\sigma \overset{*}{\Rightarrow} u_1$. Tako dobijamo

$$\sigma \Rightarrow a\beta \Rightarrow ab\sigma \overset{*}{\Rightarrow} abu_1 = w,$$

tj. $w \in L(G,\sigma)$.

Neka je $i > 1$. Kako je $|a^i bu_1|_a = |a^i bu_1|_b$, to važi $|u_1|_a < |u_1|_b$, pa postoji prefiks v_1 od u_1 najmanje dužine takav da je $|v_1|_a < |v_1|_b$. Tada je jasno da je $v_1 = w_1 b$, pri čemu je $|w_1|_a = |w_1|_b$ i dobili smo $w = a^i bw_1 bu_2$. Ako je $i = 2$, stavimo da je $u_2 = w_2$. Za $i > 2$, na isti način zaključujemo da postoji prefiks v_2 od u_2 najmanje dužine, takav da je $|v_2|_a < |v_2|_b$, pa je $v_2 = w_2 b$ i $|w_2|_a = |w_2|_b$. Ponavljajući isti postupak i puta, dobijamo da je $w = a^i bw_1 bw_2 b \ldots bw_i$, gde je $|w_j|_a = |w_j|_b$, za sve $1 \leqslant j \leqslant i$. Prema indukcijskoj pretpostavci, za sve j, $1 \leqslant j \leqslant i$ važi $\sigma \overset{*}{\Rightarrow} w_j$, pa imamo izvođenje

$$\sigma \Rightarrow a\beta \Rightarrow aa\beta\beta \overset{*}{\Rightarrow} a^i\beta^i \Rightarrow a^i\beta\beta^{i-1} \Rightarrow a^i b\sigma\beta^{i-1} \overset{*}{\Rightarrow} a^i bw_1\beta^{i-1} \overset{*}{\Rightarrow} a^i bw_1 bw_2 b \ldots bw_i,$$

što znači da $w \in L(G,\sigma)$. Ovim je tvrđenje dokazano. □

Zadatak 1.30. *Neka je data gramatika $G = (V,X,\pi)$, gde je $X = \{a,b\}$, $V \setminus X = \{\sigma,\alpha,\beta\}$ i pravila iz π su data sa $\sigma \to a\beta$, $\sigma \to b\alpha\sigma$, $\sigma \to a\beta\sigma$, $\alpha \to b\alpha\alpha$, $\alpha \to a$, $\beta \to a\beta\beta$, $\beta \to b$. Naći jezik $L(G,\sigma)$ generisan ovom gramatikom.*

Rešenje: Induktivno se pokazuje da je jezik $L(G,\sigma)$ generisan ovom gramatikom ekvivalentan jeziku dobijenom u prethodnom zadatku, tj. da je $L(G,\sigma)$ skup svih reči w kod kojih je $|w|_a = |w|_b$. □

Zadatak 1.31. *Neka je data gramatika $G = (V,X,\pi)$, gde je $X = \{a,b,c\}$, $V \setminus X = \{\sigma,\alpha,\beta\}$ i pravila iz π su data sa $\sigma \to abc$, $\sigma \to a\alpha bc$, $\alpha b \to b\alpha$, $\alpha c \to \beta bcc$, $a\beta \to aa\alpha$,*

$b\beta \to \beta b$, $a\beta \to aa$. Ovom gramatikom je generisan jezik $L(G,\sigma) = \{a^n b^n c^n \mid n \geqslant 1\}$. Dokazati.

Rešenje: Dokažimo da postoji izvođenje $\sigma \overset{*}{\Rightarrow} a^n \alpha b^n c^n$ indukcijom po n .

Za $n = 1$ tvrđenje važi direktnom primenom pravila $\sigma \to a\alpha bc$ iz π. Pretpostavimo da tvrđenje važi za neki broj $n > 1$. Tada imamo

$$\sigma \overset{*}{\Rightarrow} a^n \alpha b^n c^n \overset{*}{\Rightarrow} a^n b^n \alpha c^n \Rightarrow a^n b^n \beta bcc^n \overset{*}{\Rightarrow} a^n \beta b^{n+1} c^{n+1} \Rightarrow a^n \alpha b^n c^n.$$

Tako, za proizvoljan broj $n \geqslant 1$ dobijamo

$$\sigma \overset{*}{\Rightarrow} a^n \alpha b^n c^n \overset{*}{\Rightarrow} a^n \beta b^{n+1} c^{n+1} \Rightarrow a^{n+1} b^{n+1} c^{n+1},$$

što znači da $a^n b^n c^n \in L(G,\sigma)$, za svaki $n \geqslant 1$.

Obratno, za proizvoljnu reč $w \in L(G,\sigma)$ postoji niz izvođenja

$$\sigma \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_{n-1} \Rightarrow w_k = w,$$

$w_1, w_2, \ldots w_k \in V^*$ i indukcijom po k može se pokazati da je

$$|w_k|_a + |w_k|_\beta = |w_k|_b = |w_k|_c.$$

Sa druge strane, sva izvođenja u gramatici G očuvavaju poredak reči (sva slova a su ispred svih slova b, koja su ispred svih slova c). Dakle, ako je reč $w \in L(G,\sigma)$, onda postoji broj $n \geqslant 1$ tako da je $w = a^n b^n c^n$, što je i trebalo dokazati. $\square$

Zadatak 1.32. *Neka je data gramatika $G = (V, X, \pi)$, gde je $X = \{a, b\}$, skup pomoćnih simbola $V \setminus X = \{\sigma, \alpha, \beta\}$ i neka su pravila iz π data sa $\sigma \to \alpha\beta$, $\alpha \to a\alpha$, $\alpha \to a$, $\beta \to b\beta$, $a\beta \to b$. Dokazati da ova gramatika generiše jezik $L(G,\sigma) = \{a^n b^m \mid n, m \geqslant 1\}$.*

Rešenje: Pogledaj prethodni zadatak. $\square$

Zadatak 1.33. *Data je gramatika $G = (V, X, \pi)$, gde je $X = \{a, b\}$, $V \setminus X = \{\sigma, \lambda\}$ i pravila iz π su $\sigma \to \lambda\lambda$, $\lambda \to \lambda\lambda\lambda$, $\lambda \to a$, $\lambda \to b\lambda$, $\lambda \to \lambda b$.*

(a) *Naći jezik $L(G,\sigma)$;*

(b) *Koje se reči jezika $L(G,\sigma)$ mogu dobiti izvođenjima koja imaju četiri ili više koraka?*

(c) *Za bilo koje $m, n, k \geqslant 0$ opisati izvođenja, u gramatici G, reči $b^m a b^n a b^k$.*

Rešenje: (a) Ako sa L označimo jezik $L = X^* a X^* a X^*$, indukcijom po dužini reči pokazujemo da je $L \subseteq L(G,\sigma)$, a indukcijom po dužini izvođenja da je $L(G,\sigma) \subseteq L$.

(b) Reči $\{a^i \mid i \geqslant 5\}$ i reči $\{u \in L \mid u \neq a^i$ i $|u| \geqslant 4\}$ mogu se dobiti izvođenjima od četiri ili više koraka. $\square$

Zadatak 1.34. *Data je gramatika $G = (V, X, \pi)$, gde je $X = \{x, y\}$, $V \setminus X = \{\sigma, \lambda\}$ i $\pi = \{\sigma \to x\lambda x, \sigma \to y\lambda y, \sigma \to e, \lambda \to \sigma\sigma\}$.*

(a) *Naći jezik* $L(G,\sigma)$*;*
(b) *Naći izvođenje reči baabbb u gramatici G.*

Rešenje: (a) Indukcijom, po dužini reči na jednu stranu, a po dužini izvođenja na drugu stranu, jednostavno se pokazuje da je $L(G,\sigma) = (x^*y^*)^* = X^*$. □

Zadatak 1.35. *Konstruisati formalnu gramatiku kojom je moguće opisati svaki aritmetički izraz sa tri promenljive, koji može da sadrži zagrade, pri čemu je bitan prioritet operacija.*

Rešenje: Azbuka kojom se predstavlja formalna gramatika je

$$V = \{a,b,c,+,-,*,/,(,)\}.$$

Definišimo skup pomoćnih simbola $V \setminus X = \{I,O,M\}$, gde I označava izraz, O operand i M množilac. Skup π pravila izvođenja biće:

$I \to O, I \to I+O, I \to I-O,$
$O \to M, O \to O*M, O \to O/M,$
$M \to a, M \to b, M \to c, M \to (I).$

Ova gramatika $G = (V,X,\pi)$ jednoznačno opisuje jezik $L(G,I)$ svih aritmetičkih izraza sa tri promenljive. □

Zadatak 1.36. *Konstruisati formalnu gramatiku kojom se može opisati svaki aritmetički izraz sa tri promenljive u kome je dopuštena i operacija korenovanja.*

Rešenje: Vidi prethodni zadatak. □

Zadatak 1.37. *Neka su* L_1 *i* L_2 *jezici generisani gramatikama* $G_1 = (V_1,X,\pi_1)$ *i* $G_2 = (V_2,X,\pi_2)$*, tim redom. Konstruisati gramatike koje generišu:*

(a) *uniju jezika* L_1 *i* L_2*, tj. jezik* $L = L_1 \cup L_2$*;*
(b) *proizvod ova dva jezika* $L = L_1L_2$*;*
(c) *jezik* $\overline{L_1}$*, koji sadrži sve reverzne reči jezika* L_1*;*
(d) *jezik* L_1^**.*

Rešenje: (a) Bez umanjenja opštosti možemo pretpostaviti da su skupovi pomoćnih simbola datih gramatika $L_1 = L(G_1,\sigma_1)$ i $L_2 = L(G_2,\sigma_2)$ disjunktni, tj. $(V_1 \setminus X) \cap (V_2 \setminus X) = \emptyset$. Konstruišimo gramatiku

$$G_U = (V_1 \cup V_2 \cup \{\sigma\}, X, \pi_1 \cup \pi_2 \cup \{\sigma \to \sigma_1 + \sigma_2\}).$$

Pokazaćemo da je $L(G_U,\sigma) = L = L_1 \cup L_2$.

Posmatrajmo reč $w \in L(G_U,\sigma)$. To znači da postoji izvođenje $\sigma \overset{*}{\Rightarrow} w$, pa imamo $\sigma \to \sigma_1 \overset{*}{\Rightarrow}_{G_1} w$ ili $\sigma \to \sigma_1 \overset{*}{\Rightarrow}_{G_2} w$, odnosno $w \in L_1$ ili $w \in L_2$, što znači da $w \in L_1 \cup L_2$.

Obratno, neka je $w \in L_1 \cup L_2$. Pretpostavimo da $w \in L_1$. To znači da postoji izvođenje $\sigma_1 \overset{*}{\Rightarrow}_{G_1} w$ i kako, po definiciji, postoji pravilo $\sigma \to \sigma_1$ u gramatici G_U, to imamo $\sigma \overset{*}{\Rightarrow}_{G_U} w$, pa $w \in L$ (analogno pokazujemo tvrđenje i za reč $w \in L_2$).

(b) Konstruišimo gramatiku

$$G_P = (V_1 \cup V_2 \cup \{\sigma\}, X, \pi_1 \cup \pi_2 \cup \{\sigma \to \sigma_1\sigma_2\}).$$

Pokazaćemo da je $L(G_P,\sigma) = L_1L_2 = L$.

Ako je $w \in L_1L_2$. Tada postoje reči $u \in L_1$ i $v \in L_2$ takve da je $w = uv$ i postoje izvođenja $\sigma_1 \overset{*}{\Rightarrow}_{G_1} u$ i $\sigma_2 \overset{*}{\Rightarrow}_{G_2} v$. To znači da postoji izvođenje

$$\sigma \to \sigma_1\sigma_2 \overset{*}{\Rightarrow}_{G_P} uv = w,$$

pa je $L \subseteq L(G_P,\sigma)$. Sa druge strane, ako je $w \in L(G_P,\sigma)$, onda postoji izvođenje $\sigma \to \sigma_1\sigma_2 \overset{*}{\Rightarrow}_{G_P} w$ i tada je $w = uv$, za neke reči u,v za koje je $\sigma_1 \overset{*}{\Rightarrow}_{G_1} u$ i $\sigma_2 \overset{*}{\Rightarrow}_{G_2} v$. Dakle, $u \in L_1$ i $v \in L_2$, tj. $w \in L$. Ovim smo pokazali da je $L(G_P,\sigma) = L$.

(c) Konstruišimo gramatiku $G = (V_1, X, \pi)$ u kojoj je $u \to v$ pravilo iz π ako i samo ako je $\overline{u} \to \overline{v}$ pravilo iz π_1 (za svaki pomoćni simbol α važi $\alpha = \overline{\alpha}$). Dokazaćemo da je $L = \overline{L_1} = L(G,\sigma_1)$. Za proizvoljnu reč $u \in L$ važi da $\overline{u} \in L_1$, pa je jasno da postoji izvođenje $\overline{\sigma_1} = \sigma_1 \overset{*}{\Rightarrow}_G u$.

Obratno, za svaku reč $u \in L(G,\sigma_1)$ postoji izvođenje $\sigma_1 \overset{*}{\Rightarrow}_G u$, što znači da postoji izvođenje $\sigma_1 \overset{*}{\Rightarrow}_{G_1} \overline{u}$. Jasno je da $\overline{u} \in L_1$, tj. $u \in L$. Ovim je tvrđenje dokazano.

(d) Uvedimo novi pomoćni simbol $\lambda \notin V_1 \setminus X$ i konstruišimo gramatiku

$$G = (V_1 \cup \lambda, X, \pi_1 \cup \{\lambda \to \lambda\sigma_1 + e\}.$$

Pokazaćemo da je $L = L_1^* = L(G,\lambda)$.

Proizvoljna reč $w \in L$ je ili prazna reč $w = e$ ili se w može napisati u obliku $w = w_1w_2\dots w_n$, gde $w_i \in L_1$, za $i = \overline{1,n}$. Ako je $w = e$ u skupu pravila izvođenja imamo $\lambda \to e$, tj. $w \in L(G,\lambda)$. U protivnom, za svaku reč w_i, za $i = \overline{1,n}$ postoji izviđenje $\sigma_1 \overset{*}{\Rightarrow}_{G_1} w_i$. Dakle,

$$\begin{aligned}\lambda \to \lambda\sigma_1 \overset{*}{\Rightarrow}_G \lambda w_n \Rightarrow_G \lambda\sigma_1 w_n \overset{*}{\Rightarrow}_G \lambda wn-1 w_n \Rightarrow_G \cdots \Rightarrow_G \\ \Rightarrow_G \lambda w_1w_2\dots w_{n-1}w_n \Rightarrow_G e w_1w_2\dots w_{n-1}w_n = w,\end{aligned}$$

odakle zaključujemo $L \subseteq L(G,\lambda)$.

Obratnu inkluziju pokazaćemo indukcijom po dužini izvođenja.

Jedino izvođenje dužine jedan je $\lambda \to e$ i $e \in L$, te u ovom slučaju tvrđenje važi. Pretpostavimo da svaka reč koja se može dobiti iz λ izvođenjem dužine manje ili jednake n pripada jeziku L.

Uzmimo reč $w \in L(G,\lambda)$ koja se može dobiti izvođenjem $\lambda \overset{*}{\Rightarrow}_G w$ dužine $n+1$. Prvi korak ovom izvođenju je $\lambda \to \lambda\sigma$, pa možemo pisati $w = uv$, pri čemu postoje izvođenja $\lambda \overset{*}{\Rightarrow}_G u$ i $\sigma \overset{*}{\Rightarrow}_{G_1} v$ od kojih ni jedno nije duže od n (vidi Zadatak 1.21.). Kako je svako pravilo iz π_1 istovremeno pravilo u π, to na obe reči primenjujemo indukcijsku pretpostavku i dobijamo da $u,v \in L$, odnosno da reč $w = uv \in L$. Ovim je dokaz kompletiran. □

1.3. Hijerarhija Čomskog (Chomsky)

Kalsu svih formalnih gramatika zovemo gramatikama tipa 0. Već smo pomenuli jezike tipa 0, tj. jezike generisane nekom formalnom gramatikom.

Formalnu gramatiku $G = (V, X, \pi)$ nazivamo *kontekstno-zavisnom* gramatikom , ili *gramatikom tipa* 1, ako svako pravilo iz π ima oblik

$$u\alpha v \to upv$$

gde je $\alpha \in V \setminus X$, $p \in V^*$ i $u, v \in (V \setminus X)^*$. Odgovarajuće jezike nazivamo *kontekstno-zavisnim jezicima* ili *jezicima tipa* 1.

Ako je svako pravilo iz π oblika $\alpha \to p$, gde je $\alpha \in V \setminus X$ i $p \in V^*$, tada gramatiku G nazivamo *kontekstno-nezavisnom* gramatikom, *kontekstno-slobodnom* gramatikom ili *gramatikom tipa* 2.

Jezike generisane ovakvim gramatikama nazivamo *kontekstno-nezavisnim jezicima* ili *jezicima tipa* 2.

Osim ovih gramatika, veoma su važne i *regularne gramatike*, koje se ponegde nazivaju i *gramatikama tipa* 3, *desno-linearnim gramatikama* ili *racionalnim gramatikama*. Kod ovih gramatika svako pravilo ima oblik

$$\alpha \to p\beta$$

gde su $\alpha, \beta \in V \setminus X$ i $p \in X^+$, ili $\alpha \to q$, gde je $\alpha \in V \setminus X$ i $q \in X^*$. Jezike generisane ovim gramatikama nazivamo *regularnim jezicima* ili *jezicima tipa* 3.

Ovakvu klasifikaciju gramatika i jezika prvi je napravio američki lingvista Noam Chomsky, i naziva se *hijerarhija Chomsky*.

Ako za $k \in \{0,1,2,3\}$, sa $\mathcal{L}_k$ označimo klasu svih jezika tipa k, i ako sa $\mathcal{L}_2'$ označimo klasu svih jezika iz $\mathcal{L}_2$ koji ne sadrže praznu reč, tada imamo da je

$$\mathcal{L}_3 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_0 \;\text{ i }\; \mathcal{L}_2' \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_0.$$

Postoje primeri koji potvrđuju da su prethodne inkluzije stroge.

Za dve gramatike G_1 i G_2 kažemo da su *ekvivalentne* ako generišu isti jezik, tj. ako je $L(G_1, \sigma_1) = L(G_2, \sigma_2)$.

Zadatak 1.38. *Neka je X dati alfabet. Dokazati da postoji neprebrojivo mnogo jezika nad X koji nisu generisani gramatikom.*

Rešenje: Skup svih jezika nad alfabetom X je neprebrojiv , tj.

$$|\{L \mid L \subseteq X^*\}| = |\mathcal{P}(X^*)| = 2^{\aleph_0}.$$

Sa druge strane svaka gramatika se zadaje sa konačno mnogo simbola, što znači da je skup svih gramatika prebrojiv skup. Tako ostaje neprebrojivo jezika koji nisu generisani gramatikama. □

Zadatak 1.39. *Sve klase jezika $\mathcal{L}_k$, $k \in \{0,1,2,3\}$ su zatvorene za uniju, proizvod i Kleene-jevu zvezda operaciju. Dokazati.*

Rešenje: Pogledati dokaz Zadatka 1.37. □

Zadatak 1.40. *Neka su X^* i Y^* monoidi reči nad alfabetima X i Y tim redom i neka je $\phi : X^* \to Y^*$ homomorfizam. Ako je jezik L kontekstno-nezavisan u X^*, onda je $\phi(L)$ konteksno-nezavisavisan jezik u Y^*. Dokazati.*

Rešenje: Neka je $G = (V, X, \pi)$ kontekstno-nezavisna gramatika koja generiše jezik L, tj. $L = L(G, \sigma)$. Definišimo preslikavanje $\widehat{\phi} : V \to (V \setminus X) \cup Y$ sa

$$\widehat{\phi}(v) = \begin{cases} \phi(v) & , \ v \in X \\ v & , \ v \in V \setminus X \,. \end{cases}$$

Neka je $\widehat{\pi}$ skup izvođenja takvih da, svakom izvođenju $\alpha \to w$ iz π, odgovara izvođenje u $\widehat{\pi}$ oblika $\widehat{\phi}(\alpha) = \alpha \to \widehat{\phi}(w)$. Pokazaćemo da gramatika $\widehat{G} = (\widehat{\phi}(V), B, \widehat{\pi})$ generiše jezik $\hat{\phi}(L)$, odnosno da je $\widehat{\phi}(L) = L(\widehat{G}, \sigma)$ (bez umanjenja opštosti prirodno proširujemo homomorfizam $\widehat{\phi} : V^* \to ((V \setminus X) \cup Y)^*$).

Za proizvoljnu reč $w \in \phi(L)$ postoji reč $u \in L$ za koju postoji niz izvođenja

$$\sigma \Rightarrow_G u_1 \Rightarrow_G \cdots \Rightarrow_G u_n = u$$

i $w = \phi(u) = \widehat{\phi}(u)$. Tada imamo

$$\widehat{\phi}(\sigma) = \sigma \Rightarrow_{\widehat{G}} \widehat{\phi}(u_1) \Rightarrow_{\widehat{G}} \cdots \Rightarrow_{\widehat{G}} \widehat{\phi}(u_n) = \widehat{\phi}(u) = \phi(u) = w,$$

pa $w \in L(\widehat{G}, \sigma)$.

Prepostavimo, sada, da $w \in L(\widehat{G}, \sigma)$ i da se dobija izvođenjem

$$\sigma \Rightarrow_{\widehat{G}} w_1 \Rightarrow_{\widehat{G}} \cdots \Rightarrow_{\widehat{G}} w_n = w.$$

To znači da u π postoji izvođenje

$$\sigma \Rightarrow_G v_1 \Rightarrow_G \cdots \Rightarrow_G v_n = v,$$

pri čemu je $\widehat{\phi}(v_i) = w_i$ za $i \in \{1, 2, \ldots n\}$, pa $v \in L(G, \sigma)$. Jasno da važi

$$w = \widehat{\phi}(v) = \phi(v) \in \phi(L),$$

tj. $\phi(L) = L(\widehat{G}, \sigma)$. Ovim smo kompletirali dokaz. □

Zadatak 1.41. *Neka je gramatika $G = (V, X, \pi)$ kontekstno-nezavisna gramatika i neka je $\mu_1\mu_2 \overset{*}{\Rightarrow} \eta$ izvođenje u G, gde su $\mu_1, \mu_2, \eta \in V^*$. Tada je η oblika $\eta = \eta_1\eta_2$, postoje izvođenja*

$$\mu_1 \overset{*}{\Rightarrow} \eta_1 \quad i \quad \mu_2 \overset{*}{\Rightarrow} \eta_2$$

i ni jedno od tih izvođenja nije duže od niza izvođenja $\mu_1\mu_2 \overset{*}{\Rightarrow} \eta$. Dokazati.

Rešenje: Tvrđenje zadatka dokazaćemo indukcijom po dužini izvođenja

$$\mu_1\mu_2 \overset{*}{\Rightarrow} \eta.$$

Ako je niz dužine jedan, upotrebljeno je pravilo $\alpha \to z$, za neki $\alpha \in V \setminus X$ i $z \in V^*$, što znači da je α sadržano ili u μ_1 ili u μ_2. Bez umanjenja opštosti pretpostavimo da je $\mu_1 = u\alpha v$, $u, v \in V^*$. Tada je $\mu_1\mu_2 = u\alpha v\mu_2 \Rightarrow uzv\mu_2 = \eta$, pa je $\eta = \eta_1\eta_2$, $\eta_1 = uzv$, $\eta_2 = \mu_2$ i postoje izvođenja $\mu_1 \Rightarrow uzv$ i $\mu_2 \overset{*}{\Rightarrow} \eta_2$, dužine 1 i 0, tim redom.

Pretpostavimo da tvrđenje važi za svako izvođenje dužine manje od n i dokažimo da važi za izvođenje $\mu_1\mu_2 \overset{*}{\Rightarrow} \eta$ dužine n. Prvi korak u ovom izvođenju koji ima oblik $\mu_1\mu_2 \Rightarrow \mu_1'\mu_2$ koristi pravilo $\alpha \to z$, za neki $\alpha \in V \setminus X$ i $z \in V^*$, pa je $\mu_1 = u\alpha v$ i $\mu_1' = uzv$. Izvođenje $\mu_1'\mu_2 \overset{*}{\Rightarrow} \eta$ je dužine $n-1$ pa, primenom indukcijske pretpostavke, dobijamo da je $\eta = \eta_1\eta_2$ i ni jedno od izvođenja

$$\mu_1' \overset{*}{\Rightarrow} \eta_1 \quad \text{i} \quad \mu_2 \overset{*}{\Rightarrow} \eta_2$$

nije dužine veće od $n-1$. Dakle, izvođenje $\mu \Rightarrow \mu_1' \overset{*}{\Rightarrow} \eta_1$ je dužine ne veće od n, čime je tvrđenje dokazano. □

Zadatak 1.42. *Dokazati da za svaku kontekstno-nezavisnu gramatiku postoji ekvivalentna kontekstno-nezavisna gramatika koja zadovoljava sve uslove za konteksno-zavisne gramatike.*

Rešenje: Neka je $G = (V, X, \pi)$ data kontekstno-nezavisna gramatika i pretpostavimo da $e \notin L(G, \sigma)$. Definišimo skup $U \subseteq V$ na sledeći način:

$$U = \{\mu \in V \mid \mu \overset{*}{\Rightarrow}_G e \in \pi\}.$$

Nije teško videti da se skup U može konstruisati u konačno mnogo koraka. Naime, definišimo niz skupova $U_1, U_2, \ldots, U_n, \ldots$ na sledeći način:

$$\begin{aligned} U_1 &= \{\mu \in V \mid \mu \to e \in \pi\} \\ U_{n+1} &= U_n \cup \{\mu \in V \mid (\exists \alpha \in {U_n}^*)\ \mu \to \alpha \in \pi\} \\ U &= \bigcup_{n \geqslant 1} U_n. \end{aligned}$$

Tada je $U_1 \subseteq U_2 \subseteq \cdots \subseteq U_n \subseteq \cdots \subseteq V \setminus X$. Pošto je $V \setminus X$ konačan skup, sledi da postoji indeks k, tako da se svi skupovi U_m za $m \geqslant k$ poklapaju, pa je $U = U_k$. Tražena gramatika $G_1 = (V, X, \pi_1)$ ima izvođenja oblika $\mu \to \alpha_1$, gde je $\alpha_1 \neq e$ i postoji reč $\alpha \in X^*$ takva da je $\mu \to \alpha$ izvođenje gramatike G, a α_1 se dobija iz α izostavljanjem praznih reči ili više pojavljivanja jednog ili više simbola iz U. Dokazaćemo da je $L(G, \sigma) \setminus \{e\}$ jezik generisan gramatikom G_1.

Zaista, ako se reč w može izvesti u gramatici G_1, onda se odgovarajuća izvođenja mogu konstruisati i u gramatici G. Obratno, ako se neka reč $w \neq e$

može izvesti u gramatici G, onda se ona može izvesti i u gramatici G_1, tako što se izvođenja oblika $\mu \to e$ koja učestvuju u datom izvođenju "spoje" sa izvođenjima oblika $\mu \to \alpha$, $\alpha \neq e$, tako da na kraju dobijemo izvođenja iz π_1. Ovim smo pokazali da je jezik generisan gramatikom G_1, tačno $L(G,\sigma) \setminus \{e\}$.

Ako $e \in L(G,\sigma)$, onda je dovoljno u gramatici G_1, skupu $V \setminus X$ dodati novi simbol σ_1 kao novi početni simbol . Ako skupu izvođenja π_1 dodamo dva nova izvođenja $\sigma_1 \to \sigma$ i $\sigma_1 \to e$, dobijamo gramatiku koja generiše isti jezik kao G i ima traženi oblik. □

Zadatak 1.43. *Za kontekstno-nezavisnu gramatiku postoji algoritam za utvrđivanje da li jezik generisan datom kontekstno-nezavisnom gramatikom sadrži praznu reč ili ne. Dokazati.*

Rešenje: Direktna posledica dokaza tvrđenja prethodnog zadatka. □

Zadatak 1.44. *Dokazati da je svaki kontekstno-nezavisan jezik ujedno i kontekstno-zavisan, tj.* $\mathcal{L}_2 \subseteq \mathcal{L}_1$.

Rešenje: Tvrđenje je direktna posledica Zadatka 1.42. i definicija gramatika tipa 1 i 2. □

Zadatak 1.45. *Konstruisati konteksno-nezavisnu gramatiku koja generiše jezik*

$$L = \{0^n 1^{2n} \mid n \geqslant 0\}.$$

Rešenje: Pokazaćemo dva načina za konstrukciju tražene gramatike:

Prvi način: Primetimo da pravilo $\sigma \to 0\sigma 11$ uvek generiše odgovarajući par 0 i 11. Primenom ovog pravila n puta, dobijamo izraz oblika $0^n\sigma(11)^n$. Ovako jednostavna pravila su veoma korisna u konstrukciji kontekstno-nezavisnih gramatika.

Drugi način: Drugi način da se nađe gramatika je dobijanje rekurzivnog izraza od dužih reči (izraza) u L izraženih preko kraćih reči jezika L. Naime, duža reč $0^{n+1}1^{2(n+1)}$ u L može da se napiše kao $0(0^n1^{2n})11$. Zaključujemo da nam je potrebno pravilo $\sigma \to 0\sigma 11$ da bi iz izvođenja $\sigma \stackrel{*}{\Rightarrow} 0^n1^{2n}$ dobili izvođenje $\sigma \stackrel{*}{\Rightarrow} 0^{n+1}1^{2(n+1)}$. □

Zadatak 1.46. *Konstruisati kontekstno-nezavisnu gramatiku koja generiše jezik*

$$L = \{u \in \{a,b\}^* \mid \textit{svaki prefiks od } u \textit{ ima bar onoliko } a\textit{-ova, koliko i } b\textit{-ova}\}.$$

Rešenje: Za konstrukciju tražene gramatike koristićemo metod rekurzije.

Jesno je da prezna reč $e \in L$. Primetimo, da u proizvoljnoj reči $u \in L$, prvo slovo mora biti a, jer je prvo slovo i samo prefiks od u. Dakle, imamo $u = av$. Ako $v \in L$, onda je $u = av$ traženi rekurzivni izraz.

Posmatrajmo slučaj kada $v \notin L$. U ovom slučaju, v mora da sadrži prefiks koji ima jedno slovo b više nego a. Neka je p najkraći ovakav prefiks od v. Tada se p završava slovom b i pišemo $p = wb$ i $u = awbz$. Prema pretpostavci za p, svaki prefiks od w ima bar onoliko slova a koliko i b. To znači da $w \in L$.

Dalje, $p = wb$ ima tačno jedno b više nego a, pa awb ima isti broj slova a i b. Pretpostavimo, sada da je q prefiks od z. Tada je $awbq$ prefiks od u, pa nema više b-ova nego a-ova, što znači da isto važi i za q. Ovim smo pokazali da $q \in L$.

Prema prethodnoj analizi vidimo da se neprazna reč $u \in L$ može izraziti ili kao $u = av$, za neku reč $v \in L$ ili je $u = awbz$, za neki $w, z \in L$. Koristeći σ kao polazni simbol koji generiše sve reči iz L, možemo generisati gramatiku $G = (\{a, b, \sigma\}, \{a, b\}, \pi)$, gde je $\pi = \{\sigma \to e, \sigma \to a\sigma, \sigma \to a\sigma b\sigma\}$. □

Zadatak 1.47. *Konstruisati kontekstno-nezavisnu gramatiku koja generiše jezik*

$$L = \{u \in \{a, b\}^* \mid 2|u|_a = |u|_b\}.$$

Rešenje: Neka je $f(u) = 2|u|_a - |u|_b$. Tada je $L = \{u \in \{a, b\}^* \mid f(u) = 0\}$. Vrednosti funkcije f za prefiks-podreči proizvoljne reči menjaju se na sledeći način:

(a) rastuće su za vrednosti 2;
(b) opadajuće su za vrednosti 1.

Definišimo neterminalne simbole $\sigma, \alpha, \beta, \gamma$ koji predstavljaju četiri jezika nad $\{a, b\}$ na sledeći način:

(i) σ predstavlja jezik $L = \{u \in \{a, b\}^* \mid f(u) = 0\}$;
(ii) $\alpha = \{u \in \{a, b\}^* \mid f(u) = 1\}$;
(iii) $\beta = \{u \in \{a, b\}^* \mid f(u) = -1\}$;
(iv) $\gamma = \{u \in \{a, b\}^* \mid f(u) = 2\}$.

Očigledno je da važi $\sigma = e \cup a\beta' \cup b\alpha$, gde je $\beta' = \{u \in \{a, b\}^* \mid f(u) = -2\}$. Za svaku reč $w \in \beta'$ dužine n i prefiks w_i dužine i takav da je $f(w_i) = -1$, postoji prefiks w_j od w, takav da je $f(w_j) = -1$. Dakle, $\beta' = \beta\beta$, ili ekvivalentno, $\sigma = e \cup a\beta\beta \cup b\alpha$. Primetimo da je $\alpha = b\gamma \cup a\beta$. Iz gornje analize, takođe, sledi $\beta = b\sigma \cup a\beta\beta\beta$.

Na kraju, primetimo da γ može da se izrazi kao $a\sigma \cup b\alpha\gamma \cup b\gamma\alpha$. Da bi ovo pokazali, označimo sa $\gamma' = \{u \mid f(u) = 3\}$. Tada, za svaku reč $w \in \gamma'$ dužine n i prefiks w_i dužine i takav da je $f(w_i) = 1$ ili $f(w_i) = 2$, postoji prefiks w_j od w, takav da je $f(w_j) = 2$ ili $f(w_j) = 1$, redom.

Iz svega prethodnog, dobijamo sledeću konteksno-nezavisnu gramatiku $G = (\{a, b, \sigma, \alpha, \beta, \gamma\}, \{a, b\}, \pi)$, u kojoj su pravila izvođenja

$$\begin{array}{ll} \sigma \to e + a\beta\beta + b\alpha & \qquad \alpha \to b\gamma + a\beta \\ \beta \to b\sigma + a\beta\beta\beta & \qquad \gamma \to a\sigma + b\alpha\gamma + b\gamma\alpha, \end{array}$$

pravila iz π, tako da je dati jezik $L = L(G, \sigma)$. □

Zadatak 1.48. *Konstruisati kontekstno-nezavisnu gramatiku koja generiše jezik*

$$L = \{a^m b^n c^p d^q \mid m + n = p + q\}.$$

Rešenje: Jednakost $m + n = p + q$ pomoćiće nam u konstrukciji pravila izvođenja. Ideja je da slova a i b uparimo sa slovima c i d i to ćemo učiniti rekurzivno.

(1) Simbol σ definiše pravilo $\sigma \to a\sigma d$ koje generiše isto broj a-ova i d-ova. U protivnom, zavisno od toga da li je $m \geqslant q$ dobijamo pravila $\sigma \to \alpha + \beta$.
(2) Simbol α označava slučaj $m \geqslant q$. Tada se generiše isti broj a-ova i c-ova ili prelazimo na pomoćni simbol γ. Tako konstruišemo pravila $\alpha \to a\alpha c + \gamma$.
(3) Simbol β koristimo za slučaj $m \leqslant q$. Pritom se generiše isti broj a-ova i c-ova ili se pomeramo na pomoćni simbol γ. Tako konstruišemo pravila $\beta \to b\beta d + \gamma$.
(4) Na kraju, pomoćnim simbolom γ sparujemo simbole b i c koristeći pravila $\gamma \to b\gamma c + e$.

Tako dobijamo skup pravila izvođenja

$$\sigma \to a\sigma d + \alpha + \beta \qquad \alpha \to a\alpha c + \gamma$$
$$\beta \to b\beta d + \gamma \qquad \gamma \to b\gamma c + e,$$

u gramatici koja generiše dati jezik. □

Zadatak 1.49. *Konstruisati konteksno-nezavisnu gramatiku koja generiše jezik*

$$L = \{a^m b^n \mid 3m \leqslant 5n \leqslant 4m\}.$$

Rešenje: Najpre posmatramo jednostavniji jezik

$$L_0 = \{a^m b^n \mid 3m \leqslant 5n \leqslant 4m,\, m \equiv 0\,(\mathrm{mod}\,5)\}.$$

Za reč $a^{5p} b^n \in L_0$, mora da važi $15p \leqslant 5n \leqslant 20p$ ili $3p \leqslant n \leqslant 4p$. To znači da je

$$L_0 = \{a^{5p} b^{3p+k} \mid 0 \leqslant k \leqslant p\}.$$

Lako se pokazuje da je jezik L_0 generisan gramatikom $G_0 = (\{a,b,\alpha\},\{a,b\},\pi)$ sa sledećim pravilima:

$$\alpha \to a^5 \alpha b^3 + a^5 \alpha b^4 + e,$$

tj. da je $L_0 = L(G_0,\alpha)$. Gornja ideja sparivanja pet simbola a sa tri ili četiri simbola b je najbitnija za ovaj problem. Primenićemo ovu ideju i na ostale stringove u L. Definišimo, za $i \in \{1,2,3,4\}$, jezike

$$L_i = \{a^m b^n \mid 3m \leqslant 5n \leqslant 4m,\, m \equiv i\,(mod\,5)\}.$$

Za svaki L_i, $i \in \{1,2,3,4\}$ ideju uparivanja slova možemo iskoristiti za konstrukciju gramatika koje ih generišu.

Proizvoljna reč jezika L_4 ima oblik $a^{5p+4} b^n$, gde je $3(5p+4) \leqslant 5n \leqslant 4(5p+4)$, ili $3p + \frac{12}{5} \leqslant n \leqslant 4p + \frac{16}{5}$. Obzirom da n mora biti ceo broj važi relacija

$$3p + 3 \leqslant n \leqslant 4p + 3.$$

Ova relacija nam govori da možemo upariti svakih pet a-ova sa tri ili četiri b-ova. Tako je jezik L_4 generisan gramatikom $G_4 = (\{a,b,\sigma_4,\alpha\},\{a,b\},\pi)$ sa pravilima

$$\sigma_4 \to a^4\alpha b^3 \qquad \alpha \to a^5\alpha b^3 + a^5\alpha b^4 + e,$$

odnosno $L_4 = L(G_4, \sigma_4)$.

Gramatika G_3 koja ga generiše jezik $L_3 = L(G_3, \sigma_3)$, pored pravila gramatike G_0 sadrži i pravilo oblika $\sigma_3 \to a^3\alpha b^2$.

Za jezik L_2 nejdnakost $3(5p+2) \leqslant 5n \leqslant 4(5p+2)$ se može pojednostaviti do $3p+2 \leqslant n \leqslant 4p+1$. Ako uparujemo $5p$ kopija a-ova sa j kopija b-ova, uz uslov $3p \leqslant j \leqslant 4p$ ne možemo spojiti još dva a ni sa jednim brojem b-ova. Ako uparimo dva a sa dva b možemo dobiti $4p+2$ kopije b-ova, što ne zadovoljava jednakost, a ako uparimo ta dva a sa samo jednim b možemo dobiti $3p+1$ kopija simbola b. Da bi rešili ovaj problem, menjamo gornju relaciju sa

$$3(p-1)+5 \leqslant n \leqslant 4(p-1)+5.$$

Ovo znači da su reči u L_2 oblika $a^{5(p-1)+7}b^{q+5}$, pri čemu je $3(p-1) \leqslant q \leqslant 4(p-1)$. (Za p=0 ne postoji n za koje je nejednakost zadovoljena). Prema tome, treba da uparimo sedam simbola a sa pet b, tj. treba nam pravilo $\sigma_2 = a^7\alpha b^5$.

Jezik L_1 sličan je jeziku L_2. Osnovna nejednakost $3(5p+1) \leqslant 5n \leqslant 4(5p+1)$ ekvivalentna je sa $3p+1 \leqslant n \leqslant 4p$. Odavde dobijamo

$$3(p-1)+4 \leqslant q \leqslant 4(p-1)+4.$$

Pored pravila gramatike G_0 treba nam pravilo $\sigma_1 \to a^6\alpha b^4$.

Kako je L unija jezika L_i, za $0 \leqslant i \leqslant 4$, kombinovanjem ovih pravila dobijamo skup pravila

$$\begin{aligned}
&\sigma \to a^4\alpha b^3 + a^3\alpha b^2 + a^7\alpha b^5 + a^6\alpha b^4 + \alpha \\
&\alpha \to a^5\alpha b^3 + a^5\alpha b^4 + e,
\end{aligned}$$

u gramatici koja generiše jezik L. □

1.4. Regularni jezici i regularni izrazi

Regularni jezici (ili *regularni skupovi*) nad alfabetom X definišu se rekurzivno na sledeći način:

(1) Prazan skup $\emptyset$ je regularan jezik.
(2) Za svako slovo $x \in X$, $\{x\}$ je regularan jezik.
(3) Ako su L_1 i L_2 regularni jezici, onda su $L_1 \cup L_2$, L_1L_2 i L_1^* regularni jezici

Da bi pojednostavili predstavljanje regularnih jezika koristimo *regularne izraze* nad alfabetom X na sledeći način:

(1) $\emptyset$ je regularan izraz koji predstavlja prazan skup.
(2) e je regularan izraz koji predstavlja jezik $\{e\}$.

(3) Za $x \in X$, x jeste regularni izraz koji predstavlja jezik $\{x\}$.
(4) Ako su r_{L_1} i r_{L_2} regularni izrazi koji predstavljaju jezike L_1 i L_2, redom, onda su $(r_{L_1}) \cup (r_{L_2})$, $(r_{L_1})(r_{L_2})$ i $(r_{L_1})^*$ regularni izrazi koji predstavljaju $L_1 \cup L_2$, L_1L_2 i L_1^*, redom.

Da bi smanjili broj zagrada u regularnim izrazima dajemo prioritet nekim operacijama:
(1) Zvezda operacija ima prioritet nad unijom i konkatenacijom.
(2) Konkatenacija ima prioritet nad unijom.
Jednostavno se pokazuje da operacije $+$ i $\cdot$ u regularnim izrazima zadovoljavaju distributivni zakon: Za proizvoljne regularne izraze r_1, r_2 i s važi

$$\begin{aligned} s(r_1 + r_2) &= sr_1 + sr_2, \\ (r_1 + r_2)s &= r_1 s + r_2 s. \end{aligned}$$

Zadatak 1.50. *Naći regularni izraz koji predstavlja binarne zapise celih brojeva koji su stepeni broja* 4.

Rešenje: Binarno predstavljanje broja $4^n = 2^{2n}$ je

$$1\underbrace{00\ldots0}_{2n}.$$

Prema tome, možemo ga predstaviti izrazom $1(00)^*$. □

Zadatak 1.51. *Naći regularni izraz koji predstavlja binarne reči u kojima se podreč* 001 *javlja najmanje jedanput.*

Rešenje: Tražena reč je oblika $u001v$, gde su u i v proizvoljni binarni stringovi. Tako, regularni izraz koji predstavlja dati skup reči, možemo predstaviti u obliku: $(0+1)^*001(0+1)^*$. □

Zadatak 1.52. *Naći regularni izraz koji predstavlja skup B binarnih reči sa najviše jednim parom uzastopnih nula i najviše jednim parom uzastopnih jedinica.*

Rešenje: Reč $u \in B$ može da ima jedan od sledećih oblika:

$$e,\ u_1 0,\ u_0 1,\ u_1 00 v_1,\ u_0 11 v_0,\ u_1 00 w_1 11 v_0,\ u_0 11 w_0 00 v_1,$$

gde su $u_0, u_1, v_0, v_1, w_0, w_1$ reči u kojima se 00 i 11 ne javljaju kao podreči, i u_0 se završava sa 0, u_1 se završava jedinicom, v_0 počinje nulom, v_1 počinje jedinicom, w_0 počinje nulom i završava jedinicom i w_1 počinje jedinicom i završava nulom. Ove rači mogu biti predstavljene regularnim izrazima:

$$\begin{array}{lll} u_1 0 : (e+0)(10)^* & 0v_1 : (01)^*(e+0) & 0w_1 1 : (01)^* \\ u_0 1 : (e+1)(01)^* & 1v_0 : (10)^*(e+1) & 1w_0 0 : (10)^*. \end{array}$$

Primenom ovih pravila na reč u dobijamo regularni izraz koji predstavlja traženi skup B. □

Zadatak 1.53. *Svaki regularan jezik ima regularan izraz u disjunktivnoj normalnoj formi* $r_1 + r_2 + \cdots + r_n$, *u kome ni jedan* r_i, *za* $i = 1,2,\ldots n$ *ne sadrži operaciju* $''+''$. *Dokazati.*

Rešenje: Dokaz izvodimo indukcijom po složenosti regularnog jezika.

Jezik $L = \emptyset$ sadrži regularan izraz $\emptyset$ u disjunktivnoj normalnoj formi. Za proizvoljno slovo x, jezik $L = \{x\}$ ima regularan izraz x u disjunktivnoj normalnoj formi.

Pretpostavimo da jezik L_1 sadrži regularan izraz $r_1 + r_2 + \cdots + r_m$ u disjunktivnoj normalnoj formi i jezik L_2 sadrži regularan izraz $s_1 + s_2 + \cdots + s_n$ u disjunktivnoj normalnoj formi. Tada

(i) jezik $L_1 \cup L_2$ sadrži regularan izraz $r_1 + r_2 + \cdots + r_m + s_1 + s_2 + \cdots + s_n$.

(ii) $L_1 L_2$ ima regularan izraz

$$\Big(\sum_{i=1}^{m} r_i\Big)\Big(\sum_{j=1}^{n} s_j\Big) = \sum_{i=1}^{m}\sum_{j=1}^{n} r_i s_j.$$

(iii) L_1^* ima regularan izraz $(\sum_{i=1}^{m} r_i)^* = (r_1^* r_2^* \cdots r_m^*)^*$ (jednostavno se pokazuje).

Odavde je jasno da svaki regularni jezik sadrži regularan izraz u disjunktivnoj normalnoj formi. □

Zadatak 1.54. *Dokazati da nad svakom azbukom* X *postoji neprebrojivo mnogo neregularnih jezika.*

Rešenje: Kako je $X \neq \emptyset$, skup X^* je beskonačan, pa je skup svih jezika nad monoidom X^*(skup svih podskupova od X^*, tj. $\mathcal{P}(X^*)$) neprebrojiv. Sa druge strane, regularnih izraza nad konačnim skupom X ima samo prebrojivo mnogo. Tako postoji neprebrojivo mnogo jezika nad X koji se ne mogu predstaviti regularnim izrazom. □

1.5. Reprezentacija regularnih izraza pomoću grafova

Usmereni graf je uređeni par $G = (V,E)$, gde je V neprazan skup, čije elemente nazivamo *čvorovima* grafa, dok je $E \subseteq V \times V$ i njegove elemente nazivamo *granama* grafa. Za granu $(a,b) \in E$ kažemo da polazi iz čvora a i završava se u b i kažemo da je usmerena grana grafa G. Prema tome, grana (a,b) je izlazna grana za čvor a i ulazna grana za čvor b.

Petlja je grana koja i počinje izavršava se u istom čvoru, tj. koja je istovremeno i ulazna i izlazna grana za isti čvor.

Put u grafu predstavlja konačan niz grana $e_1, e_2, \ldots, e_n$, takav da za svaki $i \in \{1,2,\ldots,n-1\}$, početni čvor grane e_{i+1} jeste završni čvor grane e_i. *Ciklus* je put koji počinje i završava se u istom čvoru.

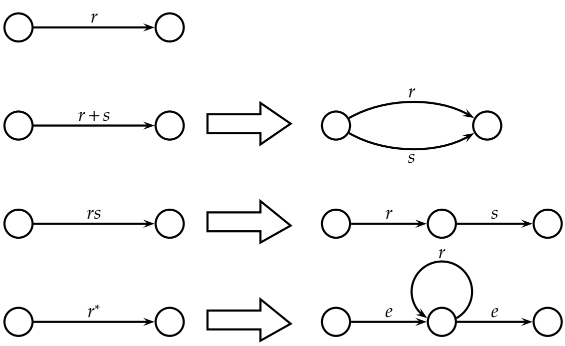
U dijagramu grafa $G = (V,E)$ čvor se obično predstavlja krugom unutar koga se piše oznaka čvora, dok strelica koja počinje u čvoru a i završava se u čvoru b predstavlja granu (a,b). Kako ponekad postoji više takvih grana, svakoj grani pridružićemo *oznaku*, tj. posmatraćemo, tzv. *označene grafove*. Formalno, grana u označenom grafu je trojka (a,x,b), gde su a i b čvorovi, a x je oznaka grane.

Postoji interesantan način za predstavljanje regularnih izraza označenim grafovima pri čemu su oznake grana iz skupa $X \cup \{e\}$. Označeni graf kojim se može predstaviti dati regularan izraz r dobija se na sledeći način:

1) Počinjemo sa dva čvora-*inicijalnim* i *završnim* i crtamo granu, od inicijalnog ka završnom čvoru, koju označavamo sa r.
2) Naredni postupak ponavljamo sve dok oznaka svake grane ne bude slovo iz $X \cup \{e\}$:

   Svaku granu koja ima oznaku $r+s$ menjamo sa dve grane sa oznakama r i s.
   Svaku granu sa oznakom rs menjamo dodavanjem novog čvora i dve grane označene sa r i s.
   Svaku granu čija je oznaka r^* zamenjujemo novim čvorom i dodajemo tri nove grane označene sa e,r,e kao na datoj slici.

3) Brišemo sve grane sa oznakom $\emptyset$.



(R1) Graf reprezentacija $G(r)$ regularnog izraza r.

Zadatak 1.55. *Neka je r regularan izraz. Dokazati da reč u pripada jeziku $L(r)$ ako i samo ako postoji put u grafu $G(r)$ koji je označen sa u.*

Rešenje: Primetimo da važi sledeće:

Reč $u \in L(r)$ ako i samo ako postoji inicijalni čvor a_1 i završni a_k i put među čvorovima $a_1, a_2, \dots a_k$ grafa G takav da $u \in L(r_1)L(r_2)\dots L(r_{k-1})$, pri čemu je r_i oznaka grane (a_i, a_{i+1}) za $i = 1, \dots, k-1$.

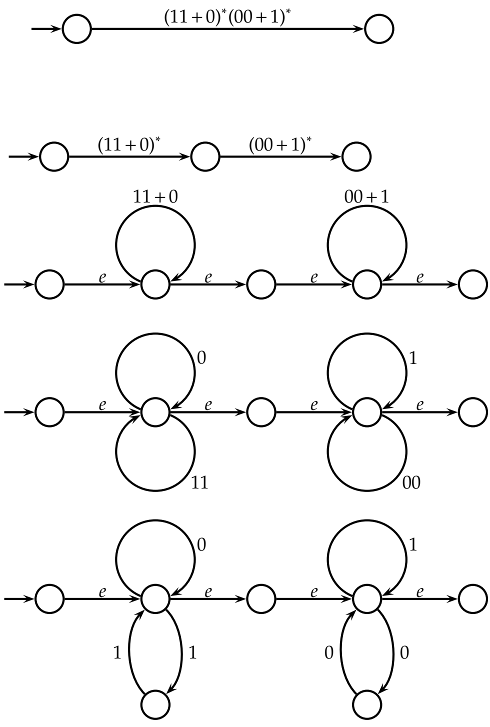
Pomoćno tvrđenje **S:** Prethodno tvrđenje važi za izraze oblika $x+y$, xy, x^*, gde su x,y slova ulaznog alfabeta (videti graf reprezentaciju (R1)).

Dalje, zbog načina zamene broja i oznaka grana u grafu, koji predstavlja dati izraz, tvrđenje će važiti i za izraze koji se dobijaju kombinacijom i većim brojem ponavljanja izraza ovog oblika. To znači da tvrđenje važi, pri formiranju grafa G, na kraju drugog koraka.

U trećem koraku brišemo grane sa oznakom $\emptyset$, jer je $L(\emptyset) = \emptyset$. Dakle, na kraju trećeg koraka svaka grana je označena tačno jednim simbolom iz $X \cup \{e\}$, te pomoćno tvrđenje **S** povlači da $u \in L(r)$ ako i samo ako postoji put u $G(r)$ od a_1 do a_k čija je oznaka upravo u.

Zadatak 1.56. *Konstruisati graf $G(r)$ regularnog izraza $r = (11+0)^*(00+1)^*$.*

Rešenje: Postupak kojim konstruišemo traženi graf predstavićemo grafički na sledeći način:



Dobili smo označeni graf $G(r)$ izraza $r = (11+0)^*(00+1)^*$. $\square$

Zadatak 1.57. *Neka je r regularan izraz. Tada svaka e-grana (u,v) u $G(r)$ koja je jedinstvena izlazna grana nefinalnog čvora a ili jedinstvena ulazna grana neinicijalnog čvora b može da se skupi u jedan čvor, pri čemu se zadržavaju sve osobine grafa. Dokazati.*

Rešenje: Da bi uočili zašto je moguće obrisati e-grane, pretpostavimo da je e-grana (a,b) jedinstvena izlazna grana iz a, uz pretpostavku da a nije finalni čvor. Neka je $G'(r)$ graf dobijen iz $G(r)$ skupljanjem grane (a,b) u jedan čvor

c. Označimo sa a_1 inicijalni, a sa a_k finalni čvor, između kojih je put označen izrazom r. Pokazaćemo da za svaki put π od a_1 do a_k u $G(r)$ postoji put π' iz a_1 u a_k u $G'(r)$ označen na isti način kao π i obratno.

Neka je π put od a_1 do a_k. Ako put ne prolazi kroz čvor a, jasno je da isti put postoji i u $G'(r)$ i da su oznake nepromenjene. Ako π prolazi kroz a delimo ga na dva puta, jedan od a_1 do prvog pojavljivanja čvora a i drugi, od prvog pojavljivanja a do a_k. Obzirom da je (a,b) jedina izlazna grana iz a, drugi deo puta mora da počinje ovom granom sa oznakom e. Skupljanje grane (a,b) u jedan čvor c ne menja prvi deo puta, dok skupljanje početka drugog dela puta u jedan čvor čuva oznake. Tako dobijamo put π' u $G'(r)$ koji ima iste oznake kakve ima i put π u $G(r)$.

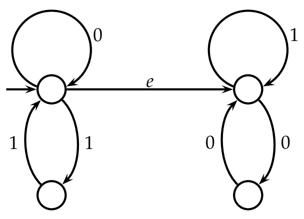
Obratno, neka je π' put od a_1 do a_k u $G'(r)$. Ako put ne prolazi kroz c jasno je da nije došlo do promene puta π iz $G(r)$. Ako put prolazi kroz c, označimo sa π'_1 prvi deo puta od a_1 do c i sa π'_2 deo puta od c do a_k. Kako je (a,b) jedina izlazna e-grana iz a, prva grana (c,x) u π'_2, ako postoji, odgovara grani (b,x) u $G(r)$ (ili, ako je $x=c$, onda (c,c) odgovara (b,b)).

Takođe, ako poslednja grana u π'_1 jeste (y,c), onda, ili postoji grana (y,a) u $G(r)$ ili grana (y,b) u $G(r)$ koja ima istu oznaku kao (y,c). U prvom slučaju menjamo c na putu π'_1 granom (a,b), dok u drugom slučaju čvor c, jednostavno, zamenjujemo čvorom b. Na ovaj način eliminisali smo pojavljivanje čvora c u π' bez promena oznaka.

Nastavljamo postupak sve dok ne eliminišemo sva pojavljivanja čvora c i dobijamo put π u $G(r)$ sa istim oznakama. Na ovaj način dobijamo put π koji počinje u a_1 i završava se u a_k. Ako c nije finalni čvor u $G'(r)$ nismo promenili poslednji čvor na putu π', što znači da se i π i π' završavaju u istom čvoru a_k. Ako je c finalni čvor u $G'(r)$, onda b mora da bude finalni čvor u $G(r)$, obzirom da je, prema pretpostavci, a nefinalni čvor. Ovim smo kompletirali dokaz prvog dela zadatka.

Drugi deo tvrđenja pokazuje se dualno. □

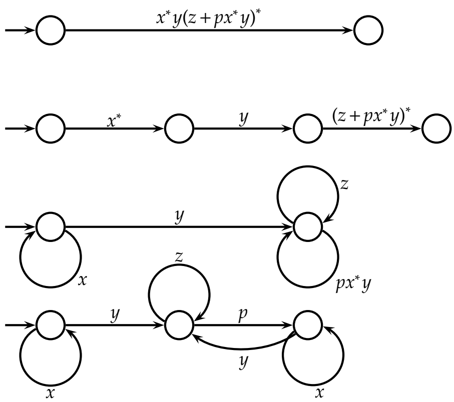
Zadatak 1.58. *Korišćenjem prethodnog zadatka konstruisati graf određen izrazom* $(11+0)^*(00+1)^*$ *iz Zadatka 1.56.*



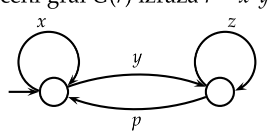
Dobili smo znatno jednostavniji graf $G(r)$. □

Zadatak 1.59. *Konstruisati* $G(r)$ *za* $r = x^*y(z+px^*y)^*$.

Rešenje: Konstrukciju traženog grafa predstavićemo na sledeći način:

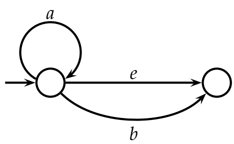


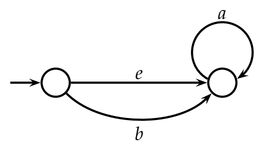
Slika 1.1 Označeni graf $G(r)$ izraza $r = x^* y(z+px^* y)^*$.



Primetimo da smo poslednjim grafom pojednostavili graf koji je dobijen na Slici 1.1. □

Zadatak 1.60. *Odrediti regularne izraze predstavljene sledećim grafovima:*

(a) 

(b) 

Rešenje: (a) $r = a^*(e+b) = a^* + a^*b$; (b) $r = (e+b)a^* = a^* + ba^*$. □

Zadaci za samostalni rad

1. Za jezike L_1, L_2, L_3, L_4 dokazati sledeće identitete:

   (a) $L_1(L_2L_1)^* = (L_1L_2)^*L_1$.
   (b) $(L_1+L_2)^* = (L_1^*L_2^*)^*$.
   (c) $L_1(L_2+L_3) = L_1L_2 + L_1L_3$.
   (d) $L_1^*L_2(L_4L_1^*L_2+L_3)^* = (L_1+L_2L_3^*L_4)^*L_2L_3^*$.

2. Dokazati: $(L_1+L_2+L_3)^* = (L_1^*L_2^*L_3^*)^*$.
3. Dokazati: $((L_1^*+L_3^*+L_1^*L_3^*L_2^*))^* = ((L_1^*L_2^*)^*+L_3^*)^*$.
4. Dokazati:

   (a) $L_1^*(L_2L_1+L_1^*)L_2+(L_2+L_1)^* = (L_1^*L_2^*)^*$.
   (b) $(L_1L_2)^*L_1 = L_1(L_2L_1)^*$.

5. Konstruisati konteksno-nezavisnu gramatiku koja generiše jezik

$$L = \{uz\overline{u} | u \in \{x,y\}^*\}$$

nad alfabetom $X = \{x,y,z\}$.

6. Naći konteksno nezavisnu gramatiku koja generiše jezik

$$L = \{x^n y b^{n+2m} x^m | m,n \geqslant 0\}.$$

7. Pokazati da su sledeći jezici konteksno-nezavisni:

   (a) $L = \{uxvy | u,v \in \{x,y\}^*, |u| = |v|\}$
   (b) $L = \{u_1 z u_2 z \dots z u_k z z u_j^{-1} | k \geqslant 1, u_i \in \{x,y\}^+, i = 1,2,\dots,k, 1 \leqslant j \leqslant k\}$

8. Naći jezik koji je generisan gramatikom $G = (V, X, \pi)$, nad alfabetom $X = \{x,y\}$, pri čemu je $V \setminus X = \{\sigma, \lambda\}$ i skup izvođenja

$$\sigma \to x\sigma, \ \sigma \to x\sigma y\sigma, \ \sigma \to \lambda.$$

9. Naći regularne gramatike koje generišu jezike:

   (a) $L = \{x,y\}^* x^2 \{x,y\}^*$
   (b) $L = \{u \in \{x,y\}^* | |u|_x = 2\}$

10. Da li se jezik $L = \{x^p | p \text{ prost broj}\}$ može generisati regularnom gramatikom?
11. Naći najkraću nepraznu reč za svaki od sledećih jezika:

    (a)$10 + (0+11)0^*1$;
    (b)$(00 + 11 + (01+10)(00+11)^*(01+10))^*$;
    (c)$((00+11)^* + (001+110)^*)^*$.

12. Naći označene grafove koji predstavljaju sledeće regularne izraze:

    (a)$(00+10)(101)^* + 01$;
    (b) $((00+11)^* + (001+110)^*)^*$;
    (c) $(a + bc^*d)^*bc^*$.

13. Naći najjednostavniji graf kojim se može predstaviti e.

Glava 2
Regularni jezici i konačni automati

2.1. Deterministički konačni automati

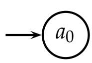
Deterministički konačan automat je uređena petorka $\mathcal{A} = (A, a_0, X, \delta, T)$u kojoj je

A – konačan, neprazan skup stanja;
a_0 – inicijalno stanje;
X – ulazni alfabet;
$\delta : A \times X \to A$ – funkcija prelaza;
$T \subseteq A$ – neprazan skup *završnih* (*finalnih*) stanja.

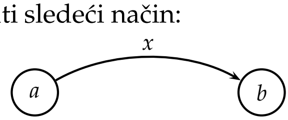
Kako je δ funkcija iz $A \times X$ u A, to postoji tačno jedno stanje $b \in A$ tako da je $b = \delta(a, x)$, odnosno postoji tačno jedano stanje u koje se sa x prelazi iz a.

Princip rada ovako definisanog automata je sledeći:

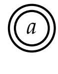
Na početku rada, automat se nalazi u jednom stanju a_0, koje nazivamo *inicijalno stanje*. Inicijalno stanje ćemo grafički označavati ulazećom strelicom na sledeći način:



Prelaz iz jednog stanja u drugo, pod uticajem nekog ulaznog slova, određen je *funkcijom prelaza*. Naime, ako se u izvesnom trenutku automat nalazi u stanju a, pod uticajem ulaznog slova x prelazi u stanje $b = \delta(a, x)$, što ćemo grafički predstavljati sledeći način:



Pored fiksiranja inicijalnog stanja $a_0 \in A$, unapred ćemo fiksirati i skup stanja $T \subseteq A$, koji nazivamo skup *finalnih stanja* (*završnih stanja* ili *terminalnih stanja*). Završna stanja označavaćemo grafički duplim kružićima, na sledeći način:



Automat prelazi iz stanja u stanje čitajući reč ulaznog alfabeta, slovo po slovo. U zavisnosti od toga da li se posle toga našao u stanju koje pripada datom skupu T završnih stanja ili ne, on *prihvata* (*prepoznaje*, *raspoznaje*), odnosno *ne prihvata* (*ne prepoznaje*, *ne raspoznaje*) tu reč.

Dakle, *konačan determinstrički automat* je jednostavan apstraktan matematički model mašine, koji, kako ćemo kasnije pokazati, raspoznaje upravo klasu regularnih jezika. Intuitivno, automat čita ulaznu reč slovo po slovo, po jedno slovo u diskretnoj jedinici vremena, i pošto je ulaz potpuno pročitan odlučuje o tome da li da ga prihvati ili odbije.

Rad automata ne sastoji se samo u jednom prelazu iz stanja u stanje, pod uticajem jednog ulaznog signala, već iz niza uzastopnih prelaza, pod dejstvom niza uzastopnih ulaznih signala.

Na osnovu ovoga uvodimo formalnu, induktivnu definiciju kojom se funkcija prelaza δ sa domena $A \times X$ prirodno proširuje na domen $A \times X^*$, na sledeći način:

(1) $\delta(a,e) = a$;
(2) Za svaku reč $u \in X^*$ i svako slovo $x \in X$, ako je definisano $\delta(a,u)$, važi

$$\delta(a,ux) = \delta(\delta(a,u),x).$$

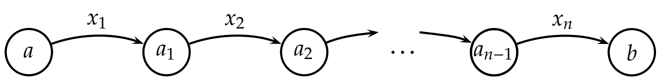
Jednostavno se pokazuje da, za proizvoljne reči $u,v \in X^*$ važi

$$\delta(a,uv) = \delta(\delta(a,u),v).$$

Neka je ulazna reč $u \in X^*$ predstavljena u obliku $u = x_1x_2\cdots x_n$, gde su $x_1,x_2,\ldots,x_n \in X$ slova ulaznog alfabeta, neka su $a,b \in A$, i neka je

$$\delta(a,x_1) = a_1,\ \delta(a_1,x_2) = a_2,\ \ldots,\ \delta(a_{n-1},x_n) = b.$$

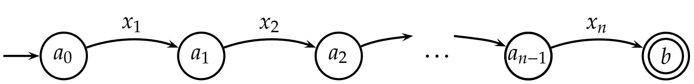
Tada kažemo da automat A pod uticajem ulazne reči u prelazi iz stanja a u stanje b preko niza *međustanja* $a_1,a_2,\ldots,a_{n-1}$. U tom slučaju je $b = \delta(a,u)$ i to se može grafički predstaviti sa



Označimo sa $L(A)$ skup svih reči ulaznog alfabeta koje su prihvaćene konačnim determinističkim automatom $\mathcal{A}$. Skup $L(A)$ naziva se *jezik automata* $\mathcal{A}$. Formalno $L(A)$ definišemo sa

$$L(A) = \{u \in X^* | \delta(a_0,u) \in T\}$$

To ćemo grafički predstaviti sa:



Iz prethodnog se može videti da najprirodniji način za predstavljanje automata jeste njihovo zadavanje pomoću grafova prelaza.

Graf prelaza automata $\mathcal{A} = (A,a_0,X,\delta,T)$ je označen, usmereni graf čiji su čvorovi stanja automata, a oznake grana su slova ulaznog alfabeta. Iz stanja $a \in A$, pod uticajem ulaznog simbola $x \in X$, automat prelazi u stanje $b = \delta(a,x)$, pri čemu graf prelaza ima granu (a,b) koja je označena sa x.

Konačni automati se, takođe, mogu zadavati takozvanim *tablicama prelaza*.

Tablica prelaza automata $A = (A, a_0, X, \delta, T)$ je pravougaona tablica sa vrstama koje odgovaraju svakom ulaznom simbolu i kolonama koje odgovaraju stanjima.

Na mestu u tablici koje odgovara vrsti određenoj ulaznim slovom $x \in X$ i koloni određenoj stanjem $a \in A$ upisuje se stanje $\delta(a, x)$.

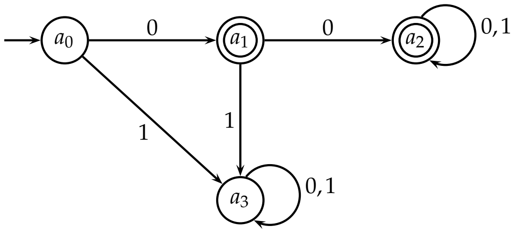
$$\begin{array}{c|ccc} A & \dots & a & \dots \\ \hline \vdots & & \vdots & \\ x & \dots & \delta(a,x) & \dots \\ \vdots & & \vdots & \end{array}$$

Zadatak 2.61. (a) *Konstruisati graf prelaza automata $\mathcal{A} = (A, a_0, X, \delta, T)$, gde je $A = \{a_0, a_1, a_2, a_3\}$, $X = \{0, 1\}$, $T = \{a_1, a_2\}$ i funkcija prelaza δ zadata sa:*

δ	a_0	a_1	a_2	a_3
0	a_1	a_2	a_2	a_3
1	a_3	a_3	a_2	a_3

(b) *Da li automat $\mathcal{A}$ prihvata reči* 000 *i* 010.

Rešenje: (a) Graf prelaza datog automata $\mathcal{A}$ je:



(b) Put koji je označen sa 000 kreće iz inicijalnog stanja a_0 i prolazi kroz stanja $\delta(a_0, 0) = a_1$, $\delta(a_1, 0) = a_2$ i $\delta(a_2, 0) = a_2$. Prema tome, $\delta(a_0, 000) = a_2 \in T$, što znači da $\mathcal{A}$ prihvata reč 000. Slično, $\delta(a_0, 010) = a_3 \notin T$, pa $\mathcal{A}$ ne prihvata reč 010. □

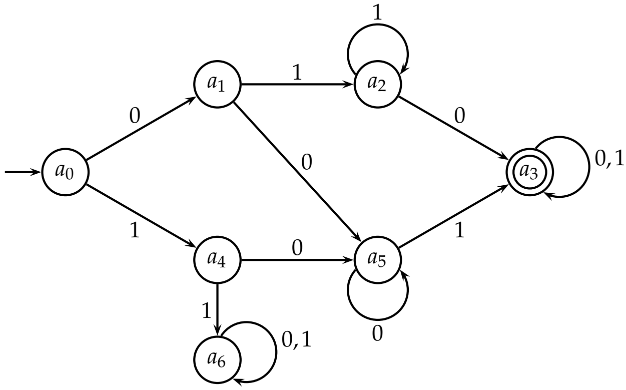
Zadatak 2.62. *Odrediti skup $L(A)$ svih reči koje prihvata automat $\mathcal{A}$ definisan u Zadatku 2.61.*

Rešenje: Primetimo najpre da, ako automat $\mathcal{A}$ jednom dostigne stanje a_3, iz njega ne može izaći, te neće prihvatiti ni jednu reč koja počinje slovom 1 ili sa 01. Sa druge strane, ako $\mathcal{A}$ jednom dostigne stanje a_2 više ga ne napušta, pa $\mathcal{A}$ prihvata sve reči oblika 000^* i 001^*. Takođe, automat $\mathcal{A}$ prihvata i reč 0 koja vodi u završno stanje a_1. Dakle,

$$L(A) = 0 + 000^* + 001^* = 0 + 00(0^* + 1^*)$$

jeste skup svih reči prihvaćenih datim automatom. □

Zadatak 2.63. *Odrediti skup $L(A)$ svih reči koje prihvata automat zadat grafom:*



Rešenje: Primetimo da iz stanja a_6 nema izlaza, pa automat odbacuje reči koje počinju sa 11. Kako je stanje $a_3 \in T$ jedino finalno stanje, treba odrediti sve puteve kojima se iz a_0 može stići u a_3. Dakle, tražene puteve čine grane koje povezuju, redom čvorove (stanja)

$$(a_0, a_1, a_2, \ldots, a_2, a_3),$$
$$(a_0, a_1, a_5, \ldots, a_5, a_3),$$
$$(a_0, a_4, a_5, \ldots, a_5, a_3).$$

Jasno je dati automat prihvata reči koje počinju sa 011^*0, 000^*1 i 100^*1. Dakle, jezik datog automata je

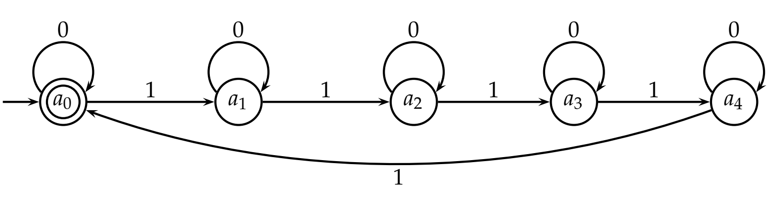
$$L(A) = (011^*0 + 000^*1 + 100^*1)(0+1)^*$$

Primetimo da jezik $L(A)$ možemo naći i rekurzivno. Za stanje $a \in A$, označimo sa L_a jezik koji automat raspoznaje stanjem a kao završnim stanjem. Tada dobijamo sledeću rekurzivnu definiciju jezika:

$$L(A) = L_{a_3} = (L_{a_2}0 + L_{a_5}1)(0+1)^*,$$
$$L_{a_2} = L_{a_1}11^*, \qquad L_{a_5} = (L_{a_4}0 + L_{a_1}0)0^*$$
$$L_{a_4} = L_{a_0}1, \qquad L_{a_1} = L_{a_0}0, \qquad L_{a_0} = \{e\}.$$

Odavde dobijamo jezike $L_{a_1} = 0$, $L_{a_2} = 011^*$, $L_{a_4} = 1$, $L_{a_5} = (10+00)0^*$, pa je traženi jezik $L(A) = L_{a_3} = (011^*0 + 100^*1 + 000^*1)(0+1)^*$. □

Zadatak 2.64. *Odrediti skup $L(A)$ svih reči koje prihvata automat sa slike:*



Rešenje: Na sličan način kao u prethodnom zadatku pokazujemo da je

$$L(A) = 0^*(10^*10^*10^*10^*1)^*$$

skup reči koje raspoznaje automat predstavljen datim grafom prelaza. □

Zadatak 2.65. *Neka su za cele brojeve $n, d \geqslant 1$ definisani konačni deterministički automati $\mathcal{A}_{n,d} = (A, a_0, X, \delta_{n,d}, T)$, gde su $A = \{a_0, a_1, \ldots, a_{n-1}\}$, $X = \{x_0, x_1, \ldots, x_{d-1}\}$, funkcije prelaza su $\delta_{n,d}(a_i, x_k) = a_{(di+k) \bmod n}$ i skup završnih stanja $T = \{a_1\}$.*

(a) *Nacrtati graf prelaza automata $\mathcal{A}_{7,2}$.*
(b) *Neka je $n = 7$, $d = 2$, $x_0 = 0$ i $x_1 = 1$. Naći $\delta_{n,d}(a_3, 0101)$ i $\delta_{n,d}(a_1, 11010)$.*
(c) *Pokazati da je svako stanje $a_j \in A$ dostižno, tj. da postoji reč $u \in X^*$ takva da je $\delta_{n,d}(a_0, u) = a_j$.*

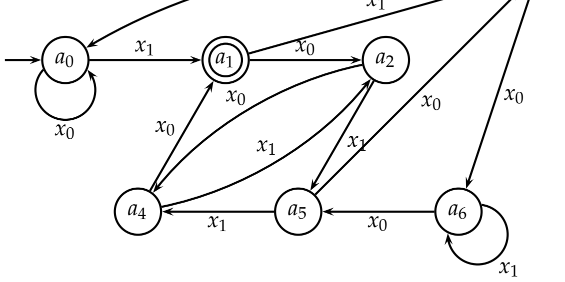
Rešenje: (a) Automat $\mathcal{A}_{7,2} = (A, a_0, X, \delta, T)$ ima skup stanja $A = \{a_0, a_1, \ldots, a_6\}$, ulazni alfabet $X = \{x_0, x_1\}$, dok vrednosti funkcije prelaza $\delta_{7,2}$ računamo na sledeći način:

$$\begin{gathered}
\delta_{7,2}(a_0, x_0) = a_{0\cdot 2+0 \bmod 7} = a_0 \\
\delta_{7,2}(a_0, x_1) = a_{0\cdot 2+1 \bmod 7} = a_1 \\
\delta_{7,2}(a_1, x_0) = a_{1\cdot 2+0 \bmod 7} = a_2 \\
\delta_{7,2}(a_1, x_1) = a_{1\cdot 2+1 \bmod 7} = a_3 \\
\vdots \\
\delta_{7,2}(a_6, x_0) = a_{6\cdot 2+0 \bmod 7} = a_5 \\
\delta_{7,2}(a_6, x_1) = a_{6\cdot 2+1 \bmod 7} = a_6
\end{gathered}$$

Tako dobijamo tablicu prelaza:

$\delta_{7,2}$	a_0	a_1	a_2	a_3	a_4	a_5	a_6
x_0	a_0	a_2	a_4	a_6	a_1	a_3	a_5
x_1	a_1	a_3	a_5	a_0	a_2	a_4	a_6

i graf prelaza automata $\mathcal{A}_{7,2}$:



(b) $$\begin{aligned}\delta_{7,2}(a_3,0101) &= \delta_{7,2}(a_3,x_0x_1x_0x_1) = \delta_{7,2}(\delta(a_3,x_0),x_1x_0x_1)\\ &= \delta_{7,2}(a_6,x_1x_0x_1) = \delta_{7,2}(\delta_{7,2}(a_6,x_1),x_0x_1)\\ &= \delta_{7,2}(a_6,x_0x_1) = \delta_{7,2}(\delta_{7,2}(a_6,x_0),x_1) = \delta_{7,2}(a_5,x_1) = a_4.\end{aligned}$$

Analogno dobijamo da je $\delta(a_1,11010) = \delta(a_1,x_1x_1x_0x_1x_0) = a_2$.

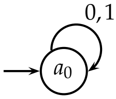
(c) Sa grafa prelaza vidimo da, za svako stanje, postoji put koji od inicijalnog stanja vodi do njega, na primer:

$$\delta(a_0,x_1) = a_1; \quad \delta(a_0,x_1x_0) = a_2; \quad \delta(a_0,x_1x_1) = a_3;$$
$$\delta(a_0,x_1x_0x_0) = a_4; \quad \delta(a_0,x_1x_0x_1) = a_5; \quad \delta(a_0,x_1x_1x_0) = a_6.$$

Jasno je da su sva stanja datog automata dostižna. □

Zadatak 2.66. *Konstruisati automat koji ne prihvata ni jednu reč ulaznog alfabeta* $X = \{0,1\}$.

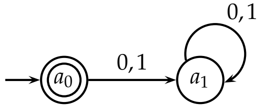
Rešenje: Traženi automat $\mathcal{A} = (A,a_0,X,\delta,T)$ je predstavljen grafom prelaza,



gde je skup stanja $A = \{a_0\}$ jednoelementan, a $T = \emptyset$. □

Zadatak 2.67. *Konstruisati automat koji raspoznaje samo praznu reč nad ulaznim alfabetom* $X = \{0,1\}$.

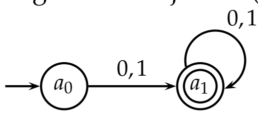
Rešenje: Graf prelaza ovog automata je $\mathcal{A} = (A,a_0,X,\delta,T)$ je



Skup stanja $A = \{a_0,a_1\}$, a skup finalnih stanja $T = \{a_0\}$. □

Zadatak 2.68. *Konstruisati automat koji raspoznaje sve reči nad datim alfabetom* $X = \{0,1\}$ *osim prazne reči, odnosno raspoznaje polugrupu reči* X^+.

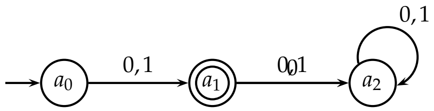
Rešenje: Graf prelaza traženog automata je $\mathcal{A} = (A,a_0,X,\delta,T)$ je



sa skupom stanja $A = \{a_0,a_1\}$ i skupom finalnih stanja $T = \{a_1\}$. □

Zadatak 2.69. *Konstruisati automat koji prihvata tačno reči dužine jedan nad alfabetom* $X = \{0,1\}$, *odnosno samo slova* 0 *i* 1.

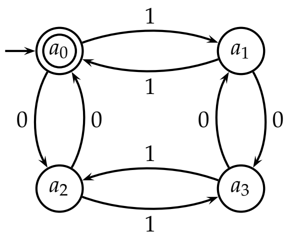
Rešenje: Automat koji prihvata tačno reči dužine jedan je $\mathcal{A} = (A,a_0,X,\delta,T)$



sa skupom stanja $A = \{a_0, a_1, a_2\}$ i skupom finalnih stanja $T = \{a_1\}$. □

Zadatak 2.70. *Konstruisati automat koji raspoznaje tačno one reči binarnog alfabeta koje imaju paran broj nula i paran broj jedinica.*

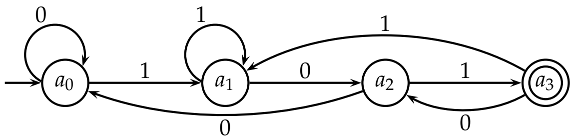
Rešenje: Traženi automat možemo predstaviti grafom prelaza:



Inicijalno stanje je a_0, skup stanja automata $A = \{a_0, a_1, a_2, a_3\}$ i skup završnih stanja $T = \{a_0\}$. □

Zadatak 2.71. *Konstruisati automat koji raspoznaje tačno one binarne reči koje se završavaju sa* 101*.*

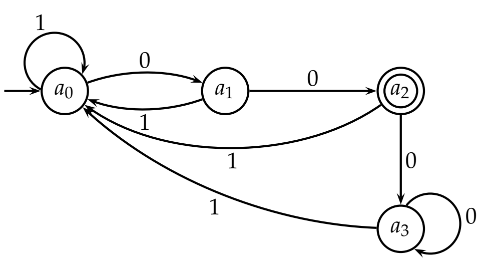
Rešenje: Traženi automat $\mathcal{A}$ možemo predstaviti grafom:



Jednostavno se pokazuje da je, upravo $L(A) = \{u101 \mid u \in \{0,1\}^*\}$ jezik koji se raspoznaje datim automatom. □

Zadatak 2.72. *Konstruisati automat koji prihvata tačno one binarne reči koje se završavaju sa tačno dve uzastopne nule.*

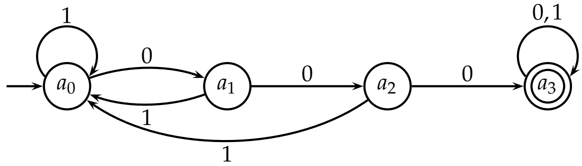
Rešenje: Ovakav automat odbacuje reči koje na kraju imaju više od dve nule, pa dobijamo:



Dakle, automat ima skup stanja $A = \{a_0, a_1, a_2, a_3\}$, inicijalno stanje je a_0 i skup završnih stanja $T = \{a_2\}$. $\square$

Zadatak 2.73. *Konstruisati automat koji prihvata tačno one binarne reči koje za podreč imaju* 000.

Rešenje: Graf prelaza traženog automata je

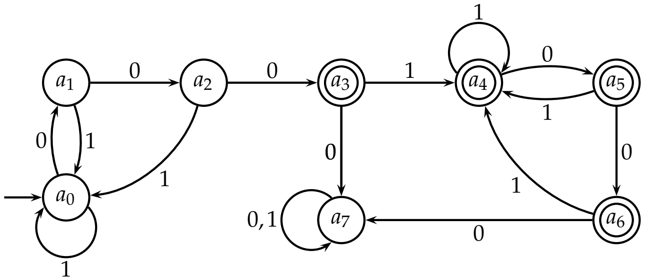


Inicijalno stanje ovog automata je a_0, a skup završnih stanja je $T = \{a_3\}$. $\square$

Zadatak 2.74. *Konstruisati automat koji prihvata tačno one binarne reči koje tačno na jednom mestu imaju tačno tri uzastopne nule.*

Napomena: *Smatramo da se u reči* 0000 *tri uzastopne nule javljaju dva puta.*

Rešenje: Graf prelaza traženog automata je:



Automat ima osam stanja, inicijalno stanje je a_0, a $T = \{a_3, a_4, a_5, a_6\}$ jeste skup završnih stanja. $\square$

Zadatak 2.75. *Konstruisati automat koji prihvata binarne reči koje predstavljaju binarne zapise pozitivnih celih brojeva kongruentnih sa nulom po modulu* 5.

Rešenje: Elementi skupa ostataka po modulu 5 su $\{0,1,2,3,4\}$, pa ćemo pretpostaviti da traženi automat ima pet stanja. Neka je $A = \{a_0, a_1, a_2, a_3, a_4\}$ skup stanja ovog automata. Ako je $x_1x_2\dots x_k \in \{0,1\}^*$ definišimo funkciju prelaza sa

$$\delta(a_0, x_1x_2\dots x_k) = a_i, \ \text{ako je } \ x_1x_2\dots x_k \equiv i\,(mod\ 5).$$

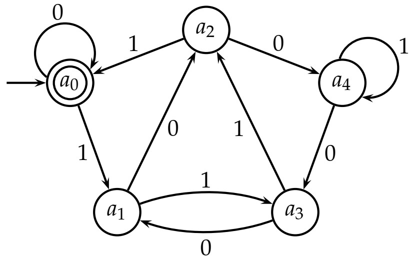
Pri tome, po definiciji, funkcija δ zadovoljava $\delta(\delta(a_0,u),x) = \delta(a_0,ux)$, za proizvoljnu reč $u \in \{0,1\}^*$ i $x \in \{0,1\}$. Da bi odredili vrednosti funkcije prelaza, primetimo da, za $u \equiv i\,(mod\ 5)$ važi:

$$\begin{aligned} ux \equiv j \pmod 5 \ &\Leftrightarrow ux \equiv 2u + x \pmod 5 \\ &\Leftrightarrow \ ux \equiv 2i + x \pmod 5. \end{aligned}$$

Na osnovu ovoga formiramo tablicu prelaza:

δ	a_0	a_1	a_2	a_3	a_4
0	a_0	a_2	a_4	a_1	a_3
1	a_1	a_3	a_0	a_2	a_4

Dakle, graf prelaza traženog automata je:



Kako automat raspoznaje binarne zapise brojeva kongruentnih nuli po modulu 5, inicijalno stanje je istovremeno i završno stanje ovog automata, tj. $T = \{a_0\}$. $\square$

2.2. Minimalni automat jezika

Da bi smo dokazali postojanje automata koji raspoznaje dati jezik L, uvodimo pojam razlomka jezika. U opštem slučaju, automat koji prihvata dati jezik L ne mora biti konačan. Neka je dat jezik $L \subseteq X^*$ i reč $u \in X^*$.

Razlomak jezika L ili *izvod jezika* L u odnosu na reč u, u oznaci $L.u$, je jezik u X^* definisan sa

$$L.u = \{w \in X^* \,|\, uw \in L\}.$$

Jednostavno se pokazuje da, za proizvoljan jezik $L \subseteq X^*$, reči $u, v \in X^*$ i praznu reč $e \in X^*$ važi sledeće:

(i) $(L.u).v = L.uv$;
(ii) $L.e = L$;
(iii) $e \in L.u \;\Leftrightarrow\; u \in L$.

Za jezik $L \subseteq X^*$, označimo sa A_L skup svih razlomaka od L, tj.

$$A_L = \{L.u \,|\, u \in X^*\},$$

i neka je podskup $T_L \subseteq A_L$ definisan sa

$$T_L = \{L.u \,|\, u \in L\}.$$

Na osnovu dela (iii) prethodnog tvrđenja, T_L se može izraziti i sa:

$$T_L = \{H \in A_L \,|\, e \in H\}.$$

Primetimo da, za proizvoljan razlomak $H \in A_L$ i $v \in X^*$ važi $H.v \in A_L$. Naime, ako je $H = L.u$, za neku reč $u \in X^*$, na osnovu tvrđenja (ii) imamo da je

$$H.v = (L.u).v = L.uv \in A_L. \tag{2.1}$$

Odavde se vidi da ima smisla definisati preslikavanje $\delta_L : A_L \times X \to A_L$ sa

$$\delta_L(H,x) = H.x,$$

za $H \in A_L$ i $x \in X$.

Dakle, $\mathcal{A}_L = (A_L, L, X, \delta_L, T_L)$ je automat sa inicijalnim stanjem L i skupom završnih stanja T_L, dok je proširena funkcija prelaza $\delta_L : A_L \times X^* \to A_L$ data sa $\delta_L(H,v) = H.v$ dobro definisana prema (2.1).

Teorema 2.1. *Proizvoljan jezik $L \subseteq X^*$ može se raspoznati automatom*

$$\mathcal{A}_L = (A_L, L, X, \delta_L, T_L)$$

(koji nije obavezno konačan).

Naredna teorema nam daje algoritam za konstrukciju svih razlomaka jezika L, u slučaju kada su alfabet X i skup A_L konačni.

Teorema 2.2. *Neka je $L \subseteq X^*$ proizvoljan jezik. Definišimo induktivno niz $\{A_k\}_{k \in \mathbb{N}^0}$ podskupova od A_L sa:*

$$\begin{aligned} A_0 &= \{L\}, \\ A_{k+1} &= A_k \cup \{H.x \mid H \in A_k,\, x \in X\}, \qquad k \in \mathbb{N}^0. \end{aligned}$$

Tada:

(a)*Niz $\{A_k\}_{k \in \mathbb{N}^0}$ je rastući.*

(b)*Ako postoji $k \in \mathbb{N}^0$ takav da je $A_k = A_{k+1}$, tada je $A_k = A_L$.*

(c)*Ako je A_L konačan skup, tada postoji $k \in \mathbb{N}^0$ takav da je $A_k = A_L$.*

Među svim automatima koji raspoznaju dati jezik, $\mathcal{A}_L$ je automat *najmanje kardinalnosti*, odnosno, automat sa najmanjim brojem stanja, ako se radi o konačnom automatu.

Automat najmanje kardinalnosti koji raspoznaje dati jezik $L \subseteq X^*$ zovemo *minimalni automat* jezika L.

Zadatak 2.76. *Konstruisati minimalni automat koji raspoznaje jezik*

$$L = \{x^m y^n \mid m, n \in \mathbb{N}\},$$

nad alfabetom $X = \{x, y\}$.

Rešenje: Najpre određujemo skup A_1:

$$\begin{aligned} L.x &= \{u \in X^* \mid xu \in L\} = L \cup \{y\}^+, \\ L.y &= \{u \in X^* \mid yu \in L\} = \emptyset, \end{aligned}$$

pa je $A_1 = \{L, L_1, L_2\}$, gde je $L_1 = L.x = L \cup \{y\}^+$ i $L_2 = L.y = \emptyset$.

Dalje, određujemo skup A_2:

$$\begin{aligned}
L_1.x &= L.x^2 = \{u \in X^* \,|\, x^2u \in L\} = L \cup \{y\}^+ = L_1,\\
L_1.y &= L.xy = \{u \in X^* \,|\, xyu \in L\} = \{y\}^*,\\
L_2.x &= \emptyset.x = \emptyset = L_2,\\
L_2.y &= \emptyset.y = \emptyset = L_2,
\end{aligned}$$

pa je $A_2 = \{L, L_1, L_2, L_3\}$, gde je $L_3 = L.xy = \{y\}^*$.

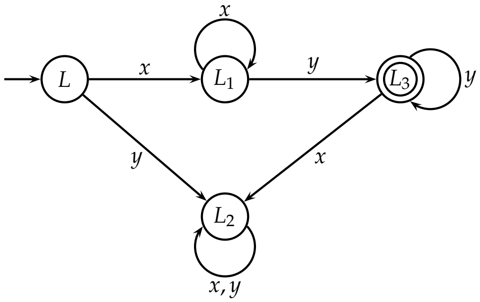
Nastavljajući isti postupak određujemo skup A_3 i dobijamo

$$\begin{aligned}
L_3.x &= L.xyx = \{u \in X^* \,|\, xyxu \in L\} = \emptyset = L_2,\\
L_3.y &= L.xy^2 = \{u \in X^* \,|\, xy^2u \in L\} = \{y\}^* = L_3,
\end{aligned}$$

pa je $A_3 = A_2$. Prema tome, $A_L = A_2 = \{L, L_1, L_2, L_3\}$,

Kako je $L_3 = \{y\}^*$ jedini razlomak iz A_L koji sadrži praznu reč e, to je $T_L = \{L_3\}$.

Dakle, automat $\mathcal{A}_L$ je zadat grafom:



što znači da je minimalni automat $\mathcal{A}_L$ koji raspoznaje dati jezik konačan. □

Zadatak 2.77. *Neka je $X = \{x, y\}$ i $L = \{x^n y^n \,|\, n \in \mathbb{N}\}$. Kontruisati automat $\mathcal{A}_L$ jezika L.*

Rešenje: Primetimo, najpre, da je

$$\begin{aligned}
L.x &= \{u \in X^* \,|\, xu \in L\} = \{x^{n-1}y^n \,|\, n \in \mathbb{N}\},\\
L.y &= \{u \in X^* \,|\, yu \in L\} = \emptyset,
\end{aligned}$$

pa je $A_1 = \{L, L_1, K_1\}$, gde je $L_1 = \{x^{n-1}y^n \,|\, n \in \mathbb{N}\}$ i $K_1 = \emptyset$.

Zatim imamo da je

$$\begin{aligned}
L_1.x &= \{u \in X^* \,|\, xu \in L_1\} = \{x^{n-2}y^n \,|\, n \geqslant 2\},\\
L_1.y &= \{u \in X^* \,|\, yu \in L_1\} = \{e\},\\
K_1.x &= K_1.y = \emptyset = K_1,
\end{aligned}$$

odakle je $A_2 = A_1 \cup \{L_2, K_2\}$, gde je $L_2 = \{x^{n-2}y^n \,|\, n \geqslant 2\}$ i $K_2 = \{e\}$.

Dalje je

$$\begin{aligned} L_2.x &= \{u \in X^* \,|\, xu \in L_2\} = \{x^{n-3}y^n \,|\, n \geqslant 3\}, \\ L_2.y &= \{u \in X^* \,|\, yu \in L_2\} = \{y\}, \\ K_2.x &= \{u \in X^* \,|\, xu \in K_2\} = \{u \in X^* \,|\, xu = e\} = \emptyset = K_1, \\ K_2.y &= \{u \in X^* \,|\, yu \in K_2\} = \{u \in X^* \,|\, yu = e\} = \emptyset = K_1, \end{aligned}$$

pa je $A_3 = A_2 \cup \{L_3, K_3\}$, gde je $L_3 = \{x^{n-3}y^n \,|\, n \geqslant 3\}$ i $K_3 = \{y\}$.

Nastavljajući na isti način u sledećem koraku dobijamo da je

$$\begin{aligned} L_3.x &= \{u \in X^* \,|\, xu \in L_3\} = \{x^{n-4}y^n \,|\, n \geqslant 4\}, \\ L_3.y &= \{u \in X^* \,|\, yu \in L_3\} = \{y^2\}, \\ K_3.x &= \{u \in X^* \,|\, xu \in K_3\} = \{u \in X^* \,|\, xu = y\} = \emptyset = K_1, \\ K_3.y &= \{u \in X^* \,|\, yu \in K_3\} = \{u \in X^* \,|\, yu = y\} = \{e\} = K_2, \end{aligned}$$

pa je $A_4 = A_3 \cup \{L_4, K_4\}$, gde je $L_4 = \{x^{n-4}y^n \,|\, n \geqslant 4\}$ i $K_4 = \{y^2\}$.

Sada već možemo zaključiti da za proizvoljan $m \in \mathbb{N}$ važi

$$A_m = A_{m-1} \cup \{L_m, K_m\}, \tag{2.2}$$

gde su L_m i K_m zadati sa

$$\begin{aligned} L_m &= \{x^{n-m}y^n \,|\, n \geqslant m\}, \\ K_m &= \{y^{m-2}\}, \text{ za } m \geqslant 2, \qquad K_1 = \emptyset. \end{aligned}$$

To ćemo dokazati indukcijom.

Pretpostavimo da je naše tvrđenje tačno. Tada je

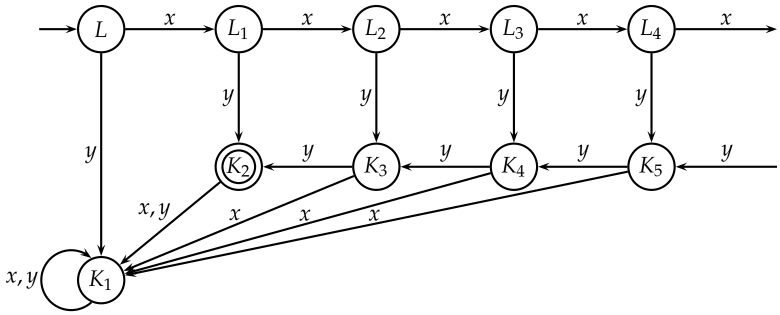
$$\begin{aligned} L_m.x &= \{u \in X^* \,|\, xu \in L_m\} = \{x^{n-m-1}y^n \,|\, n \geqslant m+1\} = L_{m+1}, \\ L_m.y &= \{u \in X^* \,|\, yu \in L_m\} = \{y^{m-1}\} = K_{m+1}, \\ K_m.x &= \{u \in X^* \,|\, xu \in K_m\} = \{u \in X^* \,|\, xu = y^{m-2}\} = \emptyset = K_1, \\ K_m.y &= \{u \in X^* \,|\, yu \in K_m\} = \{u \in X^* \,|\, yu = y^{m-2}\} = \{y^{m-1}\} = K_{m+1}, \end{aligned}$$

odakle dobijamo da je $A_{m+1} = A_m \cup \{L_{m+1}, K_{m+1}\}$.

Iz svega zaključujemo da ima beskonačno mnogo razlomaka datog jezika $L = \{x^n y^n \,|\, n \in \mathbb{N}\}$, kao i da je $\mathcal{A}_L$ automat sa beskonačno mnogo stanja, koji se može grafički predstaviti kao na Slici 2.1.

Kako je K_2 jedini razlomak koji sadrži praznu reč, to je $T_L = \{K_2\}$, pa $\mathcal{A}_L$ raspoznaje jezik L stanjem K_2. □

Zadatak 2.78. *Za jezik $L = \{x^n y^n \,|\, n \in \mathbb{N}\}$ nad alfabetom $X = \{x, y\}$ ne postoji konačan automat koji ga raspoznaje. Dokazati.*



Slika 2.1 Automat $\mathcal{A}_L$ jezika $L = \{x^n y^n \mid n \in \mathbb{N}\}$.

Rešenje: Jasno je da ovo tvrđenje sledi direktno iz prethodnog zadatka, jer je $\mathcal{A}_L$ konstruisan u prethodnom zadatku, minimalan među automatima koji raspoznaju L, a on je beskonačan. Ovde dajemo i drugačije, direktno rešenje.

Pretpostavićemo suprotno, da postoji konačan automat koji raspoznaje L. Neka je to automat $\mathcal{A} = (A, a_0, X, \delta, T)$.

Uvedimo oznaku $a_n = \delta(a_0, x^n)$, za $n \in \mathbb{N}$. Kako smo pretpostavili da je A konačan skup, to je $a_m = a_n$, za neke $m, n \in \mathbb{N}$, takve da je $m \neq n$. Međutim, u ovom slučaju dobijamo:

$$\begin{aligned}\delta(a_0, x^m y^n) &= \delta(\delta(a_0, x^m), y^n) = \delta(a_m, y^n)\\ &= \delta(a_n, y^n) = \delta(\delta(a_0, x^n), y^n) = \delta(a_0, x^n y^n) \in T,\end{aligned}$$

a to znači da je $x^m y^n \in L$, što je u suprotnosti sa definicijom jezika L.

Dakle, ne postoji konačan automat koji raspoznaje L. □

2.3. Minimizacija automata bez izlaza

Razmotrimo situaciju kada minimalni automat jezika L treba konstruisati polazeći ne od jezika L, već od nekog automata $\mathcal{A}$ koji raspoznaje taj jezik.

Drugim rečima, treba redukovati broj stanja automata tako da se dobije automat sa minimalnim brojem stanja koji raspoznaje isti jezik kao i polazni automat.

Postupak konstrukcije minimalnog automata $\mathcal{A}_L$ jezika L, polazeći od automata $\mathcal{A}$ koji raspoznaje L, naziva se *minimizacija automata* $\mathcal{A}$.

Neka je dat automat $\mathcal{A} = (A, a_0, X, \delta, T)$ sa inicijalnmim stanjem a_0 i skupom završnih stanja T.

Za stanje $a \in A$ kažemo da je *dostižno stanje* ako postoji reč $u \in X^*$ tako da je $\delta(a_0, u) = a$. U suprotnom, a je *nedostižno stanje*.

Drugim rečima, stanje a je dostižno ako se do njega može stići iz inicijalnog stanja, odnosno, ako u grafu prelaza automata postoji put iz inicijalnog stanja a_0 u stanje a.

Automat čija su sva stanja dostižna zovemo *dostižan automat*. Definišimo automat $\mathcal{A}^d = (A^d, a_0, X, \delta^d, T^d)$ na sledeći način:

A^d je skup svih dostižnih stanja automata A;
$\delta^d : A^d \times X \to A^d$ je restrikcija preslikavanja δ na $A^d \times X$;
$T^d = T \cap A^d$, tj. T^d je skup svih dostižnih završnih stanja od A.

Uočimo da je inicijalno stanje dostižno, tj. $a_0 \in A^d$.

Za proizvoljne $a \in A^d$ i $x \in X$ imamo da je $a = \delta(a_0, u)$, za neku reč $u \in X^*$, pa je $\delta(a,x) = \delta(\delta(a_0,u),x) = \delta(a_0,ux)$, što znači da je $\delta(a,x) \in A^d$.

Dakle, δ^d zaista slika $A^d \times X$ u A^d, pa je $\mathcal{A}^d$ automat, koji zovemo *dostižni deo* automata $\mathcal{A}$.

Jasno, dostižni deo automata je dostižan automat.

Napomena 3. *Dostižni deo $\mathcal{A}^d$ se dobija iz automata $\mathcal{A}$ na veoma jednostavan način: brisanjem svih njegovih nedostižnih stanja i svih prelaza koji polaze iz nedostižnog stanja ili se završavaju u njemu.*

Dakle, odbacivanjem nedostižih stanja ništa ne menjamo u radu automata, a pri tome automat pojednostavljujemo smanjujući mu broj stanja.

Teorema 2.3. *Za proizvoljan automat $\mathcal{A}$ i njegov dostižni deo $\mathcal{A}^d$ je*

$$L(A) = L(A^d).$$

Prema prethodnom zaključujemo da se nalaženje minimalnog automata $\mathcal{A}_L$ datog jezika L svodi na nalaženje svih razlomaka jezika L.

Daćemo dva algoritma za minimizaciju datog automata.

Prvi algoritam za minimizaciju automata blizak je algoritmu za određivanje minimalnog automata datog jezika.

Krećemo od proizvoljnog automata $\mathcal{A} = (A, a_0, X, \delta, T)$.

Za $a \in A$, sa $T.a$ označićemo skup

$$T.a = \{u \in X^* \mid \delta(a,u) \in T\}.$$

Po analogiji sa razlomcima jezika, skup $T.a$ nazivaćemo *razlomak skupa T* određen stanjem a automata A.

Mada je T skup stanja, a a stanje automata A, razlomak $T.a$ nije skup stanja, već jezik u X^*.

Skup svih razlomaka skupa T označićemo sa A_T.

Teorema 2.4. *Neka je dat automat $\mathcal{A} = (A, a_0, X, \delta, T)$. Tada za proizvoljne $a \in A$ i $u \in X^*$ važi*

$$T.\delta(a,u) = (T.a).u\,.$$

Osim toga, ako je $\mathcal{A}$ dostižan automat i L je jezik koji taj automat raspoznaje, tada je $A_L = A_T$ (skup svih razlomaka jezika L jednak je skupu svih razlomaka skupa T).

Algoritam za konstrukciju minimalnog automata jezika $L \subseteq X^*$ koji se zasniva na nalaženju svih razlomaka jezika L, može se primeniti i za minimizaciju automata $\mathcal{A} = (A, a_0, X, \delta, T)$ koji raspoznaje jezik L.

Naime, najpre bi bio određen jezik L, a zatim i njegovi razlomci.

Međutim, u slučaju kada je jezik zadat preko automata $\mathcal{A}$ koji ga raspoznaje, onda često može biti mnogo lakše nalaziti razlomke jezika kao razlomke skupa stanja T, korišćenjem prethodne teoreme.

Prema tome, postupak za konstrukciju minimalnog automata jezika L polazeći od automata $\mathcal{A}$ sastoji se u sledećem:

1. Najpre nalazimo dostižni deo $\mathcal{A}^d$ automata $\mathcal{A}$.
   Kao što znamo, automat $\mathcal{A}^d$ raspoznaje L skupom $T^d = T \cap A^d$.
2. U automatu $\mathcal{A}^d$ nalazimo skup razlomaka skupa T^d. Kako je $\mathcal{A}^d$ dostižan automat, to prema prethodnoj teoremi imamo da je skup razlomaka skupa stanja T^d jednak skupu A_L razlomaka jezika L.

Naredni algoritam za minimizaciju automata koji raspoznaje dati jezik koristi ideje i metode koji dolaze iz algebre, odnosno, dva centralna algebarska koncepta: *kongruencije* i *homomorfizme*.

Podsetimo se da proizvoljna *relacija ekvivalencije* ϱ na skupu A razbija taj skup na međusobno disjunktne podskupove – klase ekvivalencije.

Relacija ekvivalencije grupiše, udružuje u jednu klasu sve one elemente koje objedinjuje neko zajedničko svojstvo – ono koje opisuje ta relacija.

Faktor skup A/ϱ je skup klasa ekvivalencije relacije ekvivalencije ϱ. Skup A/ϱ ima manji broj elemenata nego A. Neka sada A ima izvesnu algebarsku strukturu, odnosno, neka je na A definisan izvestan sistem operacija.

Da bi operacije sa skupa A mogli da na prirodan način prenesemo na faktor skup A/ϱ, nije dovoljno da ϱ bude samo relacija ekvivalencije, već je potrebno da relacija ekvivalencije ϱ bude *saglasna* sa operacijama na A, i takve relacije ekvivalencije zovemo *kongruencije*. Kongruencije imaju dvojaku ulogu:

- treba da grupišu sve elemente sa izvesnim zajedničkim svojstvom i sažmu ih u jedan element i da time redukuju broj elemenata iz A;
- treba da omoguće da se operacije sa A na prirodan način prenesu na odgovarajući faktor skup.

Kada sve elemente iz A sa izvesnim zajedničkim svojstvom grupišemo u klasu, i potom toj klasi pridružimo određeni element faktor skupa A/ϱ, dobijamo preslikavanje iz A u A/ϱ, koje zovemo *prirodno preslikavanje* relacije ekvivalencije ϱ.

Ako je ϱ saglasna sa operacijama na A, onda je i njeno prirodno preslikavanje na izvestan način saglasno sa operacijama na A, odnosno, prenosi izvesna algebarska svojstva sa A na A/ϱ.

Takva preslikavanja, saglasna sa operacijama, koja prenose izvesna algebarska svojstva, zovemo *homomorfizmi*.

Funkcija prelaza automata u strogom smislu nije algebarska operacija, ali ima izvesna svojstva bliska algebarskim operacijama. To nam omogućava da po analogiji sa odgovarajućim algebarskim pojmovima, definišemo pojmove *kongruencije* i *homomorfizma* automata.

Neka je dat automat $\mathcal{A} = (A, a_0, X, \delta, T)$ i relacija ekvivalencije ϱ na skupu stanja A tog automata.

Za ϱ kažemo da je *kongruencija* na automatu A ako je *saglasna* sa funkcijom prelaza δ, odnosno ako važi sledeće:

za proizvoljne $a, b \in A$, iz $(a, b) \in \varrho$ sledi $(\delta(a, x), \delta(b, x)) \in \varrho$, za svaki $x \in X$.

Jednostavno se dokazuje da prethodni uslov , za proizvoljne $a, b \in A$, iz $(a, b) \in \varrho$ povlači $(\delta(a, u), \delta(b, u)) \in \varrho$, za svaki $u \in X^*$.

Ako je ϱ kongruencija na automatu $\mathcal{A}$, tada slično kao kod algebarskih struktura uvodimo pojam faktor-automata na sledeći način:

Na faktor-skupu A/ϱ definišemo preslikavanje

$$\delta_\varrho : (A/\varrho) \times X \to A/\varrho,$$

sa

$$\delta_\varrho(a\varrho, x) = (\delta(a, x))\varrho$$

za sve $a \in A$ i $x \in X$, gde je sa $a\varrho$ označena ϱ-klasa stanja a.

Dakle, $\mathcal{A}/\varrho = (A/\varrho, a_0\varrho, X, \delta_\varrho, T\varrho)$ je automat koji se naziva *faktor-automat* automata $\mathcal{A}$ u odnosu na relaciju kongruencije ϱ, sa skupom završnih stanja $T\varrho = \{a\varrho | a \in T\}$.

Ako su $\mathcal{A} = (A, a_0, X, \delta, T)$ i $\mathcal{A}' = (A', a'_0, X, \delta', T')$ automati, onda je $\varphi : A \to A'$ *homomorfizam* automata $\mathcal{A}$ u automat $\mathcal{A}'$ ako za proizvoljne $a \in A$ i $x \in X$ važi:

$$\varphi(\delta(a, x)) = \delta'(\varphi(a), x),$$
$$\varphi(a_0) = a'_0 \ \text{ i } \ T' = \varphi(T) = \{\varphi(a) | a \in T\}.$$

Jednostavno se dokazuje da, ako je φ homomorfizam, onda za proizvoljno stanje $a \in A$ i ulaznu reč $u \in X^*$ važi $\varphi(\delta(a, u)) = \delta'(\varphi(a), u)$.

Surjektivni homomorfizam automata $\mathcal{A}$ na $\mathcal{A}'$ naziva se *epimorfizam*. U tom slučaju automat $\mathcal{A}'$ zovemo *homomorfna slika* automata $\mathcal{A}$.

Bijektivni homomorfizam $\varphi : \mathcal{A} \to \mathcal{A}'$ zove se *izomorfizam* automata $\mathcal{A}$ i $\mathcal{A}'$, a $\mathcal{A}$ i $\mathcal{A}'$ su *izomorfni automati*.

Drugi algoritam za minimizaciju zasniva se na nalaženju međusobno ekvivalentnih stanja i svaka klasa međusobno ekvivalentnih stanja se zamenjuje samo jednim stanjem.

Za automat $\mathcal{A} = (A, a_0, X, \delta, T)$, relaciju π_T na $\mathcal{A}$ određenu sa T definišemo sa:

$$(a, b) \in \pi_T \ \Leftrightarrow \ T.a = T.b, \ \text{ za } \ a, b \in A.$$

Ovako definisana relacija π_T je kongruencija na $\mathcal{A}$.

Teorema 2.5. *Neka je $A = (A, a_0, X, \delta, T)$ dostižan automat koji raspoznaje jezik $L \subseteq X^*$.*

Tada je faktor-automat $\mathcal{A}/\pi_T$ izomorfan minimalnom automatu $\mathcal{A}_L$ jezika L.

Kongruenciju π_T moguće je konstruisati korišćenjem niza relacija koji se uvodi u narednoj teoremi.

Pre toga, za automat $A = (A, a_0, X, \delta, T)$, neka je ε_T relacija ekvivalencije na A koja ima samo dve klase: T i $A \setminus T$, tj.

$$\varepsilon_T = T \times T \cup (A \setminus T) \times (A \setminus T).$$

Drugim rečima, za proizvoljne $a, b \in A$ važi

$$(a,b) \in \varepsilon_T \quad \Leftrightarrow \quad (a \in T \ \Leftrightarrow \ b \in T).$$

Pokazuje se da među relacijama π_T i ε_T na automatu $\mathcal{A}$ postoji sledeća veza: za proizvoljna stanja $a, b \in A$ važi

$$(a,b) \in \pi_T \quad \Leftrightarrow \quad (\forall u \in X^*)\ (\delta(a,u), \delta(b,u)) \in \varepsilon_T.$$

Teorema 2.6. *Neka je dat automat $\mathcal{A} = (A, a_0, X, \delta, T)$. Definišimo niz relacija $\{\pi_T^{(k)}\}_{k \in \mathbb{N}^0}$ na A sa: $\pi_T^{(0)} = \varepsilon_T$ i*

$$\pi_T^{(k+1)} = \left\{ (a,b) \in \pi_T^{(k)} \ \middle| \ (\forall x \in X)\, (\delta(a,x), \delta(b,x)) \in \pi_T^{(k)} \right\}.$$

Tada važi sledeće:

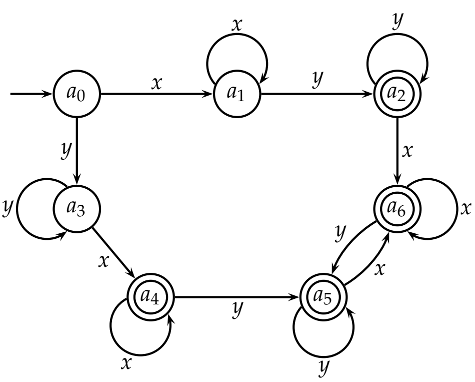
(a) *Svaki član niza $\{\pi_T^{(k)}\}_{k \in \mathbb{N}^0}$ je relacija ekvivalencije na A i važi*

$$\varepsilon_T = \pi_T^{(0)} \supseteq \pi_T^{(1)} \supseteq \ldots \supseteq \pi_T^{(k)} \supseteq \pi_T^{(k+1)} \supseteq \ldots \supseteq \pi_T.$$

(b) *Ako je $\pi_T^{(k)} = \pi_T^{(k+1)}$, za neki $k \in \mathbb{N}^0$, tada je $\pi_T^{(k)} = \pi_T^{(k+m)} = \pi_T$, za svaki $m \in \mathbb{N}^0$.*

(c) *Ako je A konačan automat, tada postoji $k \in \mathbb{N}^0$ tako da je $\pi_T^{(k)} = \pi_T$.*

Zadatak 2.79. *Minimizirati automat sa slike:*



Rešenje: Minimiziraćemo dati automat na sva tri načina, tj. primenom svakog od datih algoritama za minimizaciju.

I način: Da bi našli razlomke jezika L koji se raspoznaje datim automatom, nad alfabetom $X = \{x, y\}$,odredićemo taj jezik. Dati automat prihvata jezik

$$L = \{u \in X^* \,|\, \delta(a_0, u) \in T\} = \{x\}^+\{y\}^+X^* \cup \{y\}^+\{x\}^+X^*$$

$$\begin{aligned} L.x &= \{u \in X^* \,|\, xu \in L\} = \{x\}^*\{y\}^+X^* = L_1, \\ L.y &= \{u \in X^* \,|\, yu \in L\} = \{y\}^*\{x\}^+X^* = L_2, \end{aligned}$$

pa je $A_1 = \{L, L_1, L_2\}$. Dalje, određujemo skup A_2:

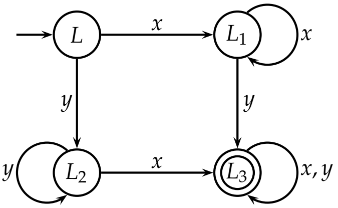
$$\begin{aligned} L_1.x &= \{u \in X^* \,|\, xu \in L_1\} = L_1 \\ L_1.y &= \{u \in X^* \,|\, yu \in L_1\} = \{y\}^*X^* = X^* = L_3, \\ L_2.x &= \{u \in X^* \,|\, xu \in L_2\} = \{x\}^*X^* = X^* = L_3, \\ L_2.y &= \{u \in X^* \,|\, yu \in L_2\} = L_2, \end{aligned}$$

pa je $A_2 = \{L, L_1, L_2, L_3\}$. Određujemo istim postupkom skup A_3 i dobijamo

$$\begin{aligned} L_3.x &= \{u \in X^* \,|\, xu \in L_3\} = X^* = L_3, \\ L_3.y &= \{u \in X^* \,|\, yu \in L_3\} = X^* = L_3, \end{aligned}$$

pa je $A_3 = A_2$. Prema tome, $A_L = A_2 = \{L, L_1, L_2, L_3\}$,

Dakle, automat $\mathcal{A}_L$ je zadat grafom:



Kako je $L_3 = X^*$ jedini razlomak iz skupa A_L koji sadrži praznu reč e, to je $T_L = \{L_3\}$.

II način: Dati automat je dostižan, pa ćemo odmah formirati razlomke skupa završnih stanja $T = \{a_2, a_4, a_5, a_6\}$ određene stanjima automata:

$$\begin{aligned} T.a_0 &= \{u \in X^* \,|\, \delta(a_0, u) \in T\} = \{x\}^+\{y\}^+X^* \cup \{y\}^+\{x\}^+X^* = L, \\ T.a_1 &= \{u \in X^* \,|\, \delta(a_1, u) \in T\} = \{x\}^*\{y\}^+X^* = L_1, \\ T.a_2 &= \{u \in X^* \,|\, \delta(a_2, u) \in T\} = X^* = L_3, \\ T.a_3 &= \{u \in X^* \,|\, \delta(a_3, u) \in T\} = \{y\}^*\{x\}^+X^* = L_2, \\ T.a_4 &= T.a_5 = T.a_6 = L_3. \end{aligned}$$

Tako dobijamo da je skup $A_T^{(1)} = \{L, L_1, L_2, L_3\}$. Nastavljajući postupak dobijamo:

$$
\begin{aligned}
L.x &= (T.a_0).x = T.\delta(a_0,x) = T.a_1 = L_1,\\
L.y &= (T.a_0).y = T.\delta(a_0,y) = T.a_3 = L_2,\\
L_1.x &= (T.a_1).x = T.\delta(a_1,x) = T.a_1 = L_1,\\
L_1.y &= (T.a_1).y = T.\delta(a_1,y) = T.a_2 = L_3,\\
L_2.x &= (T.a_3).x = T.\delta(a_3,x) = T.a_4 = L_3,\\
L_2.y &= (T.a_3).y = T.\delta(a_3,y) = T.a_3 = L_3,\\
L_3.x &= (T.a_2).x = T.\delta(a_2,x) = T.a_6 = L_3,\\
L_3.y &= (T.a_2).y = T.\delta(a_2,y) = T.a_2 = L_3,
\end{aligned}
$$

pa je $A_T^{(2)} = A_T^{(1)} = A_T$. Na ovaj način dobili smo isti atomat, sa skupom stanja $A_T = A_L = \{L, L_1, L_2, L_3\}$, čiji je skup finalnih stanja $T = \{L_3\}$.

III način: Formirajmo, najpre, listu P svih parova stanja automata A i predstavimo je tablicom parova.

Kako su relacije koje generišemo refleksivne i simetrične, to posmatramo samo deo tablice ispod glavne dijagonale.

1. korak: Sva stanja datog automata su dostižna tako da u ovom koraku nema brisanja sa liste.

2. korak: Izbacujemo iz liste P sve parove iz skupa $T \times (A \setminus T) \cup (A \setminus T) \times T$.

	a_0	a_1	a_2	a_3	a_4	a_5	a_6
a_0			×		×	×	×
a_1			×		×	×	×
a_2	×	×		×			
a_3			×		×	×	×
a_4	×	×		×			
a_5	×	×		×			
a_6	×	×		×			

posle 2. koraka – relacija $\pi_T^{(0)} = \varepsilon_T$

3. korak: Proveravamo parove koji su posle 2. koraka ostali na listi:

$(\delta(a_1,x), \delta(a_0,x)) = (a_1,a_1)$ – na listi je;
$(\delta(a_1,y), \delta(a_0,y)) = (a_4,a_3)$ – nije na listi,
brišemo parove (a_0,a_1) i (a_1,a_0);
$(\delta(a_3,x), \delta(a_0,x)) = (a_4,a_1)$ – nije na listi;
brišemo parove (a_0,a_3) i (a_3,a_0);
$(\delta(a_3,x), \delta(a_1,x)) = (a_4,a_1)$ – nije na listi,
brišemo parove (a_1,a_3) i (a_3,a_1);
$(\delta(a_4,x), \delta(a_2,x)) = (a_4,a_6)$ – na listi je,
$(\delta(a_4,y), \delta(a_2,y)) = (a_5,a_2)$ – na listi je;
$(\delta(a_5,x), \delta(a_2,x)) = (a_6,a_6)$ – na listi je,
$(\delta(a_5,y), \delta(a_2,y)) = (a_5,a_2)$ – na listi je;

$(\delta(a_6,x),\delta(a_2,x)) = (a_6,a_6)$ – na listi je,
$(\delta(a_6,y),\delta(a_2,y)) = (a_5,a_2)$ – na listi je;
$(\delta(a_5,x),\delta(a_4,x)) = (a_6,a_4)$ – na listi je,
$(\delta(a_5,y),\delta(a_4,y)) = (a_5,a_5)$ – na listi je;
$(\delta(a_6,x),\delta(a_4,x)) = (a_6,a_4)$ – na listi je,
$(\delta(a_6,y),\delta(a_4,y)) = (a_5,a_5)$ – na listi je;
$(\delta(a_6,x),\delta(a_5,x)) = (a_6,a_6)$ – na listi je,
$(\delta(a_6,y),\delta(a_4,y)) = (a_5,a_5)$ – na listi je;

Lista P sada ima sledeći izgled:

	a_0	a_1	a_2	a_3	a_4	a_5	a_6
a_0		X	X	X	X	X	X
a_1	X		X	X	X	X	X
a_2	X	X		X			
a_3	X	X	X		X	X	X
a_4	X	X		X			
a_5	X	X		X			
a_6	X	X		X			

posle 3. koraka – relacija $\pi_T^{(1)}$

4. korak: U ovom koraku proveravamo parove iz skupa $\{a_2,a_4,a_5,a_6\}$ koji su jedini ostali na listi.

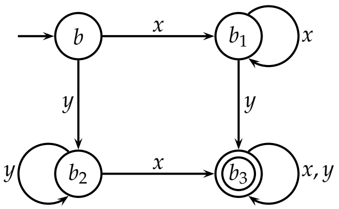
$(\delta(a_4,x),\delta(a_2,x)) = (a_4,a_6)$ – na listi je,
$(\delta(a_4,y),\delta(a_2,y)) = (a_5,a_2)$ – na listi je;
$(\delta(a_5,x),\delta(a_2,x)) = (a_6,a_6)$ – na listi je,
$(\delta(a_5,y),\delta(a_2,y)) = (a_5,a_2)$ – na listi je;
$(\delta(a_6,x),\delta(a_2,x)) = (a_6,a_6)$ – na listi je,
$(\delta(a_6,y),\delta(a_2,y)) = (a_5,a_2)$ – na listi je;
$(\delta(a_5,x),\delta(a_4,x)) = (a_6,a_4)$ – na listi je,
$(\delta(a_5,y),\delta(a_4,y)) = (a_5,a_5)$ – na listi je;
$(\delta(a_6,x),\delta(a_4,x)) = (a_6,a_4)$ – na listi je,
$(\delta(a_6,y),\delta(a_4,y)) = (a_5,a_5)$ – na listi je;
$(\delta(a_6,x),\delta(a_5,x)) = (a_6,a_6)$ – na listi je,
$(\delta(a_6,y),\delta(a_4,y)) = (a_5,a_5)$ – na listi je;

Kako su ovi parovi na listi, u ovom koraku nema brisanja.

To znači da je algoritam završen i da tražena relacija $\pi_T = \pi_T^{(1)} = \pi_T^{(2)}$ na automatu $\mathcal{A}$ ima sledeće klase:

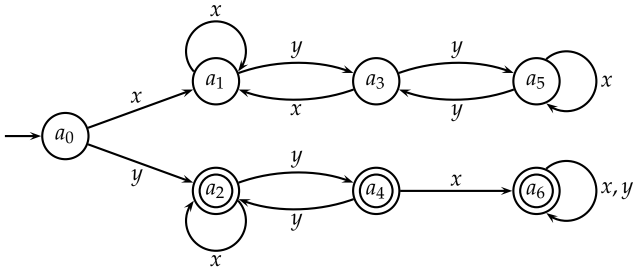
$$\{a_0\},\ \{a_1\},\ \{a_3\},\ \{a_2,a_4,a_5,a_6\}.$$

Traženi minimalni automat predstavljen je grafom:



Njegova stanja su $b = \{a_0\}$, $b_1 = \{a_1\}$, $b_2 = \{a_3\}$, $b_3 = \{a_2, a_4, a_5, a_6\}$, a skup završnih stanja je $T = \{b_3\}$. $\square$

Zadatak 2.80. *Minimizirati automat $\mathcal{A} = (A, a_0, X, \delta, T)$ predstavljen sledećim grafom:*



Rešenje: Primetimo da automat $\mathcal{A}$ raspoznaje jezik $L = yX^*$.

Zaista, ako je $u \in L = T.a_0$, tj. $\delta(a_0, u) \in T$, tada je jasno da mora biti $u = yv$, za neku reč $v \in X^*$. Prema tome, $L \subseteq yX^*$.

Sa druge strane, za proizvoljnu reč $v \in X^*$ imamo da je

$$\delta(a_0, yv) = \delta(\delta(a_0, y), v) = \delta(a_2, v) \in T,$$

jer je $a_2 \in T$ i skup stanja T je zatvoren za sve prelaze, tj. iz skupa T nema izlaza. Dakle, $yv \in L$, što znači da je $yX^* \subseteq L$. Time smo dokazali da je $L = yX^*$.

Lako je uočiti da je

$$T.a_2 = T.a_4 = T.a_6 = X^*$$
$$T.a_1 = T.a_3 = T.a_5 = \emptyset.$$

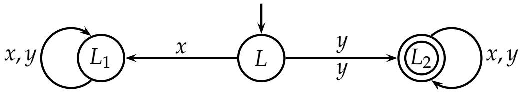
Dakle, $A_L = A_T = \{L, L_1, L_2\}$, gde je

$$L_1 = \emptyset = T.a_1 = T.a_3 = T.a_5$$
$$L_2 = X^* = T.a_2 = T.a_4 = T.a_6.$$

Takođe, imamo da je

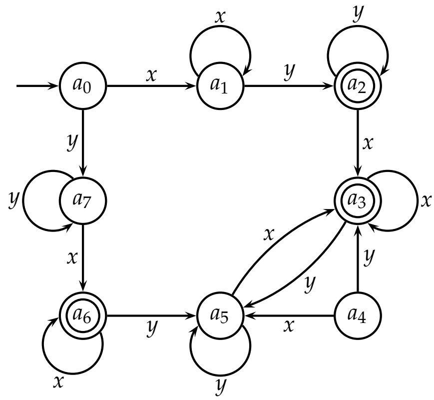
$$\begin{aligned}
L.x &= (T.a_0).x = T.\delta(a_0, x) = T.a_1 = L_1, \\
L.y &= (T.a_0).y = T.\delta(a_0, y) = T.a_2 = L_2, \\
L_1.x &= (T.a_1).x = T.\delta(a_1, x) = T.a_1 = L_1, \\
L_1.y &= (T.a_1).y = T.\delta(a_1, y) = T.a_3 = L_1, \\
L_2.x &= (T.a_2).x = T.\delta(a_2, x) = T.a_2 = L_2, \\
L_2.y &= (T.a_2).y = T.\delta(a_2, y) = T.a_4 = L_2.
\end{aligned}$$

Prema tome, minimalni automat koji raspoznaje jezik L predstaljen je grafom:



Uočimo da je $L_2 = X^*$ jedini element skupa A_L koji sadrži praznu reč e. Dakle, automat $\mathcal{A}_L$ raspoznaje L skupom $T_L = \{L_2\}$, tj. stanjem L_2. $\square$

Zadatak 2.81. *Minimizirati automat* $\mathcal{A} = (A, a_0, X, \delta, T)$ *predstavljen grafom prelaza:*



Rešenje: Minimizaciju možemo izvršiti primenom bilo kog od datih algoritama.

1. korak: Izbacujemo iz liste P sve parove iz vrste i kolone koje odgovaraju stanju a_4, jer je ono nedostižno.

	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_0					✕			
a_1					✕			
a_2					✕			
a_3					✕			
a_4	✕	✕	✕	✕	✕	✕	✕	✕
a_5					✕			
a_6					✕			
a_7					✕			

Dakle, $A^d = \{a_0, a_1, a_2, a_3, a_5, a_6, a_7\}$.

2. korak: Izbacujemo iz liste P sve parove iz skupa $T \times (A^d \setminus T) \cup (A^d \setminus T) \times T$. Vidimo da je

$$T = \{a_2, a_3, a_6\}, \quad A^d \setminus T = \{a_0, a_1, a_5, a_7\}.$$

	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_0		X	X	X	X	X	X	X
a_1	X		X	X	X	/	X	X
a_2	X	X			X	X	X	X
a_3	X	X			X	X		X
a_4	X	X	X	X	X	X	X	X
a_5	X	X	X	X	X		X	
a_6	X	X	X		X	X		X
a_7	X	X	X	X	X		X	

2.3. Lista P

3. korak: Proveravamo parove koji su posle 2. koraka ostali na listi:

$(\delta(a_1,x),\delta(a_0,x)) = (a_1,a_1)$ – na listi je;
$(\delta(a_1,y),\delta(a_0,y)) = (a_2,a_7)$ – nije na listi,
brišemo parove (a_1,a_0) i (a_0,a_1);
$(\delta(a_3,x),\delta(a_2,x)) = (a_3,a_3)$ – na listi je;
$(\delta(a_3,y),\delta(a_2,y)) = (a_5,a_2)$ – nije na listi,
brišemo parove (a_3,a_2) i (a_2,a_3);
$(\delta(a_5,x),\delta(a_0,x)) = (a_3,a_1)$ – nije na listi,
brišemo parove (a_5,a_0) i (a_0,a_5);
$(\delta(a_5,x),\delta(a_1,x)) = (a_3,a_1)$ – nije na listi,
brišemo parove (a_5,a_1) i (a_1,a_5);
$(\delta(a_6,x),\delta(a_2,x)) = (a_6,a_3)$ – na listi je;
$(\delta(a_6,y),\delta(a_2,y)) = (a_5,a_2)$ – nije na listi,
brišemo parove (a_6,a_2) i (a_2,a_6);
$(\delta(a_6,x),\delta(a_3,x)) = (a_6,a_3)$ – na listi je;
$(\delta(a_6,y),\delta(a_3,y)) = (a_5,a_5)$ – na listi je;
$(\delta(a_7,x),\delta(a_0,x)) = (a_6,a_1)$ – nije na listi,
brišemo parove (a_7,a_0) i (a_0,a_7);
$(\delta(a_7,x),\delta(a_1,x)) = (a_6,a_1)$ – nije na listi,
brišemo parove (a_7,a_1) i (a_1,a_7);
$(\delta(a_7,x),\delta(a_5,x)) = (a_6,a_3)$ – na listi je;
$(\delta(a_7,y),\delta(a_5,y)) = (a_7,a_5)$ – na listi je.

Dakle, sa liste brišemo parove

(a_1,a_0), (a_0,a_1), (a_3,a_2), (a_2,a_3), (a_5,a_0), (a_0,a_5), (a_5,a_1), (a_1,a_5), (a_6,a_2), (a_2,a_6), (a_7,a_0), (a_0,a_7), (a_7,a_1), (a_1,a_7)

tako da lista P sada ima izgled predstavljen na slici 2.3.

4. korak: U ovom koraku proveravamo parove (a_6,a_3) i (a_7,a_5) koji su jedini ostali na listi.

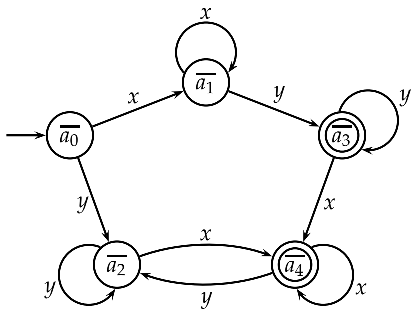
$(\delta(a_6,x),\delta(a_3,x)) = (a_6,a_3)$,
$(\delta(a_6,y),\delta(a_3,y)) = (a_5,a_5)$,
$(\delta(a_7,x),\delta(a_5,x)) = (a_6,a_3)$,
$(\delta(a_7,y),\delta(a_5,y)) = (a_7,a_5)$,

Kako su ovi parovi na listi, u ovom koraku nema brisanja.

To znači da je algoritam završen i da tražena relacija π_T na automatu A^d ima sledeće klase:

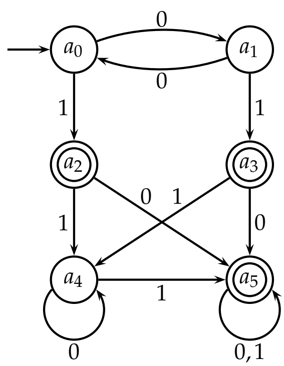
$$\{a_0\},\ \{a_1\},\ \{a_2\},\ \{a_3,a_6\},\ \{a_5,a_7\}.$$

Dakle minimalni automat se može predstaviti grafom:



Stanja dobijenig automata su $\overline{a_0} = \{a_0\}$, $\overline{a_1} = \{a_1\}$, $\overline{a_2} = \{a_2\}$, $\overline{a_3} = \{a_3,a_6\}$, $\overline{a_4} = \{a_5,a_7\}$. □

Zadatak 2.82. *Minimizirati automat sa slike:*



Rešenje: Sva stanja datog automata su dostižna tako da u 1. koraku nema brisanja sa liste.

2. korak: Izbacujemo iz liste P sve parove iz skupa $T \times (A \setminus T) \cup (A \setminus T) \times T$. Vidimo da je $T = \{a_2,a_3,a_5\}$ i $A \setminus T = \{a_0,a_1,a_4\}$.

	a_0	a_1	a_2	a_3	a_4	a_5
a_0			X	X		X
a_1			X	X		X
a_2	X	X			X	
a_3	X	X			X	
a_4			X	X		X
a_5	X	X			X	

posle ovog koraka – relacija $\pi_T^{(0)} = \varepsilon_T$

3. korak: Proveravamo parove koji su posle 2. koraka ostali na listi:

$(\delta(a_1,0),\delta(a_0,0)) = (a_0,a_1)$ – na listi je,
$(\delta(a_1,1),\delta(a_0,1)) = (a_3,a_2)$ – na listi je;
$(\delta(a_4,0),\delta(a_0,0)) = (a_4,a_1)$ – na listi je,
$(\delta(a_4,1),\delta(a_0,1)) = (a_5,a_2)$ – na listi je;
$(\delta(a_4,0),\delta(a_1,0)) = (a_4,a_0)$ – na listi je,
$(\delta(a_4,1),\delta(a_1,1)) = (a_5,a_3)$ – na listi je;
$(\delta(a_3,0),\delta(a_2,0)) = (a_5,a_5)$ – na listi je,
$(\delta(a_3,1),\delta(a_2,1)) = (a_4,a_4)$ – na listi je;
$(\delta(a_5,0),\delta(a_2,0)) = (a_5,a_5)$ – na listi je,
$(\delta(a_5,1),\delta(a_2,1)) = (a_5,a_4)$ – nije na listi;
brišemo parove (a_5,a_2) i (a_2,a_5);
$(\delta(a_5,0),\delta(a_3,0)) = (a_5,a_5)$ – na listi je,
$(\delta(a_5,1),\delta(a_3,1)) = (a_5,a_4)$ – nije na listi;
brišemo parove (a_5,a_3) i (a_3,a_5);

	a_0	a_1	a_2	a_3	a_4	a_5
a_0			X	X	X	X
a_1			X	X	X	X
a_2	X	X			X	X
a_3	X	X			X	X
a_4	X	X	X	X		X
a_5	X	X	X	X	X	

posle petog koraka – relacija $\pi_T^{(3)} = \pi_T^{(2)} = \pi_T$

4. korak: U ovom koraku proveravamo po parovima elemente klasa $\{a_0,a_1,a_4\}$ i $\{a_2,a_3\}$.

$(\delta(a_1,0),\delta(a_0,0)) = (a_0,a_1)$ – na listi je,
$(\delta(a_1,1),\delta(a_0,1)) = (a_3,a_2)$ – na listi je;
$(\delta(a_4,0),\delta(a_0,0)) = (a_4,a_1)$ – na listi je,
$(\delta(a_4,1),\delta(a_0,1)) = (a_5,a_2)$ – nije na listi;
brišemo parove (a_4,a_0) i (a_0,a_4);
$(\delta(a_4,0),\delta(a_1,0)) = (a_4,a_0)$ – na listi je,
$(\delta(a_4,1),\delta(a_1,1)) = (a_5,a_3)$ – nije na listi;
brišemo parove (a_4,a_1) i (a_1,a_4);
$(\delta(a_3,0),\delta(a_2,0)) = (a_5,a_5)$ – na listi je,
$(\delta(a_3,1),\delta(a_2,1)) = (a_4,a_4)$ – na listi je;

5. korak: Proveravamo parove (a_1,a_0) i (a_3,a_2) koji su jedini ostali na listi.

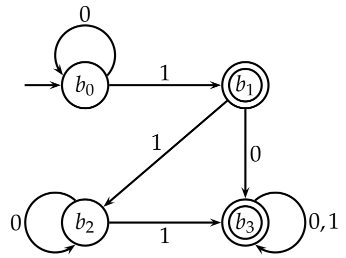
$(\delta(a_1,0),\delta(a_0,0)) = (a_0,a_1)$ – na listi je,
$(\delta(a_1,1),\delta(a_0,1)) = (a_3,a_2)$ – na listi je;
$(\delta(a_3,0),\delta(a_2,0)) = (a_5,a_5)$ – na listi je,
$(\delta(a_3,1),\delta(a_2,1)) = (a_4,a_4)$ – na listi je.

Kako su ovi parovi na listi, u ovom koraku nema brisanja.

To znači da je algoritam završen i da tražena relacija $\pi_T = \pi_T^{(2)} = \pi_T^{(3)}$ na automatu $\mathcal{A}$ ima sledeće klase:

$$\{a_0,a_1\},\ \{a_2,a_3\},\ \{a_4\},\ \{a_5\}.$$

Traženi minimalni automat predstavljen je grafom:



Njegova stanja su $b_0 = \{a_0,a_1\}$, $b_1 = \{a_2,a_3\}$, $b_2 = \{a_4\}$, $b_3 = \{a_5\}$, a skup završnih stanja $T = \{b_1,b_3\}$. □

2.4. Sintaksički monoid jezika

Slično raspoznavanju jezika automatima definiše se i raspoznavanje jezika monoidom.

Monoid S raspoznaje jezik $L \subseteq X^*$ skupom $H \subseteq S$, ako postoji homomorfizam $\varphi : X^* \to S$ takav da je $L = \varphi^{-1}(H)$, tj.

$$L = \{u \in X^* \,|\, \varphi(u) \in H\} \quad \Leftrightarrow \quad (\forall u \in X^*)\ u \in L \ \Leftrightarrow\ \varphi(u) \in H.$$

Može se reći da monoid S raspoznaje jezik L homomorfizmom $\varphi : X^* \to S$ ako je $L = (\varphi \circ \varphi^{-1})(L)$.

Glavna kongruencija na X^* određena jezikom L je relacija definisana sa:

$$(u,v) \in P_L \quad \Leftrightarrow \quad (\forall p,q \in X^*)\ (puq \in L \ \Leftrightarrow\ pvq \in L).$$

Svaki par reči $(p,q) \in X^*$ za koji važi da je $puq \in L$ nazivamo *kontekstom reči $u \in X^*$ u odnosu na jezik L*. Za dve reči u i v koje se javljaju u istim kontekstima u jeziku L kažemo da su *sintaksički ekvivalentne*. Ako na faktor skupu X^*/P_L, u odnosu na operaciju konkatenacije, prirodno definišemo operaciju $''\cdot''$ na sledeći način:

$$uP_L \cdot vP_L = (uv)P_L, \text{ za proizvoljne } uP_L,\ vP_L \in X^*.$$

onda je $(X^*/P_L,\cdot)$ faktor monoid koji se naziva *sintaksički monoid* jezika L i označava se sa $Syn(L)$.

Teorema 2.7. *Za proizvoljan jezik $L \subseteq X^*$, $Syn(L)$ je monoid najmanje kardinalnosti koji raspoznaje jezik L.*

Svakoj reči $u \in X^*$ možemo pridružiti preslikavanje $\eta_u : A \to A$ definisano sa

$$a\eta_u = \delta(a,u),$$

za $a \in A$. Ovo preslikavanje nazivamo *funkcijom prelaza* automata $\mathcal{A}$ *određenom ulaznom reči* u. Jasno, funkcija prelaza η_e određena praznom reči e je identičko preslikavanje skupa A.

Uvedimo sada oznake

$$M(A) = \{\eta_u \,|\, u \in X^*\} \quad \text{i} \quad T_X = \{\eta_x \,|\, x \in X\}.$$

Tada važi:

Teorema 2.8. *Za proizvoljan automat $\mathcal{A}$, $M(A)$ je monoid generisan skupom T_X.*

Monoid $M(A)$ nazivamo *monoidom prelaza* automata $\mathcal{A}$.

Teorema 2.9. *Sintaksički monoid $Syn(L)$ jezika $L \subseteq X^*$ izomorfan je monoidu prelaza minimalnog automata $\mathcal{A}_L$ jezika L.*

Daćemo algoritam za konstrukciju sintaksičkog monoida datog jezika.

Teorema 2.10. *Neka je $L \subseteq X^*$ proizvoljan jezik i $\mathcal{A}_L = (A_L, L, X, \delta_L, T_L)$ minimalni automat koji raspoznaje dati jezik. Definišimo niz $\{H^k\}_{k\in\mathbb{N}^0}$ podskupova monoida $M(A_L)$ tako da je:*

$$H^0 = \{\eta_e\},\ \ H^1 = T_X, \ldots$$
$$H^k = \{\eta_{x_1}\eta_{x_2}\ldots\eta_{x_k} = \eta_{x_1x_2\ldots x_k} \,|\, x_i \in T_X,\ i = \overline{1,k}\}.$$

Zatim, induktivno formiramo niz skupova $\{Y_k\}_{k\in\mathbb{N}^0}$ sa:

$$Y_0 = H^0,\ Y_1 = Y_0 \cup H^1,$$
$$Y_{k+1} = Y_k \cup H^{k+1}, \qquad k \in \mathbb{N}^0.$$

Tada:

(a) Niz $\{Y_k\}_{k\in\mathbb{N}^0}$ je rastući.
(b) Ako postoji $n_0 \in \mathbb{N}^0$ takav da je $Y_{n_0} = Y_{n_0+1}$, tada je $M(A_L) = Y_{n_0} = Syn(L)$.

Zadatak 2.83. *Naći monoid prelaza automata $\mathcal{A}$ sa skupom stanja $A = \{a,b,c\}$, inicijalnim stanjem a i ulaznim alfabetom $X = \{x,y\}$, čija je funkcija prelaza data tablicom:*

δ	a	b	c
x	b	c	a
y	b	b	b

Rešenje: Nalazimo monoid $M(A)$ generisan skupom T_X korišćenjem prethodne teoreme.

$$\eta_e : \begin{pmatrix} a & b & c \\ a & b & c \end{pmatrix}, \quad \eta_x : \begin{pmatrix} a & b & c \\ b & c & a \end{pmatrix}, \quad \eta_y : \begin{pmatrix} a & b & c \\ b & b & b \end{pmatrix}.$$

Dakle, $Y_0 = H^0 = \{\eta_e\}$, $H = \{\eta_x, \eta_y\}$ i $Y_1 = \{\eta_e, \eta_x, \eta_y\}$. Dalje,

$$\eta_{xx} = \eta_x\eta_x : \begin{pmatrix} a & b & c \\ c & a & b \end{pmatrix}, \quad \eta_{xy} = \eta_x\eta_y : \begin{pmatrix} a & b & c \\ b & b & b \end{pmatrix} = \eta_y = \beta,$$

$$\eta_{yx} = \eta_y\eta_x : \begin{pmatrix} a & b & c \\ c & c & c \end{pmatrix} = \gamma, \quad \eta_{yy} = \eta_y\eta_y : \begin{pmatrix} a & b & c \\ b & b & b \end{pmatrix} = \eta_y = \beta.$$

Odavde dobijamo $Y_2 = Y_1 \cup H^2 = \{\eta_e, \eta_x, \eta_y, \eta_{xx}, \eta_{yx}\} = \{\eta_e, \eta_x, \beta, \eta_{xx}, \gamma\}$.

$$\eta_{xxx} = \eta_x\eta_{xx} : \begin{pmatrix} a & b & c \\ a & b & c \end{pmatrix} = \eta_e, \quad \eta_{xxy} = \eta_x\eta_{xy} = \eta_x\beta : \begin{pmatrix} a & b & c \\ b & b & b \end{pmatrix} = \beta,$$

$$\eta_{xyx} = \eta_x\eta_{yx} = \eta_x\gamma : \begin{pmatrix} a & b & c \\ c & c & c \end{pmatrix} = \gamma, \quad \eta_{yxx} = \gamma\eta_x : \begin{pmatrix} a & b & c \\ a & a & a \end{pmatrix} = \eta_y = \alpha,$$

$$\eta_{yxy} = \eta_y\beta = \beta, \quad \eta_{xyy} = \beta\eta_y = \beta, \quad \eta_{yyx} = \beta\eta_x = \gamma, \quad \eta_{yyy} = \beta\eta_y = \beta.$$

Dobili smo skup $Y_3 = Y_2 \cup H^3 = \{\eta_e, \eta_x, \beta, \eta_{xx}, \gamma, \alpha\}$.

$$\eta_{x^4} = \eta_x\eta_{xxx} = \eta_x\eta_e = \eta_x, \qquad \eta_{x^3y} = \eta_e\eta_y = \beta, \quad \eta_{xxyx} = \beta\eta_x = \gamma,$$
$$\eta_{xxyy} = \eta_x\beta = \beta, \qquad \eta_{yx^3} = \alpha\eta_x = \beta, \quad \eta_{yxxy} = \alpha\eta_y = \beta.$$

Vidimo da je $Y_4 = Y_3 \cup H^4 = Y_3$, što znači da se monoid prelaza

$M(A) = Y_3 = \{\eta_e, \eta_x, \beta, \eta_{xx}, \gamma, \alpha\}$ može predstaviti sledećom tablicom:

	η_e	η_x	β	η_{xx}	γ	α
η_e	η_e	η_x	β	η_{xx}	γ	α
η_x	η_x	η_{xx}	β	η_e	γ	α
β	β	γ	β	α	γ	α
η_{xx}	η_{xx}	η_e	β	η_x	γ	α
γ	γ	α	β	β	γ	α
α	α	β	β	γ	γ	α

Operacija na monoidu je kompozicija dobijenih preslikavanja. □

Zadatak 2.84. *Naći sintaksički monoid jezika $L = X^*xyxX^*$.*

Rešenje: Nalaženje sintaksičkog monoida jezika L svodi se na nalaženje monoida prelaza minimalnog automata datog jezika.

$$L.x = \{u \in X^* \mid xu \in L\} = X^*xyxX^* + yxX^* = L_1,$$
$$L.y = \{u \in X^* \mid yu \in L\} = X^*xyxX^* = L,$$

pa je $A_1 = \{L, L_1\}$. Dalje, određujemo skup A_2:

$$L_1.x = \{u \in X^* \mid xu \in L_1\} = X^*xyxX^* + yxX^* = L_1$$
$$L_1.y = \{u \in X^* \mid yu \in L_1\} = X^*xyxX^* + xX^* = L_2,$$

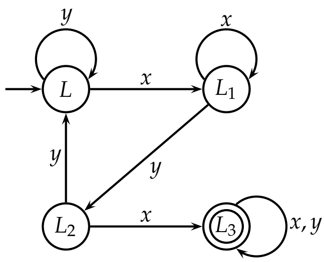
odakle dobijamo $A_2 = \{L, L_1, L_2\}$. Određujemo istim postupkom skup A_3 i dobijamo

$$L_2.x = L.x + X^* = L_1 + X^* = L_3,$$
$$L_2.y = L.y = L,$$

pa je $A_3 = \{L, L_1, L_2, L_3\}$. U narednom koraku imamo:

$$L_3.x = L_1.x + X^* = L_3,$$
$$L_3.y = L_1.y + X^* = L_2 + X^* = L_3,$$

što znači da je $A_4 = A_3$. Prema tome, $A_L = A_3 = \{L, L_1, L_2, L_3\}$. Dakle, automat $\mathcal{A}_L$ je zadat grafom:



Kako $e \in L_3$ skup završnih stanja je $T_L = \{L_3\}$. Dakle, konstruisali smo minimalni automat koji raspoznaje jezik L.

$$\eta_e : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L & L_1 & L_2 & L_3 \end{pmatrix}, \quad \eta_x : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L_1 & L_1 & L_3 & L_3 \end{pmatrix}, \quad \eta_y : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L & L_2 & L & L_3 \end{pmatrix}.$$

Dakle, $Y_0 = H^0 = \{\eta_e\}$, $H = \{\eta_x, \eta_y\}$ i $Y_1 = \{\eta_e, \eta_x, \eta_y\}$. Dalje,

$$\eta_{xx} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L_1 & L_1 & L_3 & L_3 \end{pmatrix} = \eta_x, \quad \eta_{xy} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L_2 & L_2 & L_3 & L_3 \end{pmatrix},$$

$$\eta_{yx} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L_1 & L_3 & L_1 & L_3 \end{pmatrix}, \quad \eta_{yy} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L & L & L & L_3 \end{pmatrix}.$$

Odavde dobijamo $Y_2 = Y_1 \cup H^2 = \{\eta_e, \eta_x, \eta_y, \eta_{xy}, \eta_{yx}, \eta_{yy}\}$.

$$\eta_{xxx} = \eta_x \eta_{xx} = \eta_x, \quad \eta_{xxy} = \eta_{xy}, \quad \eta_{yxx} = \eta_{yx}.$$

$$\eta_{xyx} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L_3 & L_3 & L_3 & L_3 \end{pmatrix} = 3, \quad \eta_{xyy} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L & L & L_3 & L_3 \end{pmatrix},$$

$$\eta_{yxy} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L_2 & L_3 & L_2 & L_3 \end{pmatrix}, \quad \eta_{yyx} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L_1 & L_1 & L_1 & L_3 \end{pmatrix},$$

$$\eta_{yyx} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L & L & L & L_3 \end{pmatrix} = \eta_{yy}.$$

Dobili smo skup $Y_3 = Y_2 \cup H^3 = \{\eta_e, \eta_x, \eta_y, \eta_{xy}, \eta_{yx}, \eta_{yy}, 3, \eta_{xyy}, \eta_{yxy}, \eta_{yyx}\}$.

$$\eta_{xyxx} = \eta_{xyx} = 3, \quad \eta_{xy^3} = \eta_{xyy}, \quad \eta_{yyxx} = \eta_{yyx}$$

$$\eta_{xyxy} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L_3 & L_3 & L_3 & L_3 \end{pmatrix} = 3, \quad \eta_{xyyx} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L_1 & L_1 & L_3 & L_3 \end{pmatrix} = \eta_x,$$

$$\eta_{yxyx} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L_3 & L_3 & L_3 & L_3 \end{pmatrix} = 3, \quad \eta_{yxyy} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L & L_3 & L & L_3 \end{pmatrix},$$

$$\eta_{yyxy} : \begin{pmatrix} L & L_1 & L_2 & L_3 \\ L_2 & L_2 & L_2 & L_3 \end{pmatrix}.$$

Vidimo da je $Y_4 = Y_3 \cup H^4 = \{\eta_e, \eta_x, \eta_y, \eta_{xy}, \eta_{yx}, \eta_{yy}, 3, \eta_{xyy}, \eta_{yxy}, \eta_{yyx}, \eta_{yxyy}, \eta_{yyxy}\}$. Ponavljanjem istog postupka dobijamo da je $Y_5 = Y_4$, što znači da je monoid prelaza automata koji raspoznaje jezik L:

$$M(A) = Y_4 = \{\eta_e, \eta_x, \eta_y, \eta_{xy}, \eta_{yx}, \eta_{yy}, 3, \eta_{xyy}, \eta_{yxy}, \eta_{yyx}, \eta_{yxyy}, \eta_{yyxy}\}.$$

	η_e	η_x	η_y	η_{xy}	η_{yx}	η_{yy}	3	η_{xyy}	η_{yxy}	η_{yyx}	η_{yxyy}	η_{yyxy}
η_e	η_e	η_x	η_y	η_{xy}	η_{yx}	η_{yy}	3	η_{xyy}	η_{yxy}	η_{yyx}	η_{yxyy}	η_{yyxy}
η_x	η_x	η_x	η_{xy}	η_{xy}	3	η_{xyy}	3	η_{xyy}	3	η_x	3	η_{xy}
η_y	η_y	η_{yx}	η_{yy}	η_{yxy}	η_{yyx}	η_{yy}	3	η_{yxy}	η_{yyxy}	η_{yy}	η_{yy}	η_{yyxy}
η_{xy}	η_{xy}	3	η_{xyy}	3	η_x	η_{xyy}	3	3	η_{xy}	η_x	η_{xyy}	η_{xy}
η_{yx}	η_{yx}	η_{yx}	η_{yxy}	η_{yxy}	3	η_{yxyy}	3	η_{yxyy}	3	η_{yx}	3	η_{yxy}
η_{yy}	η_{yy}	η_{yyx}	η_{yy}	η_{yyxy}	η_{yyx}	η_{yy}	3	η_{yy}	η_{yyxy}	η_{yyx}	η_{yy}	η_{yyxy}
3	3	3	3	3	3	3	3	3	3	3	3	3
η_{xyy}	η_{xyy}	η_{xyyx}	η_{xyy}	η_{xy}	η_x	η_{xyy}	3	η_{xyy}	η_{xy}	η_x	η_{xyy}	η_{xy}
η_{yxy}	η_{yxy}	3	η_{yxyy}	3	η_{yx}	η_{yxyy}	3	3	η_{yxy}	η_{yx}	η_{yxyy}	η_{yxy}
η_{yyx}	η_{yyx}	η_{yyx}	η_{yyxy}	η_{yyxy}	3	η_{yy}	3	η_{yx}	3	η_{yyx}	3	η_{yyxy}
η_{yxyy}	η_{yxyy}	η_{yx}	η_{yxyy}	η_{yxy}	η_{yx}	η_{yxyy}	3	η_{yxyy}	η_{yxy}	η_{yx}	η_{yxyy}	η_{yxy}
η_{yyxy}	η_{yyxy}	3	η_{yy}	3	η_{yyx}	η_{yy}	3	3	η_{yyxy}	η_{yyx}	η_{yy}	η_{yyxy}

Prema tome, sintaksički monoid jezika L predstavljen je datom tablicom. □

2.5. Nedeterministički automati

Konačni deterministički automati, kao što smo videli, imaju svojstvo da se iz proizvoljnog stanja a, pod uticajem ulaznog slova x, prelazi u jedno tačno određeno stanje $\delta(a,x)$.

Međutim, mogu se razmatrati i takvi automati kod kojih je iz stanja a, pod uticajem ulaznog slova x, moguće preći u više od jednog stanja, ili ne preći ni u jedno stanje.

Drugim rečima, kod ovakvih automata prelazi nisu jednoznačno određeni, pa takve automate nazivamo *nedeterministički automati*, skraćeno NDA. Formalna matematička definicija NDA je:

Nedeterministički automat je petorka $\mathcal{A} = (A, I, X, \delta, T)$ koju čine:

A – neprazan, konačan skup stanja;
$I \subseteq A$ – skup inicijalnih stanja;
X – neprazan i konačan ulazni alfabet X;
$\delta : A \times X \to \mathcal{P}(A)$ – funkcija prelaza, gde je sa $\mathcal{P}(A)$ označen partitivni skup skupa A, odnosno skup svih podskupova skupa A.
$T \subseteq A$ – skup završnih stanja.

Dakle, kod NDA se može reći da je $\delta(a,x)$ skup stanja "u koje je moguće preći" iz stanja a pod uticajem ulaznog simbola x.

Drugim rečima, iz stanja a se pod uticajem ulaznog simbola x može preći u bilo koje stanje iz skupa $\delta(a,x)$.

Kako $\mathcal{P}(A)$ sadrži i prazan skup, to prema gornjoj definiciji $\delta(a,x)$ može biti i prazan skup. U tom slučaju, iz stanja a se pod uticajem ulaznog simbola x ne može preći ni u jedno drugo stanje.

Primetimo da se deterministički automat može tretirati kao specijalan slučaj nedeterminističkog automata, kod koga su svi podskupovi $\delta(a,x)$ jednoelementni.

Daćemo i drugu ekvivalentnu definiciju nedeterminističkih automata. NDA se može definisati kao uređena petorka $\mathcal{A} = (A, I, X, E, T)$, koju čine:

A – neprazan skup stanja;
$I \subseteq A$ – skup inicijalnih stanja;
X – neprazan i konačan ulazni alfabet ;
$E \subseteq A \times X \times A$ – *relacija prelaza*;
$T \subseteq A$ – skup završnih stanja.

Ovako definisana relacija prelaza znači da se iz stanja a se pod uticajem ulaznog simbola x može preći u stanje b ako i samo ako je $(a,x,b) \in E$. Prema drugoj definiciji, vidimo da je nedeterministički automat zapravo *označeni graf*, pri čemu $(a,x,b) \in E$ možemo shvatiti kao granu grafa između čvorova a i b označenu sa x.

Neka je $\mathcal{A} = (A, I, X, \delta, T)$ dati nedeterministički automat. Za $P \in \mathcal{P}(A)$ i $x \in X$, definišimo $\delta(P,x) \in \mathcal{P}(A)$ sa

$$\delta(P,x) = \bigcup_{a \in P} \delta(a,x).$$

Preslikavanje δ prošireno je sa $A \times X$ na $\mathcal{P}(A) \times X$, poistovećivanjem elemenata iz A sa odgovarajućim jednoelementnim podskupovima iz $\mathcal{P}(A)$:

$$\delta(\{a\},x) = \delta(a,x),$$

za sve $a \in A$ i $x \in X$.

Ovim smo pokazali da se nedeterministički automat $\mathcal{A} = (A, I, X, \delta, T)$ na prirodan način može prevesti u deterministički automat

$$\mathcal{A}^p = (\mathcal{P}(A), I, X, \delta, \mathcal{T}),$$

gde je $\mathcal{T} = \{P \in \mathcal{P}(A) \,|\, P \cap T \neq \emptyset\}$. Automat A^p zvaćemo *determinizacija* nedeterminističkog automata A dobijena *podskupovnom konstrukcijom*.

Jezik $L(A) = \{u \in X^* \,|\, \delta(I, u) \cap T \neq \emptyset\}$, nazivamo *jezikom nedeterminističkog automata* $\mathcal{A}$. Ako je $L = L(A)$, onda kažemo da $\mathcal{A}$ raspoznaje jezik L skupom završnih stanja T.

U tom slučaju jezik L *može biti raspoznat nedeterminističkim automatom* $\mathcal{A}$.

Vezu između jezika koji se mogu raspoznati determinističkim i onih koji se mogu raspoznati nedeterminističkim automatom daje sledeća teorema:

Teorema 2.11. *Neka je* $\mathcal{A} = (A, I, X, \delta, T)$ *nedeterministički automat i*

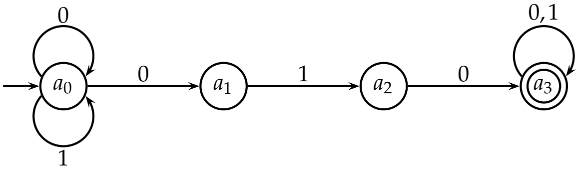
$$\mathcal{A}^p = (\mathcal{P}(A), I, X, \delta, \mathcal{T})$$

je njegova determinizacija. Tada je $L(A) = L(A^p)$.

Teorema 2.12. *Jezik* $L \subseteq X^*$ *može biti raspoznat konačnim determinističkim automatom ako i samo ako može biti raspoznat konačnim nedeterminističkim automatom.*

Zadatak 2.85. *Konstruisati nedeterministički automat koji raspoznaje tačno one reči koje imaju podreč* 010.

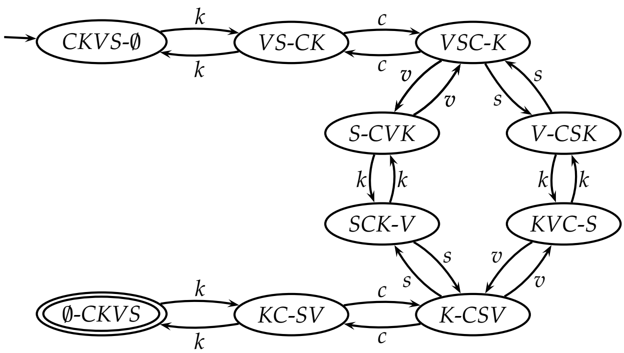
Rešenje: Traženi automat može se predstaviti grafom:



Primetimo, da NDA im jednostavniji oblik od determinističkog automata koji bi prihvatao iste reči. □

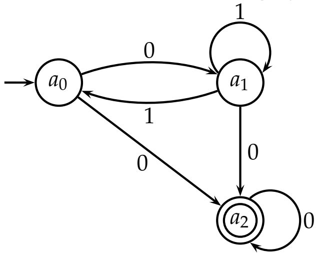
Zadatak 2.86. *Čovek, koza, vuk i salata treba da se prevezu u malom čamcu sa leve na desnu obalu reke. Čamac je mali i čovek (koji naravno jedini zna da upravlja čamcem), može povesti samo kozu, samo vuka ili samo salatu, pri čemu mora biti oprezan i ne sme ostaviti ni kozu sa vukom, ni kozu sa salatom. Konstruisati nedeterministički automat koji rešava opisani problem.*

Rešenje: Označimo slovima C, K, V, S, redom, čoveka, kozu, vuka i salatu. Stanje kada su svi na levoj obali označavaćemo sa $CKVS$-$\emptyset$, kada su vuk i salata na desnoj, a čovek i koza na levoj obali sa VS-CK, stanje kada su svi na desnoj obali sa $\emptyset$-$CKVS$, itd. Ulazni simboli neka budu c, k, v, s, gde ulazak simbola c u dato stanje znači da je čovek seo sam u čamac i sam ide preko reke, a k, v, s znači da se čovek vozi u čamcu, redom, sa kozom, vukom i salatom.



Početno stanje je $CKVS$-$\emptyset$, a završno $\emptyset$-$CKVS$. □

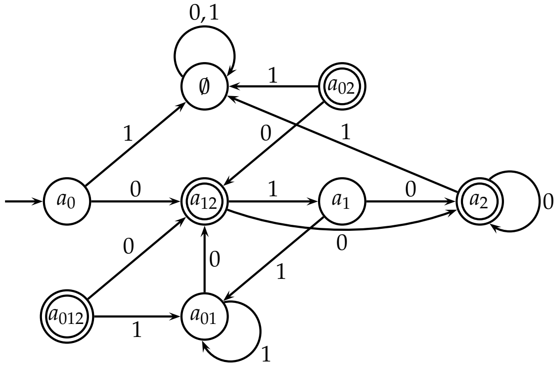
Zadatak 2.87. *Konstruisati deterministički automat koji raspoznaje isti jezik kao nedeterministički automat $\mathcal{A} = (A, I, X, \delta, T)$ dat grafom prelaza:*



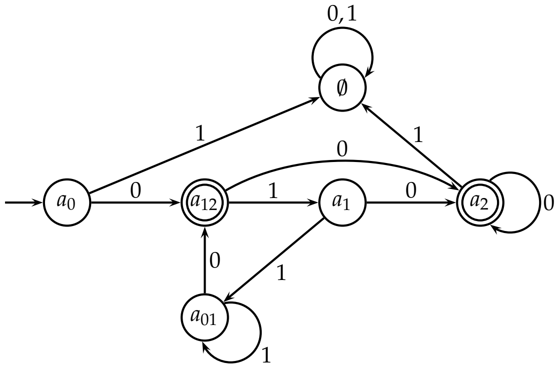
Rešenje: Prema definiciji determinizacije, stanja traženog automata su elementi skupa

$$\mathcal{P}(A) = \{\emptyset, \{a_0\}, \{a_1\}, \{a_2\}, \{a_0, a_1\}, \{a_0, a_2\}, \{a_1, a_2\}, \{a_0, a_1, a_2\}\},$$

inicijalno stanje je $\{a_0\}$, skup finalnih stanja $\tau = \{\{a_2\}, \{a_0, a_2\}, \{a_1, a_2\}, \{a_0, a_1, a_2\}\}$, a prelazi su dati sledećim grafom:

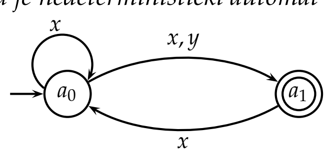


U grafu prelaza stanje a_{ij} označava podskup $\{a_i, a_j\}$ skupa A, za $i, j \in \{0,1,2\}$, dok je a_{012} stanje $a_0a_1a_2$. Primetimo da stanja a_{02} i a_{012} nisu dostižna iz inicijalnog stanja, pa ih možemo jednostavno izbrisati. Graf prelaza traženog determinističkog automata ima oblik:



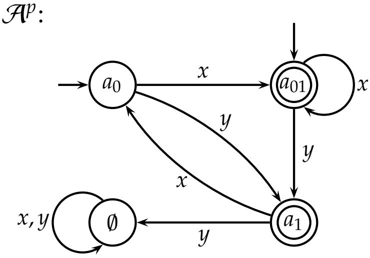
Dakle, pri determinizaciji nedeterminističkog automata možemo odmah eliminisati nedostižne podskupove, i odgovarajuće grane, čime se postupak pojednostavljuje. □

Zadatak 2.88. *Neka je nedeterministički automat $\mathcal{A}$ dat sledećim grafom:*



Naći deterministički automat koji raspoznaje isti jezik kao $\mathcal{A}$.

Rešenje: Determinizacijom automata $\mathcal{A}$ podskupovnom konstrukcijom dobijamo automat $\mathcal{A}^p$:



predstavljen je datim grafom prelaza. □

Zadatak 2.89. *Neka su $\mathcal{A}_1 = (A_1, a_0^1, X, \delta_1, T_1)$ i $\mathcal{A}_2 = (A_2, a_0^2, X, \delta_2, T_2)$ automati koji raspoznaju jezike L_1 i L_2, redom.*

Konstruisati automat $\mathcal{A} = (A, a_0, X, \delta, T)$ koji raspoznaje jezik $L_1 \cup L_2$.

Rešenje: Konstruišimo automat $\mathcal{A} = (A, a_0, X, \delta, T)$ na sledeći način:

Neka je $A = A_1 \times A_2$, $a_0 = (a_0^1, a_0^2)$, dok je funkcija prelaza $\delta : A \times X \to A$ definisana sa

$$\delta((a_1,a_2),x) = (\delta_1(a_1,x),\delta_2(a_2,x)),$$

i $T = (T_1 \times A_2) \cup (A_1 \times T_2)$.

Pokazaćemo da je $L(A) = L_1 \cup L_2$.

Zaista, iz iz definicije funkcije prelaza δ lako se dobija da je

$$\delta((a_1,a_2),u) = (\delta_1(a_1,u),\delta_2(a_2,u)),$$

za proizvoljne $a_1 \in A_1$, $a_2 \in A_2$ i $u \in X^*$, odakle sledi da je

$$\begin{aligned} u \in L_1 \cup L_2 &\Leftrightarrow u \in L_1 \text{ ili } u \in L_2 \\ &\Leftrightarrow \delta_1(a_0,u) \in T_1 \text{ ili } \delta_2(a_0,u) \in T_2 \\ &\Leftrightarrow (\delta_1(a_0,u),\delta_2(a_0,u)) \in (T_1 \times A_2) \cup (A_1 \times T_2) \\ &\Leftrightarrow \delta(a_0,u) \in T \;\Leftrightarrow\; u \in L(A). \end{aligned}$$

Jasno je da ovako konstruisan automat $\mathcal{A}$ raspoznaje jezik $L_1 \cup L_2$. □

Zadatak 2.90. *Neka su* $\mathcal{A}_1 = (A_1,a_0^1,X,\delta_1,T_1)$ *i* $\mathcal{A}_2 = (A_2,a_0^2,X,\delta_2,T_2)$ *dati automati koji, redom, raspoznaju jezike* $L_1,L_2 \subseteq X^*$.

Konstruisati automat $\mathcal{A} = (A,a_0,X,\delta,T)$ *koji raspoznaje jezik* $L_1 \cap L_2$.

Rešenje: Jednostavno se pokazuje da automat $\mathcal{A}$ definisan u prethodnom zadatku raspoznaje jezik $L_1 \cap L_2$ skupom završnih stanja $T = T_1 \times T_2$. □

Zadatak 2.91. *Ako se jezik* L *raspoznatljiv automatom* $\mathcal{A} = (A,a_0,X,\delta,T)$ *onda postoji automat koji raspoznaje komplement jezika* L. *Dokazati.*

Rešenje: Neka je L raspoznatljiv jezik nad konačnim alfabetom X. Ako konačan automat $\mathcal{A} = (A,a_0,X,\delta,T)$ raspoznaje L skupom $T \subseteq A$, tada isti automat raspoznaje komplement jezika L u X^* skupom $A \setminus T$. □

Zadatak 2.92. *Neka su* $\mathcal{A}_1 = (A_1,a_0^1,X,\delta_1,T_1)$ *i* $\mathcal{A}_2 = (A_2,a_0^2,X,\delta_2,T_2)$ *automati koji, raspoznaju, redom, jezike* $L_1,L_2 \subseteq X^*$.

Dokazati da tada postoji automat $\mathcal{A} = (A,a_0,X,\delta,T)$ *koji raspoznaje razliku jezika* $L_1 \setminus L_2$.

Rešenje: Primetimo da, za jezike $L_1,L_2 \subseteq X^*$, važi da je $L_1 \setminus L_2 = L_1 \cap L_2^{\complement}$, gde $L_2^{\complement}$ označava komplement od L_2 u X^*. □

Zadatak 2.93. *Neka su* $\mathcal{A}_1 = (A_1,a_0^1,X,\delta_1,T_1)$ *i* $\mathcal{A}_2 = (A_2,a_0^2,X,\delta_2,T_2)$ *dati automati koji, redom, raspoznaju jezike* $L_1,L_2 \subseteq X^*$.

Konstruisati automat $\mathcal{A} = (A,a_0,X,\delta,T)$ *koji raspoznaje jezik* L_1L_2.

Rešenje: Neka su $L_1,L_2 \subseteq X^*$ dati jezici nad konačnim alfabetom X.

Proizvod jezika $L = L_1L_2$ definiše se na sledeći način:

$$\begin{aligned} L = L_1L_2 &= \{u_1u_2 \mid u_1 \in L_1,\, u_2 \in L_2\} \\ &= \{u \in X^* \mid u = u_1u_2, \text{ za neke } u_1 \in L_1,\, u_2 \in L_2\}. \end{aligned}$$

Razlikovaćemo slučajeve kada je $e \in L$ i kada $e \notin L$.

Slučaj $e \notin L$: Bez umanjenja opštosti možemo pretpostaviti da je $A_1 \cap A_2 = \emptyset$.

Definišimo konačan nedeterministički automat $A = (A, I, X, \delta, T)$ na sledeći način:

Neka je skup stanja $A = A_1 \cup A_2$, $I = \{a_0^1\}$, $T = T_2$ i neka je funkcija prelaza $\delta : A \times X \to \mathcal{P}(A)$ definisana sa

$$\delta(a,x) = \begin{cases} \{\delta_1(a,x)\} & \text{ako je } a \in A_1 \text{ i } \delta_1(a,x) \notin T_1 \\ \{\delta_1(a,x), a_0^2\} & \text{ako je } a \in A_1 \text{ i } \delta_1(a,x) \in T_1 \\ \{\delta_2(a,x)\} & \text{ako je } a \in A_2. \end{cases}$$

Uzmimo proizvoljnu reč $u = u_1u_2$, gde je $u_1 \in L_1$ i $u_2 \in L_2$.

Ako automat A krene sa radom iz stanja a_0^1, onda postoji mogućnost da se automat A posle učitavanja reči u_1 nađe u stanju a_0^2.

Nastavljajući rad u automatu A_2 iz stanja a_0^2, posle učitavanja reči u_2 automat A će se naći u nekom od završnih stanja automata A_2, odnosno u nekom od završnih stanja automata A.

Dakle, postoji put u grafu automata A označen rečju $u = u_1u_2$ koji se završava u završnom stanju automata A (tzv. uspešan put), pa je $u \in L(A)$, što znači da je $L_1L_2 \subseteq L(A)$.

Obratno, neka je $u \in L(A)$, odnosno neka je u grafu automata A rečju u označen neki put koji kreće iz a_0^1, a završava se u nekom stanju $b \in T_2$.

Jedina veza automata A_1 i A_2 je preko stanja a_0^2, pa je $u = u_1u_2$ za reči $u_1, u_2 \in X^*$ takve da postoji put iz a_0^1 u a_0^2 označen sa u_1 i put iz a_0^2 u b označen sa u_2.

Jasno je da je $u_2 \in L_2$. Takođe, iz a_0^1 se sa u_1 može stići u a_0^2 ako i samo ako se iz a_0^1 sa u_1 može stići u neko završno stanje iz T_1, što znači da je $u_1 \in L_1$.

Prema tome, $u = u_1u_2 \in L_1L_2$, čime smo dokazali da je $L(A) \subseteq L_1L_2$.

Dakle, $L(A) = L_1L_2 = L$, što je i trebalo dokazati.

Slučaj $e \in L$: Ovaj slučaj je moguć samo ako je $e \in L_1$ i $e \in L_2$.

Uvedimo oznake

$$L' = L \setminus \{e\}, \quad L_1' = L_1 \setminus \{e\} \quad \text{i} \quad L_2' = L_2 \setminus \{e\}.$$

Lako se proverava da je $L' = L_1'L_2' \cup L_1' \cup L_2'$, odakle sledi da je

$$L = L_1'L_2' \cup L_1' \cup L_2' \cup \{e\}.$$

Jasno je da su L_1' i L_2' raspoznatljivi jezici, pri čemu $e \notin L_1'L_2'$, pa kao što je napred dokazano, $L_1'L_2'$ jeste raspoznatljiv jezik.

Kako je konačna unija raspoznatljivih jezika raspoznatljiv jezik, to sledi da je L raspoznatljiv jezik. Ovim smo dokazali da napred konstuisan automat raspoznaje jezik $L = L_1L_2$. □

Zadatak 2.94. *Dokazati da ako se jezik L može raspoznati konačnim automatom $\mathcal{A} = (A, a_0, X, \delta, T)$, onda postoji konačan automat koji raspoznaje jezik*

$$L^+ = \bigcup_{n\in\mathbb{N}} L^n.$$

Rešenje: Konstruišimo nedeterministički automat $\widehat{\mathcal{A}} = (A, I, X, \widehat{\delta}, T)$ na sledeći način: $I = \{a_0\}$, dok je funkcija prelaza $\widehat{\delta} : A \times X \to \mathcal{P}(A)$ definisana sa

$$\delta(a,x) = \begin{cases} \{\delta_(a,x)\} & \text{ako } \delta(a,x) \notin T \\ \{\delta_(a,x), a_0\} & \text{ako } \delta(a,x) \in T. \end{cases}$$

Jednostavno se pokazuje da automat $\widehat{\mathcal{A}}$ raspoznaje jezik L^+ skupom finalnih stanja T. □

Zadatak 2.95. *Konstruisati automat koji raspoznaje jezik $L = L_1 + L_2$, gde je jezik $L_1 = y^*x^+$ i $L_2 = x^*y^+$.*

Rešenje: Konstruisaćemo minimalne automate jezika L_1 i L_2 nalaženjem desnih razlomaka datih jezika.

$$\begin{aligned} &L_1 = a_0 \\ &L_1.x = \{u \in X^* \mid xu \in L_1\} = x^* = a_1, \\ &L_1.y = \{u \in X^* \mid yu \in L_1\} = y^*x^+ = a_0, \end{aligned}$$

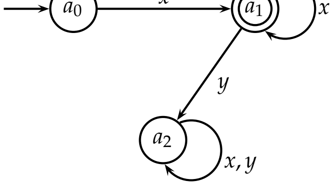
pa je $A_1 = \{a_0, a_1\}$. Dalje, određujemo skup A_2:

$$\begin{aligned} a_1.x &= \{u \in X^* \mid xu \in L_1\} = x^* = a_1 \\ a_1.y &= \{u \in X^* \mid yu \in L_1\} = \emptyset = a_2, \end{aligned}$$

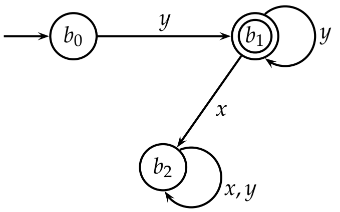
i jasno je da je $A_2 = \{a_0, a_1, a_2\}$. Određujemo istim postupkom skup A_3 i dobijamo

$$a_2.x = a_2.y = \emptyset = a_2,$$

pa je $A_3 = \{a_0, a_1, a_2\}$, što znači da je $A_3 = A_2$. Prema tome, dobili smo da je $A_{L_1} = A_2 = \{a_0, a_1, a_2\}$. Dakle, automat $\mathcal{A}_{L_1} = (A_{L_1}, a_0, X, \delta_1, T_1)$ je zadat grafom:



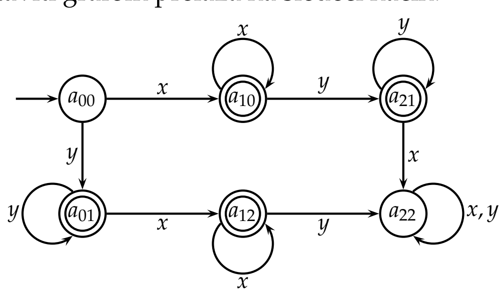
Dualno, dobijamo da je minimalni automat $\mathcal{A}_{L_2} = (A_{L_2}, b_0, X, \delta_2, T_2)$ koji raspoznaje jezik L_2 predstavljen grafom:



Prema Zadatku 2.89. jezik $L_1 + L_2$ je raspznatljiv automatom sa skupom stanja $A = A_{L_1} \times A_{L_2} = \{a_{ij} = (a_i, b_j) | i, j \in \{0,1,2\}\}$, u kome je inicijalno stanje $a_{00} = (a_0, b_0)$ i skup završnih stanja

$$T = \{a_{1i} | i \in \{0,1,2\}\} \cup \{a_{i1} | i \in \{0,1,2\}\}.$$

Po definiciji funkcije prelaza dobijamo da se do stanja a_{02}, a_{11} i a_{20} ne može stici iz inicijalnog stanja, što znači da su ova stanja suvišna. Traženi automat može se predstaviti grafom prelaza na sledeći način:

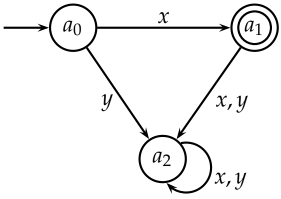


Dakle, dati automat raspoznaje uniju jezika L_1 i L_2. □

Zadatak 2.96. *Konstruisati automat koji prihvata one reči koje počinju sa x i iza svakog x je obavezno bar jedno y.*

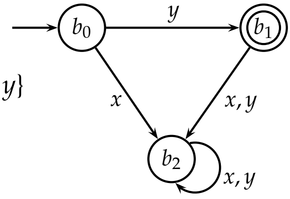
Rešenje: Primetimo da je dati jezik $L = \{xy^+\}^+$. Jedan od načina za konstrukciju automata koji raspoznaje jezik L je konstrukcija automata koji raspoznaju jezike $\{x\}$, $\{y\}$, $\{y\}^+$, proizvod $\{x\}\{y\}^+$, tim redom. Zatim, na osnovu Zadatka 2.94. dobijamo automat traženog jezika. Jednostavno se konstruišu automati koji raspoznaju jezike:

$\{x\}$

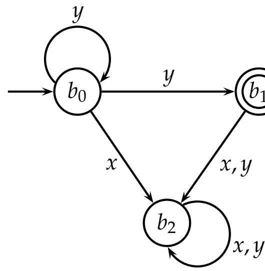


i

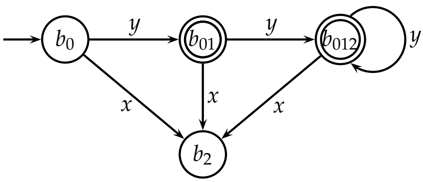
$\{y\}$

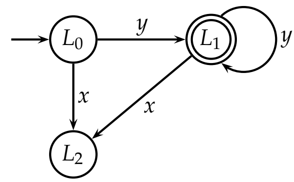


Prema Zadatku 2.94. automat koji raspoznaje jezik $\{y\}^+$ može se predstaviti grafom prelaza:



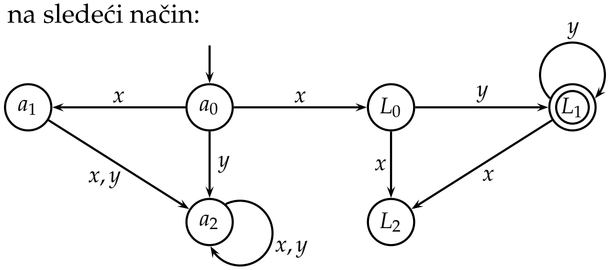
Odgovarajući deterministički automat koji raspoznaje jezik $\{y\}^+$ i njegov minimalni automat mogu se predstaviti na sledeći način:



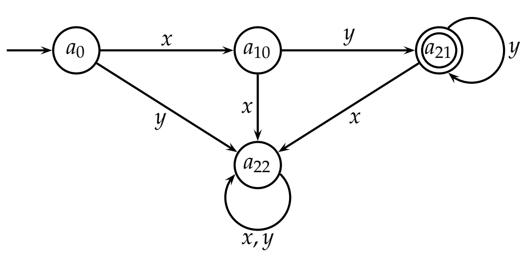


minimalni automat

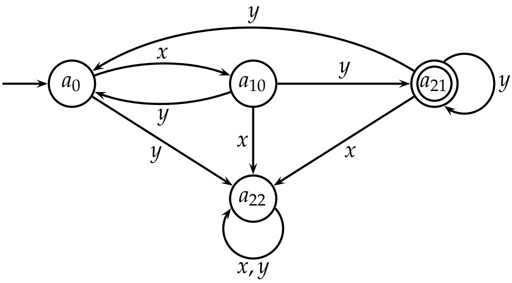
Zatim konstruišemo automat koji raspoznaje proizvod dva raspoznatljiva jezika $\{x\}\{y\}^+$ na sledeći način:



Determinizacijom ovog automata, a zatim minimizacijom dobijenog determinističkog automata dobijamo minimalni automat datog jezika, koji se može predstaviti grafom:



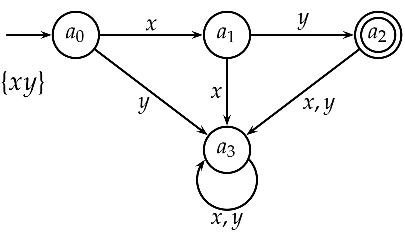
Tako automat predstavljen sledećim grafom raspoznaje jezik $L = \{xy^+\}^+$

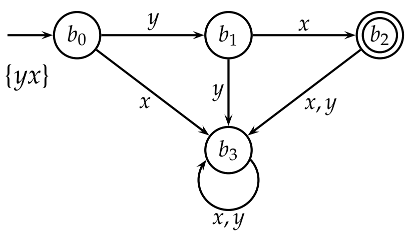


skupom finalnih stanja $T = \{a_{21}\}$. □

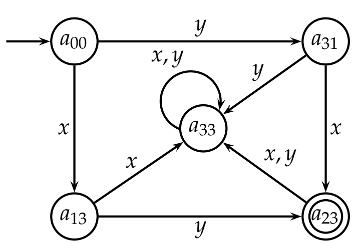
Zadatak 2.97. *Konstruisati automat koji raspoznaje jezik* $L = x^*(xy + yx)$.

Rešenje: U prethodnom zadatku konstruisali smo automate koji raspoznaju jezike $\{x\}$ i $\{y\}$, pa automate koji raspoznaju proizvode ova dva jezika možemo predstaviti na sledeći način:

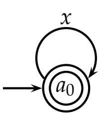




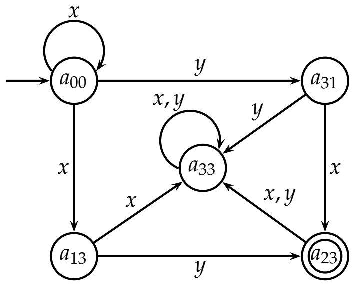
Prema Zadatku 2.89. jezik $\{xy\} + \{yx\}$ je raspznatljiv automatom sa skupom stanja $A = \{a_{ij} = (a_i, b_j) | i, j \in \{0,1,2,3\}\}$, inicijalnim stanjem $a_{00} = (a_0, b_0)$ i skupom završnih stanja $T = \{a_{2i} | i \in \{0,1,2\}\} \cup \{a_{i2} | i \in \{0,1,2\}\}$. Ako posmatramo samo dostižna stanja u ovom automatu i eliminišemo suvišna stanja dobijamo graf prelaza:



Kako je jezik $\{x\}^*$ raspoznatljiv sledećim automatom

 ,

to minimizacijom automata koji rospoznaje proizvod jezik $L = x^*(xy + yx)$ dobijamo automat:



Na ovaj način konstruisali smo minimalni deterministički automat datog jezika L. □

Zadatak 2.98. *Ako je jezik L raspoznatljiv, onda je raspoznatljiv i jezik*

$$MIN(L) = \{x \in L \mid \text{ni jedan pravi prefiks od } x \text{ ne pripada } L\}.$$

Dokazati.

Rešenje: Pretpostavimo da je $\mathcal{A} = (A, a_0, X, \delta, T)$ konačan deterministički automat koji prihvata jezik L skupom finalnih stanja T. Jednostavno se pokazuje da, tada, konačan nedeterministički automat $\mathcal{A}'$ dobijen iz $\mathcal{A}$ brisanjem svih prelaza koji izlaze iz završnih stanja raspoznaje jezik $MIN(L)$. □

Zadatak 2.99. *Ako su L_1 i L_2 raspoznatljivi jezici nad alfabetom $\{0,1\}$, dokazati da je onda raspoznatljiv i jezik*

$$L_1 \vee L_2 = \{u \vee v \mid u \in L_1, v \in L_2, |u| = |v|\}.$$

Napomena 4. *Reči se upoređuju kao nizovi, tj. upoređivanjem simbola na odgovarajućim pozicijama.*

Rešenje: Pretpostavimo da se jezici L_1 i L_2 mogu raspoznati konačnim determinističkim automatima $\mathcal{A}_1 = (A_1, a_0^1, \{0,1\}, \delta_1, T_1)$ i $\mathcal{A}_2 = (A_2, a_0^2, \{0,1\}, \delta_2, T_2)$, tim redom. Konstruisaćemo automat koji raspoznaje jezik $L_1 \vee L_2$. Ako je prelaz u novom automatu 0, onda su prelazi u oba automata označeni sa 0, dok 1 može proizaći iz $0 \vee 1$, $1 \vee 0$ ili $1 \vee 1$. Dakle, treba da omogućimo sve tri kombinacije prelaza u automatima $\mathcal{A}_1$ i $\mathcal{A}_2$. Konstruišimo automat

$$\mathcal{A} = (A_1 \times A_2, (a_0^1, a_0^2), \{0,1\}, \delta, T_1 \times T_2),$$

u kome je funkcija prelaza δ definisana na sledeći način:

$$\delta((a_1, a_2), 0) = \{(\delta_1(a_1, 0), \delta_2(a_2, 0))\}$$
$$\delta((a_1, a_2), 1) = \{(\delta_1(a_1, 0), \delta_2(a_2, 1)), (\delta_1(a_1, 1), \delta_2(a_2, 0)), (\delta_1(a_1, 1), \delta_2(a_2, 1))\}.$$

Jednostavno se pokazuje da je $\mathcal{A}$ nedeterministički automat koji prihvata tačno jezik $L_1 \vee L_2$. □

Zadatak 2.100. *Dokazati da, iz raspoznatljivosti jezika L konačnim automatom, sledi da je raspoznatljiv i jezik* $L_1 = \{uv \mid vu \in L\}$.

Rešenje: Neka je $\mathcal{A} = (A, a_0, X, \delta, T)$ konačan deterministički automat koji raspoznaje jezik L. Bez umanjenja opštosti pretpostavimo da je skup stanja $A = \{a_0, a_1, \ldots, a_n\}$. Posmatrajmo proizvoljnu reč $w \in L$. Ako ovu reč napišemo u obliku $w = vu$, za $v, u \in X^+$, to znači da postoje stanja $a_i, a_j \in A$ takva da je $\delta(a_0, v) = a_i$ i $\delta(a_i, u) = a_j \in T$. Dakle, u novom automatu $\mathcal{A}'$ svakom putu od inicijalnog do nekog završnog stanja pridružujemo niz od $j-1$ puteva. U svakom od njih, redom, međustanje a_i, za $i = \overline{1, j-1}$ proglašavamo i inicijalnim i završnim stanjem, dok stanja a_j i a_0 brišemo i zamenjujemo jednim novim stanjem $\alpha \notin A$. Za reči $v = xv'$ i $u = u'y$, $u', v' \in X^*$ inicijalno i završno stanje biće $a_i = \delta(a_0, v)$, a prelaze ćemo definisati na sledeći način:

$$\begin{aligned} \delta'(a_i, u') &= \delta(a_i, u') = a_{j-1}, & \delta'(a_{j-1}, y) &= \alpha, \\ \delta'(\alpha, x) &= a_1, & \delta'(a_1, v') &= \delta(a_1, v') = a_i. \end{aligned}$$

Jednostavno se pokazuje da ovako definisan automat $\mathcal{A}' = (A, I, X, \delta', I)$, sa istim skupom inicijalnih i završnih stanja

$$I = \bigcup_{a_j \in T} \bigcup_{i=1}^{j-1} a_i,$$

jeste nedeterministički automat koji raspoznaje jezik L_1. □

Zadatak 2.101. *Ako je jezik L raspoznatljiv konačnim automatom, raspoznatljiv je i jezik*

$$L_{\frac{1}{2}} = \{u \in X^* \mid (\exists v \in X^*)\, |u| = |v|,\, uv \in L\}.$$

Dokazati.

Rešenje: Nedeterministički automat koji raspoznaje traženi jezik možemo konstruisati na dva načina.

I *način:* Neka je $\mathcal{A} = (A, X, a_0, \delta, T)$ konačan deterministički automat sa skupom stanja $A = \{a_0, a_1, \ldots a_n\}$, koji raspoznaje jezik L. U nedeterminističkom automatu $\mathcal{A}_1$ koji raspoznaje jezik $L_{\frac{1}{2}}$ paralelno će se obavljati prelaz od a_0 pomoću reči u i prelaz iz $a_j \in T$ pod uticajem reči $\bar{v}$ unatrag. Zato će skup stanja u novom automatu biti $A_1 = A \times A$, skup inicijalnih stanja

$$I = \{(a_0, a_k) \mid a_k \in T\}$$

i skup finalnih stanja $T_1 = \{(a_i, a_i) \mid a_i \in A,\ a_i$ je dostižno stanje u $A_1\}$. Prelaze ćemo, za $x \in X$, definisati na sledeći način:

$$\delta_1((a_i, a_j), x) = \{(a_k, a_l) \mid a_k = \delta(a_i, x),\, \delta(a_l, y) = a_k,\ \text{za neki } y \in X\}.$$

Automat $\mathcal{A}_1 = (A_1, I, X, \delta_1, T_1)$ raspoznaje jezik $L_{\frac{1}{2}}$.

II *način*: Neka je $\mathcal{A} = (A, X, a_0, \delta, T)$ konačan deterministički automat, koji raspoznaje jezik L skupom finalnih stanja T. Bez umanjenja opštosti možemo pretpostaviti da je $A = \{a_0, a_1, \dots a_n\}$. Za stanja $a_i \in A$ definišimo konačne nedeterminističke automate $\mathcal{A}_i = (A, X, a_i, \hat{\delta}, T)$, sa inicijalnim stanjima a_i, $0 \leqslant i \leqslant n$, sa istom funkcijom prelaza

$$\hat{\delta}(a_j, x) = \{\delta(a_j, y) \mid y \in X\}.$$

Sada pravimo kompoziciju automata $\mathcal{A}$ i automata $\mathcal{A}_i$ u nedeterministički automat $\mathcal{A}' = (A', a_0', X, \delta', T')$ na sledeći način:

$$\begin{aligned} A' &= A^{n+2};\\ a_0' &= (a_0, a_0, a_1, \dots, a_n);\\ T' &= \{(a_i, a_{j_0}, a_{j_1}, \dots, a_{j_n}) \mid 0 \leqslant i \leqslant n,\ a_{j_t} \in T\};\\ \delta'\Big((a_i, a_{j_0}, a_{j_1}, \dots, a_{j_n}), x\Big) &= \{(\delta(a_i, x), a_{k_0}, a_{k_1}, \dots a_{k_n}) \mid a_{k_t} \in \hat{\delta}(a_{j_t}, x), \text{ za } t = \overline{0, n}\}. \end{aligned}$$

Ovako konstruisan automat, takođe, raspoznaje jezik $L_{\frac{1}{2}}$, ali ima veći broj stanja nego automat koji je konstruisan, u dokazu, na prvobitan način. Ova tehnika je međutim opštija i ima veću primenu (videti Zadatak 10). □

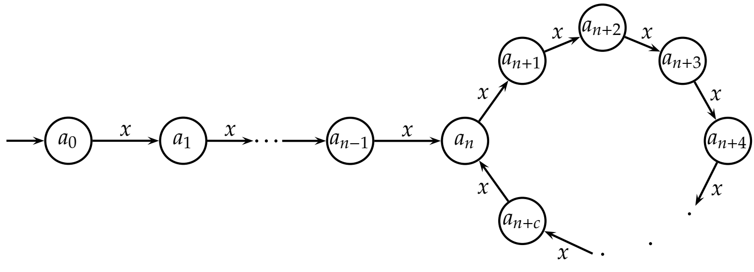
Zadatak 2.102. *Neka su $L_1 \subseteq X^*$ i $L_2 \subseteq Y^*$ dva raspoznatljiva jezika. Dokazati da je jezik*

$$L(L_1, L_2) = \{u \in L_1 \mid (\exists v \in L_2)\ |v| = |u|^2\}$$

takođe raspoznatljiv.

Rešenje: Dokaz ćemo izvesti u četiri koraka.

Najpre posmatrajmo najjednostavniji slučaj kada je $L_1 = \{x\}^*$ i $L_2 \subseteq \{x\}^*$. Pretpostavimo da konačan deterministički automat $\mathcal{A}_2$ raspoznaje jezik L_2. Pošto ulazni alfabet ima samo jedno slovo, to iz svakog stanja izlazi tačno jedna grana. Bez umanjenja opštosti možemo pretpostaviti da je graf prelaza automata $\mathcal{A}_2$ predstavljen na sledeći način:



Dakle, na putu od a_0 do prvog stanja a_n u petlji ima n slova x, dok petlja ima c grana, za $n \geqslant 0$ i $c \geqslant 1$. Pokazaćemo da, za svaki prirodan broj m takav da je $m^2 \geqslant n$ važi da $x^m \in L(\{x\}^*, L_2)$ ako i samo ako $x^{m+c} \in L(\{x\}^*, L_2)$. Primetimo da

$x^m \in L(\{x\}^*, L_2)$ ako i samo ako $x^{m^2} \in L_2$, kao i da $x^{m+k} \in L(\{x\}^*, L_2)$ ako i samo ako $x^{(m+k)^2} \in L_2$. Pošto je $m^2 \geqslant n$, reč x^{m^2} inicijalno stanje automata $\mathcal{A}_2$ vodi u neko stanje a koje se nalazi u petlji. Jasno je da, tada, za proizvoljno $l \geqslant 0$, put x^{m^2+lc} takođe završava u stanju a, tj. $x^{m^2} \in L_2$ ako i samo ako $x^{m^2+lc} \in L_2$, za sve $l \geqslant 0$. Sledi da, ako izaberemo $l = 2m + c$, dobijamo da

$$x^m \in L(\{x\}^*, L_2) \;\Leftrightarrow\; x^{m^2} \in L_2 \;\Leftrightarrow\; x^{m^2+(2m+k)k} \in L_2 \;\Leftrightarrow\; x^{m+k} \in L(\{x\}^*, L_2).$$

Ovo će nam pomoći u konstrukciji autolata $\mathcal{A}_L$ koji prihvata jezik $L(\{x\}^*, L_2)$. Ovaj automat ima graf prelaza kao i automat $\mathcal{A}_2$, sa $n_1 = \lceil \sqrt{n} \rceil$ grana od a_0^L do prvog stanja u petlji a_{n_1} i c grana u petlji. Stanje $a \in T_L$ je finalno stanje ako i samo ako je $x^{j^2} \in L_2$, gde je j dužina najkraćeg puta od a_0^L do a. Dobijeni automat raspoznaje jezik $L(\{x\}^*, L_2)$.

Dalje, posmatramo slučaj kada je $L_1 = X^*$, za neki alfabet X i $L_2 \subseteq \{x\}^*$. Kao i u prvom koraku primećujemo da reč $u \in L(X^*, L_2)$ ako i samo ako $uv \in L(X^*, L_2)$, za sve reči v dužine $|v| = c$. Takođe, $u \in L(X^*, L_2)$ ako i samo ako $w \in L(X^*, L_2)$, za sve reči dužine $|w| = |u|$. Dakle, automat $\mathcal{A}_L^1$ koji raspoznaje jezik $L(X^*, L_2)$ konstruisaćemo kao u prethodnom slučaju, pri čemu svaku granu označenu sa x zamenjujemo skupom grana označenih svim lovima alfabeta X.

U trećem koraku pretpostavimo da je $L_1 = X^*$ i $L_2 \subseteq Y^*$, za neke alfabete X i Y. Jednostavno se pokazuje da, jezik $L_0 = \{x^{|u|} \mid u \in L_2\}$ dobijen iz L_2 zamenom svakog simbola $y \in X$ slovom x, jeste raspoznatljiv i da je $L(X^*, L_2) = L(X^*, L_0)$. To znači da je $L(X^*, L_2)$ raspoznatljiv jezik, jer je, prema prethodnom koraku, raspoznatljiv i $L(X^*, L_0)$.

Dolazimo do najopštijeg slučaja $L_1 \subseteq X^*$ i $L_2 \subseteq Y^*$, za date alfabete X i Y. Jasno da je $L(L_1, L_2) = L_1 \cap L(X^*, L_2)$, te je $L(L_1, L_2)$ raspoznatljiv kao presek dva raspoznatljiva jezika. □

Zadatak 2.103. *Dokazati da iz raspoznatljivosti jezika $L \subseteq X^*$ sledi da je raspoznatljiv i jezik*

$$\mathrm{SQRT}(L) = \{u \in X^* \mid (\exists v \in X^*)\, |v| = |u|^2,\ uv \in L\}.$$

Rešenje: Neka je $\mathcal{A} = (A, a_0, X, \delta, T)$ konačan deterministički automat koji raspoznaje jezik L. Svakom stanju $a \in A$ pridružićemo skup L_1^a svih reči koje inicijalno stanje a_0 vode u a i skup L_2^a svih reči koje stanje a vode u neko završno stanje u automatu $\mathcal{A}$. To znači da je, za $a \in A$,

$$L_1^a = \{u \in X^* \mid \delta(a_0, u) = a\},$$
$$L_2^a = \{u \in X^* \mid \delta(a, u) \in T\}.$$

Vidimo da je

$$\mathrm{SQRT}(L) = \bigcup_{a\in A}\{u \in L_1^a \,|\, (\exists v \in X^*)\, |v| = |u|^2,\ v \in L_2^a\}.$$

Jasno je da su svi jezici L_1^a, L_2^a, za $a \in A$ raspoznatljivi, što, prema prethodnom zadatku znači da su raspoznatljivi i jezici $\{u \in L_1^a \,|\, (\exists v \in X^*)\, |v| = |u|^2,\ v \in L_2^a\}$, za svako stanje $a \in A$. Jezik SQRT(L) je raspoznatljiv kao konačna unija raspozntljivih jezika. □

2.6. Najveća desna kongruencija na monoidu reči

Pored sintaksičke kongruencije na slobodnom monidu X^*, definisaćemo, za jezik $L \subseteq X^*$, još jednu relaciju R_L na sledeći način: za reči $u,v \in X^*$ je

$$(u,v) \in R_L \;\Leftrightarrow\; (\forall w \in X^*)\ (uw \in L \;\Leftrightarrow\; vw \in L).$$

Jednostavno se pokazuje da je R_L relacija ekvivalencije na X^*, kao i da je desno kompatibilna (saglasna), pa je R_L *desna kongruencija* na slobodnom monoidu X^*, koja se naziva *glavnom desnom kongruencijom* jezika L.

Broj klasa ekvivalencije u odnosu na proizvoljnu relaciju ekvivalencije R naziva se *indeks* od R u oznaci $Ind(R)$. Dakle, broj klasa ekvivalencije u odnosu na relaciju R_L, tj. broj elemenata faktor skupa X^*/R_L, je $Ind(R_L)$.

Za relaciju ekvivalencije ϱ na skupu H kažemo da zasićuje podskup $H \subseteq K$ ako se K maže predstaviti u obliku unije nekih ϱ-klasa od H.

Za dati jezik $L \subseteq X^*$, sa ε_L ćemo označiti relaciju ekvivalencije takvu da važi

$$(u,v) \in \varepsilon_L \;\Leftrightarrow\; (u \in L \;\Leftrightarrow\; v \in L).$$

Zadatak 2.104. *Neka je $L \subseteq X^*$. Dokazati da relacija ekvivalencije ϱ na X^* zasićuje L ako i samo ako je $\varrho \subseteq \varepsilon_L$.*

Rešenje: Neka je ϱ relacija ekvivalencije na X^* koja zasićuje L. To znači da se jezik L može napisati u obliku $L = \bigcup\{L_i \,|\, i \in I\}$, gde L_i, $i \in I$ jesu ϱ-klase monoida X^*. Posmatrajmo proizvoljan par $(u,v) \in \varrho$. Ako $u \in L$ to znači da $u \in L_i$ za neki $i \in I$, pa iz $(u,v) \in \varrho$ sledi da $v \in L_i \subseteq L$. Na potpuno isti način pokazujemo da iz uslova $v \in L$ sledi da $u \in L$. Dakle imamo da je $(u,v) \in \varepsilon_L$, tj. $\varrho \subseteq \varepsilon_L$.

Obratno, neka je $\varrho \subseteq \varepsilon_L$. Jasno je da je $L \subseteq \bigcup\{L_i \,|\, i \in I\}$. Da bi dokazali obratnu inkluziju, treba dokazati da je $u\varrho \subseteq L$, za svaku reč $u \in L$. Zaista, neka je $u \in L$ i $v \in u\varrho$. Tada $(u,v) \in \varrho \subseteq \varepsilon_L$, pa $v \in L$, što je i trebalo dokazati. Prema tome pokazali smo da je $L = \bigcup\{L_i \,|\, i \in I\}$, što znači da ϱ zasićuje L. □

Zadatak 2.105. *Za dati jezik L, relacija R_L je najveća desna kongruencija na slobodnom monoidu X^* koja zasićuje L. Dokazati.*

Rešenje: Već smo pomenuli da je R_L desna kongruencija na X^* definisana za dati jezik $L \subseteq X^*$. Za proizvoljne reči $(u,v) \in R_L$ i $w = e$ dobijamo da $u = ue \in L$ ako i samo ako $ve = v \in L$, tj. $(u,v) \in \varepsilon_L$. Kako je $R_L \subseteq \varepsilon_L$, to prema prethodnom zadatku R_L zasićuje L. Dokazaćemo da je, proizvoljna desna kongruencija ϱ na X^*, koja zasićuje L, sadržana u R_L.

Posmatrajmo proizvoljan par $(u,v) \in \varrho$. Zbog desne saglasnosti relacije ϱ, za proizvoljnu reč $w \in X^*$ važi $(uw,vw) \in \varrho \subseteq \varepsilon_L$. Dakle, reč $uw \in L$ ako i samo ako $vw \in L$, za svaki $w \in X^*$, što znači da $(u,v) \in R_L$. Time smo pokazali da je $\varrho \subseteq R_L$, tj. da je R_L najveća desna kongruencija na X^* koja zasićuje L. □

Zadatak 2.106. *Neka je $L \subseteq X^*$ dati jezik i neka je σ proizvoljna desna kongruencija na X^*. Faktor automat $\mathcal{A}_\sigma$ raspoznaje L ako i samo ako σ zasićuje jezik L. Dokazati.*

Rešenje: Pretostavimo da automat $\mathcal{A}_\sigma = (A_\sigma, e\sigma, X, \delta_\sigma, T_\sigma)$ raspoznaje jezik L skupom finalnih stanja T_σ. To znači da je

$$L = \{u \in X^* \,|\, (e\sigma)u \in T_\sigma\} = \{u \in X^* \,|\, u\sigma \in T_\sigma\} = \bigcup_{u\sigma \in T_\sigma} u\sigma,$$

pa prema tome σ zasićije L.

Obratno, neka σ zasićuje L, tj. neka je L unija nekih σ-klasa. Označimo sa T_σ uniju svih σ-klasa sadržanih u L. Tada je

$$L = \{u \in X^* \,|\, u\sigma \in T_\sigma\} = \{u \in X^* \,|\, (e\sigma)u \in T_\sigma\},$$

pa $\mathcal{A}_\sigma$ raspoznaje L skupom završnih stanja T_σ. □

Zadatak 2.107. *Neka je $L \subseteq X^*$ proizvoljan jezik koji je raspoznatljiv konačnim automatom. Dokazati da je faktor automat*

$$\mathcal{A}_{R_L} = (X^*/R_L, eR_L, X, \delta_{R_L}, TR_L)$$

inicijalni automat izomorfan minimalnom automatu $\mathcal{A}_L = (A_L, L, X, \delta_L, T_L)$ desnih razlomaka jezika L.

Rešenje: Definišimo preslikavanje $\varphi : X^*/R_L \mapsto A_L$ sa:

$$\varphi(uR_L) = L.u,$$

za proizvoljnu reč $u \in X^*$. Najpre treba pokazati da je preslikavanje φ dobro definisano. Neka su $u,v \in X^*$ reči takve da je $uR_L = vR_L$. To znači da su $(u,v) \in R_L$, tj. da $uw \in L$ ako i samo ako $vw \in L$, za svaki $w \in X^*$. Dakle, uslov da $w \in L.u$ ekvivalentan je sa $w \in L.v$, čime smo dokazali dobru definisanost preslikavanja φ. Slično se pokazuje da iz jednakosti slika $L.u = L.v$ sledi jednakost klasa $uR_L = vR_L$, za sve $u,v \in X^*$, pa je φ preslikavanje "1-1". Sirjektivnost je očigledna, što znači da je φ bijekcija. Takođe važi $\varphi(eR_L) = L.e = L$ i

$$uR_L \in TR_L \;\Leftrightarrow\; u \in L \;\Leftrightarrow\; ue \in L \;\Leftrightarrow\; e \in L.u \;\Leftrightarrow\; L.u \in T_L \;\Leftrightarrow\; \varphi(uR_L) \in T_L,$$

pa se inicijalno stanje preslikavanjem φ slika i inicijalno stanje automata $\mathcal{A}_L$ i svako završno stanje automata $\mathcal{A}_{R_L}$ slika se u završno stanje automata desnih razlomaka. Po definiciji faktor automata, za reči $u, v \in X^*$ važi:

$$\begin{aligned}\varphi(\delta_{R_L}(uR_L, v)) = \varphi((uR_L)v) = \varphi((uv)R_L) = L.uv \\ = (L.u).v = \delta_L(L.u, v) = \delta_L(\varphi(uR_L), v).\end{aligned}$$

Jasno da je φ izomorfizam automata $\mathcal{A}_{R_L}$ na $\mathcal{A}_L$, što je i trebalo dokazati. □

Zadatak 2.108. *Jezik $L \subseteq X^*$ je raspoznatljiv konačnim automatom ako i samo ako je glavna desna kongruencija R_L konačnog indeksa. Dokazati.*

Rešenje: Kako glavna desna kongruencija R_L zasićuje L, faktor automat $\mathcal{A}_{R_L}$ raspoznaje L. Ako je $Ind(R_L)$ konačan, skup X^*/R_L je konačan, što znači da je $\mathcal{A}_{R_L}$ konačan automat koji raspoznaje jezik L.

Obratno, ako je $\mathcal{A}$ konačan automat sa n stanja koji raspoznaje L, prema prethodnom zadatku je

$$Ind(R_L) = |X^*/R_L| \leqslant n,$$

što je i trebalo dokazati. □

Zadatak 2.109. *Neka je $L = \{u \in \{x, y\}^* \,|\, |u|_x = |u|_y\}$. Da li L može biti raspoznat konačnim automatom?*

Rešenje: Proverićemo da li je glavna desna kongruencija jezika L konačnog indeksa. Za reči $u, v \in X^*$ važi

$$\begin{aligned}(u,v) \in R_L \;&\Leftrightarrow\; (\forall p \in X^*)\;\; |up|_x = |up|_y \;\Leftrightarrow\; |vp|_x = |vp|_y \\ &\Leftrightarrow\; (\forall p \in X^*)\;\; |u|_x + |p|_x = |u|_y + |p|_y \;\Leftrightarrow\; |v|_x + |p|_x = |v|_y + |p|_y \\ &\Leftrightarrow\; (\forall p \in X^*)\;\; |u|_x - |u|_y = |p|_y - |p|_x \;\Leftrightarrow\; |v|_x - |v|_y = |p|_y - |p|_x \\ &\Leftrightarrow\; |u|_x - |u|_y = |v|_x - |v|_y\,.\end{aligned}$$

Definišimo preslikavanje $\phi : X^*/R_L \mapsto \mathbb{Z}$ na sledeći način:

$$\phi(uR_L) = |u|_x - |u|_y,$$

za proizvoljnu reč $u \in X^*$. Injektivnost i dobra definisanost preslikavanja ϕ jednostavno se pokazuju. Naime, važi

$$(u,v) \in R_L \;\Leftrightarrow\; uR_L = vR_L \;\Leftrightarrow\; |u|_x - |u|_y = |v|_x - |v|_y \;\Leftrightarrow\; \phi(uR_L) = \phi(vR_L).$$

Ako je $z \in \mathbb{Z}$ ceo broj i $z > 0$ možemo naći reč u takvu da je $|u|_x - |u|_y = z$. Za $z < 0$ biramo reč u u kojoj je broj pojavljivanja slova y veći od broja pojavljivanja slova x, tj. $|u|_y - |u|_x = -z$. Jasno je da je ϕ bijekcija, pa je $|X^*/R_L| = |\mathbb{Z}| = \aleph_0$.

Kako glavna desna kongruencija R_L nije konačnog indeksa to, prema prethodnom zadatku, zaključujemo da L nije raspoznatljiv konačnom automatom. □

Zadatak 2.110. *Ispitati raspoznatljivost jezika* $L = \{u \in \{x,y\}^* \,|\, |u|_x \equiv 1 \pmod 3\}$.

Rešenje: Definišimo glavnu desnu kongruenciju jezika L. Za reči $u, v \in X^*$ važi

$$\begin{aligned}(u,v) \in R_L &\Leftrightarrow (\forall p \in X^*)\ |up|_x \equiv 1\ (\text{mod}3) \Leftrightarrow |vp|_x \equiv 1\ (\text{mod}3)\\ &\Leftrightarrow (\forall p \in X^*)\ |u|_x + |p|_x \equiv 1\ (\text{mod}3) \Leftrightarrow |v|_x + |p|_x \equiv 1\ (\text{mod}3)\\ &\Leftrightarrow (\forall p \in X^*)\ |u|_x + |p|_x \equiv |v|_x + |p|_x \equiv 1\ (\text{mod}3)\\ &\Leftrightarrow |u|_x - |v|_x \equiv |p|_x - |p|_x \equiv 0\ (\text{mod}3)\\ &\Leftrightarrow |u|_x \equiv |v|_x\ (\text{mod}3).\end{aligned}$$

Definišimo preslikavanje $\phi : X^*/R_L \mapsto \mathbb{Z}_3$ ($\mathbb{Z}_3$ je sistem ostataka po modulu 3, odnosno $\mathbb{Z}_3 = \{0,1,2\}$) na sledeći način:

$$\phi(uR_L) = a \equiv |u|_x\ (\text{mod}3),\ a \in \mathbb{Z}_3$$

za $uR_L \in X^*/R_L$. Injektivnost i dobra definisanost preslikavanja ϕ jednostavno se pokazuju. Naime, važi

$$(u,v) \in R_L \ \Leftrightarrow\ uR_L = vR_L \ \Leftrightarrow\ |u|_x \equiv |v|_x \equiv a\ (\text{mod}3) \ \Leftrightarrow\ \phi(uR_L) = \phi(vR_L).$$

Ako je $a \in \mathbb{Z}_3$ možemo naći reč u takvu da je $|u|_x \equiv a$ (mod3). Jasno je da je ϕ bijekcija, pa je $|X^*/R_L| = |\mathbb{Z}_3| = 3$.

Kako je glavna desna kongruencija R_L konačnog indeksa, jezik L je raspoznatljiv konačnom automatom (sa tri stanja). □

Zadatak 2.111. *Dokazati da jezik* $L = \{u\overline{u} \mid u \in \{x,y\}^*\}$ *nije raspoznatljiv.*

Rešenje: Očito je da uvek možemo odabrati reči $u, v \in \{x,y\}^*$, takve da je $u \neq v$, takve da $v\overline{u} \notin L$. Kako važi $u\overline{u} \in L$, to $(u,v) \notin R_L$. Prema tome, uvek se mogu naći dve različite reči iz $\{x,y\}^*$ koje pripadaju različitim klasama ekvivalencije R_L. To znači da je $Ind(R_L) = \infty$, tj. jezik L nije raspoznatljiv. □

Zadatak 2.112. *Dokazati da jezik* $L = \{0^m 1^n \,|\, (m,n) = 1\}$ *nije raspoznatljiv.*

Rešenje: Za svaka dva različita prosta boja p i q važi da $0^q 1^p \in L$ i $0^p 1^p \notin L$. Dakle, za proizvoljno izabrane proste brojeve p i q, reči 0^p i 0^q ne pripadaju istoj klasi ekvivalencije R_L.

Kako ima beskonačno mnogo prostih brojeva, zaključujemo da je $Ind(R_L) = \infty$, pa jezik L nije raspoznatljiv. □

2.7. Raspozntljivi jezici. Lema o napumpavanju

Na osnovu ranije dokazanog, jednostavno se dokazuje sledeća teorema koja daje karakterizaciju raspoznatljivih jezika:

Teorema 2.13. *Neka je $L \subseteq X^*$ jezik nad konačnim alfabetom X. Tada su sledeći uslovi ekvivalentni:*

(i) *L je raspoznatljiv;*
(ii) *L je zasićen nekom desnom kongruencijom na X^* konačnog indeksa;*
(iii) *glavna desna kongruencija R_L na X^* je konačnog indeksa;*
(iv) *L može biti raspoznat konačnim monoidom;*
(v) *L je zasićen nekom kongruencijom na X^* konačnog indeksa;*
(vi) *sintaksička kongruencija P_L na X^* je konačnog indeksa.*

Naredna teorema daje važno svojstvo raspoznatljivih jezika.

Teorema 2.14. (Lema o napumpavanju) *Neka je $L \subseteq X^*$ beskonačan, raspoznatljiv jezik. Tada postoji $n \in \mathbb{N}$ takav da svaka reč $u \in L$ dužine $|u| \geqslant n$ može biti zapisana u obliku $u = v_1wv_2$, pri čemu važi:*

(i) $v_1, v_2 \in X^*$ *i* $w \in X^+$;
(ii) $|v_1w| \leqslant n$;
(iii) $v_1w^mv_2 \in L$, *za svaki* $m \in \mathbb{N}^0$.

Zadatak 2.113. *Dokazati da jezik $L = \{x^{k^2} \,|\, k \in \mathbb{N}\}$ nad alfabetom $X = \{x\}$ nije raspoznatljiv.*

Rešenje: Pretpostavimo suprotno da je jezik L raspoznatljiv. Kako je L beskonačan jezik, to postoji broj $n \in \mathbb{N}$, takav da se svaka reč $u \in L$ dužine $|u| \geqslant n$ može napisati u obliku $u = v_1wv_2$ i zadovoljeni su uslovi Leme o napumpavanju.

Neka je $u = x^{n^2}$. Dužina reči u je $|u| = n^2 > n$, pa je $u = v_1wv_2$, za neke $v_1, v_2 \in X^*$ i $w \in X^+$. Prema uslovu (ii) Leme o napumpavanju važi $|v_1w| \leqslant n$, odakle je $1 \leqslant |w| \leqslant n$, dok prema uslovu (iii) važi da reč $v_1w^2v_2 \in L$. Pored toga zadovoljena je nejednakost:

$$n^2 < |v_1w^2v_2| \leqslant n^2 + n < (n+1)^2.$$

Ovo znači da $v_1wv_2 \notin L$, čime smo došli do kontradikcije. Dakle, jezik L nije rasploznatljiv. □

Zadatak 2.114. *Dokazati da jezik $L = \{a^p \,|\, p$ je prost broj$\}$ nije raspoznatljiv.*

Rešenje: Pođimo od obratne pretpostavke da je jezik L raspoznatljiv. Tada važe uslovi Leme o napumpavanju, pa proizvoljnu reč oblika a^p, pri čemu je prost broj $p > n$, možemo napisati u obliku $a^p = v_1wv_2$, za $v_1, v_2 \in X^*$ i $w \in X^+$. Ako pretpostavimo da je dužina $|w| = k$, tada važi $k \leqslant n < p$. Jasno je da imamo

$$|v_1w^{p+1}v_2| = |v_1wv_2| + |w^p| = p + kp = p(k+1).$$

Dakle, dužina $|v_1w^{p+1}v_2|$ nije prost broj, pa $v_1w^{p+1}v_2 \notin L$, što znači da nije zadovoljen uslov (iii) Leme o napumpavanju, tj. jezik $L = \{a^p \,|\, p - \text{prost broj}\}$ nije raspoznatljiv. □

Zadatak 2.115. *Ispitati raspoznatljivost jezika* $L = \{x^n y^n \mid n \in \mathbb{N}\}$.

Rešenje: Pretpostavimo da je L raspoznatljiv jezik. Prema Lemi o napumpavanju, reč $u = x^n y^n$ dužine $|u| = 2n > n$ možemo napisati u obliku $u = v_1 w v_2$, za $v_1, v_2 \in X^*$ i $w \in X^+$. Kako je $|v_1 w| \leqslant n$, to reč w sadrži bar jedno x i ni jedno y, dok v_2 sadrži sve y-ne i, možda neko x. Pretpostavimo, bez umanjenja opštosti, da je $w = x^k$, $k < n$. Tada je

$$|v_1 w^0 v_2| = |v_1 v_2| = n - k < n,$$

pa reč $v_1 w^0 v_2 = v_1 v_2 = x^k y^n \notin L$. Dakle, dati jezik L nije raspoznatljiv. □

Zadatak 2.116. *Da li je raspoznatljiv jezik* $L = \{u \in X^* \mid |u|_x < |u|_y\}$?

Rešenje: Pretpostavimo da je L raspoznatljiv jezik. Prema Lemi o napumpavanju, bez umanjenja opštosti, možemo pretpostaviti da je reč u oblika $u = x^n y^{n+1}$, za prirodan broj $n \in \mathbb{N}$. Očito je da $u \in L$, jer je $|u|_x = n < |u|_y = n+1$ i dužina $|u| = 2n+1 > n$, pa postoje $v_1, v_2 \in X^*$ i $w \in X^+$ takve da je $u = v_1 w v_2$. Kako je, prema drugom uslovu, $|v_1 w| \leqslant n$, to reč w sadrži bar jedno x i ni jedno y, dok v_2 sadrži sve y i, možda, neko x.

Prema tome, imamo da je $w = x^k$, $n \geqslant k \geqslant 1$, te važi

$$|v_1 w^2 v_2|_x = |v_1 w v_2|_x + |w|_x = n + k \geqslant n + 1 = |v_1 w v_2|_y = |v_1 w^2 v_2|_y,$$

pa $v_1 w^2 v_2 \notin L$. Dakle, dati jezik L nije raspoznatljiv. □

Zadatak 2.117. *Da li je jezik* $L = \{0^{n!} 1^{n!} \mid n \geqslant 1\}$ *nad binarnim alfabetom raspoznatljiv?*

Rešenje: Pođimo od pretpostavke da je L raspoznatljiv jezik. Ako je uvedemo oznaku $k = n!$, reč $u = 0^k 1^k$ dužine $|u| = 2k > k$ možemo napisati u obliku $u = v_1 w v_2$, za $v_1, v_2 \in X^*$ i $w \in X^+$. Kako je $|v_1 w| \leqslant k$, to reč w sadrži bar jednu nulu i ni jednu jedinicu. Bez umanjenja opštosti možemo pretpostaviti da je $w = 0^l$, $k \geqslant l \geqslant 1$. Tada je

$$v_1 w^2 v_2 = 0^{k+l} 1^k = 0^l 0^k 1^k = 0^{l+n!} 1^{n!}, \text{ za } 1 \leqslant l \leqslant n!$$

pa $v_1 w^2 v_2 \notin L$. Prema tome, dati jezik L nije raspoznatljiv. □

Zadatak 2.118. *Dokazati da je raspoznatljiv jezik*

$$L = \{u \in \{0,1\}^* \mid u \text{ je binarni zapis broja } n = 2^m(2^k - 1) + 1, \text{ za } m + k - \text{parno}\}.$$

Rešenje: Primetimo da se binarni zapis broja $n = 2^{m+k} - 2^m + 1$ može predstaviti u obliku $u = 1^k 0^{m-1} 1$, pri čemu je $m + k$-paran broj. Ako definišemo jezike $L_{ip} = \{i^l \mid l \text{ je paran broj}\}$ i $L_{in} = \{i^l \mid l \text{ je neparan broj}\}$, za $i \in \{0,1\}$ važi da je

$$L = L_{1p} L_{0n} 1 \cup L_{1n} L_{0p} 1.$$

Jednostavno se pokazuje da je svaki od definisanih jezika $L_{0p}, L_{0n}, L_{1p}, L_{1n}$ raspoznatljiv, što znači da su raspoznatljivi i jezici $L_{1p}L_{0n}1$ i $L_{1n}L_{0p}1$ (proizvodi raspoznatljivih jezika), pa je jezik L raspoznatljiv, kao unija raspoznatljivih jezika. □

Zadatak 2.119. *Ispitati raspoznatljivost jezika*

$$L = \{u \in \{0,1\}^* \,|\, u \text{ je binarni zapis broja } (2^{k+1}-1)^2, \text{ za prirodan broj } k\}.$$

Rešenje: Za $k \in \mathbb{N}$ važi $(2^{k+1}-1)^2 = 2^{2k+2} - 2\cdot 2^{k+1} + 1 = 2^{k+2}(2^k - 1) + 1$, što znači da je traženi jezik L sadržan u jeziku iz prethodnog zadatka. Binarni zapis broja ovog oblika je $1^{k-1}0^k1$, pa je jasno da je $1L = L'1$, gde je $L' = \{1^n0^n \,|\, n \in \mathbb{N}\}$. Prema Zadatku 2.115. jezik L' nije raspoznatljiv, te nije raspoznatljiv ni L. □

2.8. Regularni izrazi. Regularni jezici. Teorema Kleenea

Neka je $\mathcal{A} = (A, a_1, X, \delta, T)$ konačan nedeterministički automat. Bez umanjenja opštosti možemo pisati $A = \{a_1, a_2, \dots, a_n\}$, za neko $n \in \mathbb{N}$.

Za proizvoljne $i, j \in \{1, 2, \dots, n\}$, označimo sa $L_{ij}^{(0)}$ skup svih reči iz X^* koje stanje a_i automata $\mathcal{A}$ prevode direktno u stanje a_j. Jasno, proizvoljan element iz $L_{ij}^{(0)}$ je ili prazna reč ili neko slovo. Dalje, za proizvoljne $i, j, k \in \{1, 2, \dots, n\}$, označimo sa $L_{ij}^{(k)}$ skup svih reči iz X^* koje stanje a_i prevode u a_j ili direkno, ili preko niza međustanja koja pripadaju skupu $\{a_1, a_2, \dots, a_k\}$.

Teorema 2.15. *Neka je $\mathcal{A} = (A, a_1, X, \delta, T)$ konačan automat sa n stanja. Tada, za proizvoljne $i, j, k \in \{1, 2, \dots, n\}$ važi sledeća rekurentna formula:*

$$L_{ij}^{(k)} = L_{ij}^{(k-1)} \cup L_{ik}^{(k-1)} \left(L_{kk}^{(k-1)}\right)^{(*)} L_{kj}^{(k-1)}. \tag{2.3}$$

Ova teorema omogućava da dokažemo da je jezik svakog konačnog automata regularan i, ujedno, daje algoritam pomoću koga dobijamo regularan izraz koji predstavlja jezik datog automata.

Setimo se da je jedan od načina za predstavljanje regularnih jezika njihovo predstavljanje pomoću *regularnih izraza*. Videli smo da se elementarni regularni jezici mogu predstaviti primitivnim regularnim izrazima. Ako su r_{L_1} i r_{L_2} regularni izrazi koji predstavljaju jezike L_1 i L_2, redom, onda su $(r_{L_1}) \cup (r_{L_2})$, $(r_{L_1})(r_{L_2})$ i $(r_{L_1})^*$ regularni izrazi koji predstavljaju $L_1 \cup L_2$, L_1L_2 i L_1^*.

Takođe, svakom regularnom izrazu r nad alfabetom X, možemo pridružiti regularan jezik $L(r)$ koji je *interpretacija regularnog izraza r*.

Jasno je da su jezici $L_{ij}^{(0)}$, iz rekurentne formule (2.3), regularni kao elementarni jezici, pa su i svi jezici

$$L_{ij}^{(l)}, \text{ za } i,j,l \in \{1,2,\ldots,n\},$$

regularni, jer su dobijeni konačnom primenom operacija unije, proizvoda i zvezda-operacije. Prema tome, svaki od ovih jezika je interpretacija nekog regularnog izraza, tj.

$$L_{ij}^{(l)} = L\left(r_{ij}^{(l)}\right), \text{ za } i,j,l \in \{1,2,\ldots,n\}.$$

Jednostavno se pokazuje da je $L_{ij}^{(k)} = L\left(r_{ij}^{(k)}\right) = L\left(r_{ij}^{(k-1)} + r_{ik}^{(k-1)}\left(r_{kk}^{(k-1)}\right)^* r_{kj}^{(k-1)}\right)$, za $i,j,k \in \{1,2,\ldots,n\}$.

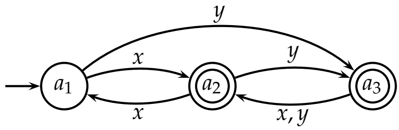
Prema tome, ako je $\mathcal{A} = (A, a_1, X, \delta, T)$ konačan automat sa skupom stanja $A = \{a_1, a_2, \ldots, a_n\}$, svi jezici $L_{ij}^{(l)}$, za $i,j,l \in \{1,2,\ldots,n\}$ su regularni. Ako je skup završnih stanja $T = \{a_{j_1}, a_{j_2}, \ldots, a_{j_m}\}$, onda je jezik automata $\mathcal{A}$ regularan, jer važi $L(A) = L_{1j_1}^{(n)} \cup L_{1j_2}^{(n)} \cup \ldots \cup L_{1j_m}^{(n)}$.

Za regulararni jezik L, indukcijom po složenosti izraza r koji ga predstavlja, dokazujemo da postoji konačan automat koji raspoznaje jezik $L = L(r)$.

Naredna teorema pokazuje da postoji ekvivalencija između pojmova *regularan jezik* i *jezik raspoznatljiv konačnim automatom*.

Teorema 2.16. (Teorema Kleenea) *Jezik $L \subseteq X^*$ nad konačnim alfabetom X je regularan ako i samo ako postoji konačan automat koji ga raspoznaje.*

Zadatak 2.120. *Naći jezik automata sa slike:*



Rešenje: Jezik koji se raspoznaje ovim automatom je $L = L_{12}^{(3)} \cup L_{13}^{(3)}$. Primenom rekurentne formule dobijamo:

$$L_{12}^{(3)} = L_{12}^{(2)} \cup L_{13}^{(2)}\left(L_{33}^{(2)}\right)^* L_{32}^{(2)};$$
$$L_{12}^{(2)} = L_{12}^{(1)} \cup L_{12}^{(1)}\left(L_{22}^{(1)}\right)^* L_{22}^{(1)};$$
$$L_{12}^{(1)} = L_{12}^{(0)} \cup L_{11}^{(0)}\left(L_{11}^{(0)}\right)^* L_{12}^{(0)} = x + ee^*x = x + x = x;$$
$$L_{12}^{(1)} = x;$$
$$L_{22}^{(1)} = L_{22}^{(0)} \cup L_{21}^{(0)}\left(L_{11}^{(0)}\right)^* L_{12}^{(0)} = e + xe^*x = e + xx;$$
$$L_{22}^{(1)} = e + xx;$$
$$L_{12}^{(2)} = x + x(e+xx)^*(e+xx) = x + x(e+xx)^+ = x(e+xx)^+ = x(xx)^*;$$
$$L_{12}^{(2)} = x(xx)^*;$$

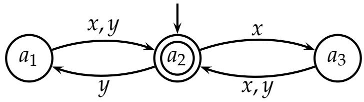
$$
\begin{aligned}
L_{13}^{(1)} &= L_{13}^{(0)} \cup L_{11}^{(0)}\left(L_{11}^{(0)}\right)^* L_{13}^{(0)} = y + ee^*y = y;\\
L_{13}^{(1)} &= y;\\
L_{13}^{(2)} &= L_{13}^{(1)} \cup L_{12}^{(1)}\left(L_{22}^{(1)}\right)^* L_{23}^{(1)};\\
L_{23}^{(1)} &= L_{23}^{(0)} \cup L_{21}^{(0)}\left(L_{11}^{(0)}\right)^* L_{13}^{(0)} = y + xe^*y = y + xy = (e+x)y;\\
L_{23}^{(1)} &= (e+x)y;\\
L_{13}^{(2)} &= y + x(e+xx)^*(e+x)y = (e + (x + x(xx)^*)(e+x))y\\
L_{13}^{(2)} &= (e + x(xx)^*(e+x))y = (e + x(xx)^* + (xx)^+)y\\
L_{13}^{(2)} &= (e + x + xx)(xx)^*y = (e+x)(xx)^*y;\\
L_{33}^{(2)} &= L_{33}^{(1)} \cup L_{32}^{(1)}\left(L_{22}^{(1)}\right)^* L_{23}^{(1)};\\
L_{32}^{(1)} &= L_{32}^{(0)} \cup L_{31}^{(0)}\left(L_{11}^{(0)}\right)^* L_{12}^{(0)} = x + y + \emptyset(e)^*x = x + y + \emptyset = x + y;\\
L_{32}^{(1)} &= x + y;\\
L_{33}^{(1)} &= L_{33}^{(0)} \cup L_{31}^{(0)}\left(L_{11}^{(0)}\right)^* L_{13}^{(0)} = e + \emptyset(e)^*y = e + \emptyset = e;\\
L_{33}^{(1)} &= e;\\
L_{33}^{(2)} &= e + (x+y)(e+xx)^*(e+x)y = e + (x+y)(xx)^*(e+x)y\\
L_{33}^{(2)} &= e + (x+y)(xx)^*(y+xy);\\
L_{32}^{(2)} &= L_{32}^{(1)} \cup L_{32}^{(1)}\left(L_{22}^{(1)}\right)^* L_{22}^{(1)} = x + y + (x+y)(e+xx)^*(e+xx)\\
L_{32}^{(2)} &= x + y + (x+y)(xx)^*(e+xx) = x + y + (x+y)((xx)^* + (xx)^+)\\
L_{32}^{(2)} &= x + y + (x+y)(xx)^* = (x+y)(xx)^*;\\
L_{12}^{(3)} &= L_{12}^{(2)} \cup L_{13}^{(2)}\left(L_{33}^{(2)}\right)^* L_{32}^{(2)}\\
L_{12}^{(3)} &= x(xx)^* + (e+x)(xx)^*y\Big(e + (x+y)(xx)^*(y+xy)\Big)^*(x+y)(xx)^*\\
L_{12}^{(3)} &= x(xx)^* + (e+x)(xx)^*y\Big((x+y)(xx)^*(y+xy)\Big)^*(x+y)(xx)^*\\
L_{13}^{(3)} &= L_{13}^{(2)} \cup L_{13}^{(2)}\left(L_{33}^{(2)}\right)^* L_{33}^{(2)} = L_{13}^{(2)} \cup L_{13}^{(2)}\left(L_{33}^{(2)}\right)^+\\
L_{13}^{(3)} &= (e+x)(xx)^*y + (e+x)(xx)^*y\Big(e + (x+y)(xx)^*(y+xy)\Big)^+\\
L_{13}^{(3)} &= (e+x)(xx)^*y\Big((x+y)(xx)^*(y+xy)\Big)^*;\\
L &= L_{12}^{(3)} \cup L_{13}^{(3)}
\end{aligned}
$$

Dakle, našli smo jezik

$$L = x(xx)^* + (e+x)(xx)^*y\Big((x+y)(xx)^*(y+xy)\Big)^*(e + (x+y)(xx)^*),$$

koji se raspoznaje datim automatom. □

Zadatak 2.121. *Odrediti jezik koji se raspoznaje automatom sa slike:*



Rešenje: Kako je stanje a_2 istovremeno i inicijalno i završno stanje, dati automat raspoznaje jezik $L = L_{22}^{(3)}$. Primenimo rekurentnu formulu:

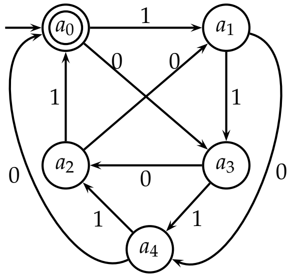
$$L_{22}^{(3)} = L_{22}^{(2)} \cup L_{23}^{(2)}\left(L_{33}^{(2)}\right)^* L_{32}^{(2)};$$
$$L_{22}^{(2)} = L_{22}^{(1)} \cup L_{22}^{(1)}\left(L_{22}^{(1)}\right)^* L_{22}^{(1)} = L_{22}^{(1)} \cup L_{22}^{(1)}\left(L_{22}^{(1)}\right)^+ = \left(L_{22}^{(1)}\right)^+;$$
$$L_{22}^{(1)} = L_{22}^{(0)} \cup L_{21}^{(0)}\left(L_{11}^{(0)}\right)^* L_{12}^{(0)} = e + ye^*(x+y) = e + y(x+y);$$
$$L_{22}^{(1)} = e + y(x+y);$$
$$L_{22}^{(2)} = \left(L_{22}^{(1)}\right)^+ = \left(e + y(x+y)\right)^+ = \left(y(x+y)\right)^*;$$
$$L_{23}^{(2)} = L_{23}^{(1)} \cup L_{22}^{(1)}\left(L_{22}^{(1)}\right)^* L_{23}^{(1)} = L_{23}^{(1)} \cup \left(L_{22}^{(1)}\right)^+ L_{23}^{(1)};$$
$$L_{23}^{(1)} = L_{23}^{(0)} \cup L_{21}^{(0)}\left(L_{11}^{(0)}\right)^* L_{13}^{(0)} = x + ye^*\emptyset = x;$$
$$L_{23}^{(1)} = x;$$
$$L_{23}^{(2)} = x + \left(e + y(x+y)\right)^+ x = x + \left(y(x+y)\right)^* x = \left(y(x+y)\right)^* x;$$
$$L_{33}^{(2)} = L_{33}^{(1)} \cup L_{32}^{(1)}\left(L_{22}^{(1)}\right)^* L_{23}^{(1)};$$
$$L_{33}^{(1)} = L_{33}^{(0)} \cup L_{31}^{(0)}\left(L_{11}^{(0)}\right)^* L_{13}^{(0)} = e + \emptyset(e)^*\emptyset = e + \emptyset = e;$$
$$L_{33}^{(1)} = e;$$
$$L_{32}^{(1)} = L_{32}^{(0)} \cup L_{31}^{(0)}\left(L_{11}^{(0)}\right)^* L_{12}^{(0)} = x + y + \emptyset(e)^*(x+y) = x + y + \emptyset = x + y;$$
$$L_{32}^{(1)} = x + y;$$
$$L_{33}^{(2)} = e + (x+y)\left(e + y(x+y)\right)^* x = e + (x+y)\left(y(x+y)\right)^* x;$$
$$L_{33}^{(2)} = e + (x+y)\left(y(x+y)\right)^* x;$$
$$L_{32}^{(2)} = L_{32}^{(1)} \cup L_{32}^{(1)}\left(L_{22}^{(1)}\right)^* L_{22}^{(1)} = x + y + (x+y)\left(e + y(x+y)\right)^+$$
$$L_{32}^{(2)} = x + y + (x+y)\left(y(x+y)\right)^* = (x+y)\left(y(x+y)\right)^*;$$
$$L_{22}^{(3)} = L_{22}^{(2)} \cup L_{23}^{(2)}\left(L_{33}^{(2)}\right)^* L_{32}^{(2)}$$
$$L_{22}^{(3)} = \left(y(x+y)\right)^* + \left(y(x+y)\right)^* x\left(e + (x+y)\left(y(x+y)\right)^* x\right)^* (x+y)\left(y(x+y)\right)^*$$
$$L_{22}^{(3)} = \left(y(x+y)\right)^* + \left(y(x+y)\right)^* x\left((x+y)\left(y(x+y)\right)^* x\right)^* (x+y)\left(y(x+y)\right)^*.$$

Dakle, dati automat raspoznaje jezik $L = L_{22}^{(3)}$. □

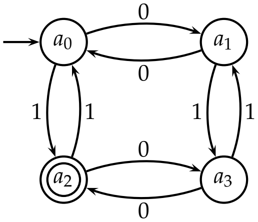
Zadaci za samostalni rad

1. Naći jezik svakog od determinističkih automata koji su predstavljeni, redom, grafovima prelaza (a), (b), (c):

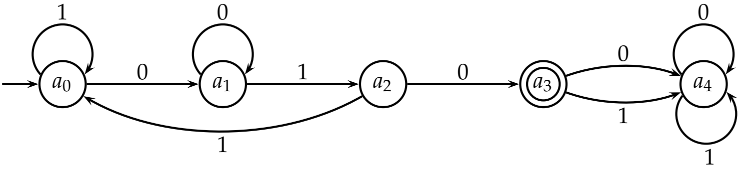
(a)



(b)



(c)



2. Konstruisati automat $\mathcal{A}$ koji prihvata sve reči nad $\{0,1\}$ koje počinju sa 11, a posle imaju proizvoljan broj nula.
3. Konstruisati automat $\mathcal{A}$ čiji je jezik

$$L(A) = \{u \in \{0,1\}^* \mid (\exists v_1, v_2 \in \{0,1\}^*)(\exists n, m \in \mathbb{N})(u = v_1 v_2 \land v_1 = 0^n \land v_2 = 1^m)\}.$$

4. Konstruisati automat koji prihvata tačno one reči koje se završavaju sa 00.
5. Konstruisati automat koji prihvata tačno reči nad binarnim alfabetom koje negde imaju tačno tri uzastopne nule.
6. Konstruisati deterministički automat $\mathcal{A}$ koji prihvata jezik:

   (a) $L(A) = \{w \in \{x,y\}^* \mid$ svakom x u w prethodi y i iza svakog x sledi $y\}$;

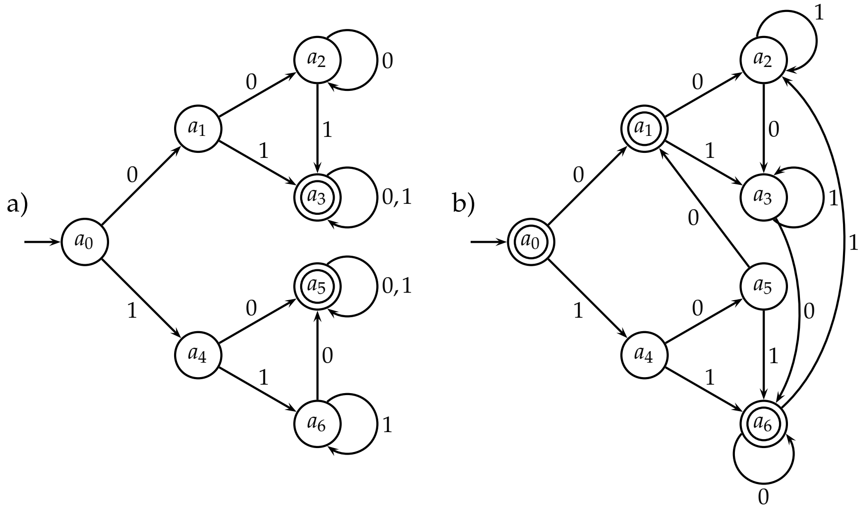
   (b) $L(A) = \{w \in \{x,y\}^* \mid w$ ima $xyxy$ kao podreč$\}$;

   (c) $L(A) = \{w \in \{x,y\}^* \mid w$ ima neparan broj slova x i paran broj slova $y\}$;

   (d) $L(A) = \{w \in \{x,y\}^* \mid w$ nema ni xx ni yy kao podreč$\}$;

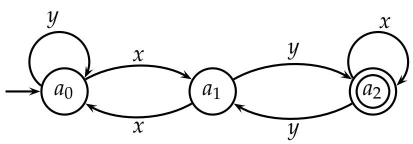
   (e) $L(A) = \{w \in \{x,y\}^* \mid w$ ima xy i yx kao podreč$\}$.

7. Minimizirati automat sa slike:



8. Neka su $\mathcal{A}_1 = (A_1, a_0^1, X, \delta_1, T_1)$ i $\mathcal{A}_2 = (A_2, a_0^2, X, \delta_2, T_2)$ automati sa disjunktnim skupovima stanja, koji raspoznaju redom jezike L_1 i L_2. Konstruisati automat $\mathcal{A} = (A, a_0, X, \delta, T)$ koji raspoznaje jezik $L_1 \cup L_2$.

9. Ako nedeterministički automat predstavljen sledećim grafom



raspoznaje jezik L, konstruisati automat koji raspoznaje jezik L^*.

10. Ako je jezik $L \subseteq X^*$ raspoznatljiv, raspoznatljiv je i jezik

$$L_{\frac{1}{3}} = \{w \mid (\exists u, v \in X^*)\, |u| = |v| = |w|,\ uvw \in L\}.$$

11. Ako je jezik L raspoznatljiv, dokazati da su raspoznatljivi i sledeći jezici:

(i) $\{u \mid u\overline{u} \in L\}$ (podsetimo, $\overline{u}$ je reverzna reč reči u);

(ii) $\{u_1 u_2 \cdots u_{2n-1} u_{2n} \mid u_2 u_1 u_4 u_3 \cdots u_{2n} u_{2n-1} \in L\}$.

12. Da li jezik $L = \{u \in \{0,1\}^* \mid |u|_1 \equiv 4 \pmod 5\}$ može da se raspozna konačnim automatom?

13. Da li jezik $L = \{x^n \mid n \in \mathbb{N}$ je složen broj$\}$ raspoznatljiv?

14. Ispitati raspoznatljivost jezika $L = \{x^{3n} y z^{2n} \mid n \in \mathbb{N}\}$.

15. Dokazati da jezik $L = \{x^m y^n \mid m, n \in \mathbb{N},\ m \geqslant n\}$ nije raspoznatljiv.

Glava 3
Kontekstno-nezavisni jezici

Kontekstno-nezavisni jezici su se pokazali veoma pogodnim za opisivanje sintakse programskih jezika, pa su našli značajne primene kod definisanja programskih jezika, u sintaksnoj analizi jezika i konstrukciji kompajlera.

Kontekstno-nezavisni jezici se mogu definisati na nekoliko načina, ekvivalentnih među sobom: kao jezici koji se mogu definisati kontekstno-nezavisnim gramatikama, kao komponente najmanjeg rešenja sistema polinomnih jednačina, potisnim automatima.

3.1. Konačni automati i konteksno-nezavisne gramatike

Zadatak 3.122. *Za svaki jezik L raspoznatljiv konačnim determinističkim automatom postoji kontekstno-nezavisna gramatika G, takva da je $L = L(G,\sigma)$. Dokazati*

Rešenje: Neka je $\mathcal{A} = (A, a_0, X, \delta, T)$ konačan deterministički automat koji raspoznaje jezik L skupom T. Konstruišimo konteksno-nezavisnu gramatiku $G = (V, X, \pi)$, u kojoj je $V = A$, $\sigma = a_0$ i

$$\pi = \{a \to xb \,|\, a \in A,\ x \in X,\ \delta(a,x) = b\} \cup \{a \to e \,|\, a \in T\}.$$

Jednostavno, indukcijom, može se pokazati da, za proizvoljna stanja $a, b \in A$ i reč $u \in X^*$, važi da je $\delta(a,u) = b$ ako i samo ako postoji izvođenje $a \overset{*}{\Rightarrow} ub$ u G. Specijalno, $u \in L$ ako i samo ako postoji izvođenje $a_0 \overset{*}{\Rightarrow} ua \Rightarrow a$, za neko stanje $a \in T$. Dakle, imamo da je $L \subseteq L(G,\sigma)$. Osim toga, jedina pravila izvođenja, koja sa desne strane nemaju neterminalne simbole, imaju oblik $a \to e$, za $a \in T$. Prema tome, $L = L(G,\sigma)$. □

Zadatak 3.123. *Ako dati, konačni nedeterministički automat $\mathcal{A} = (A, I, X, \delta, T)$ raspoznaje jezik L, konstruisati kontekstno-nezavisnu gramatiku $G = (V, X, \pi)$, za koju je $L = L(G,\sigma)$.*

Rešenje: Dati nedeterministički automat $\mathcal{A} = (A, I, X, \delta, T)$ raspoznaje jezik L skupom T. Razlikujemo dva slučaja:

1. *Slučaj $|I| = 1$*: Pretpostavimo da je $I = \{a_0\}$ i definisaćemo konteksno-nezavisnu gramatiku $G = (V, X, \pi)$, u kojoj je $V = A$, $\sigma = a_0$ i

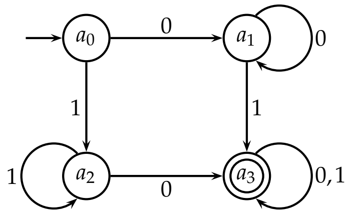
$$\pi = \{a \to xb \,|\, a \in A,\ x \in X,\ b \in \delta(a,x)\} \cup \{a \to e \,|\, a \in T\};$$

2. *Slučaj $|I| > 1$*: Konstruišimo konteksno-nezavisnu gramatiku $G = (V, X, \pi)$, u kojoj je $V = A \cup \{a_0\}$, $a_0 \notin A$, $\sigma = a_0$ i

$$\pi = \{a \to xb \,|\, a \in A, x \in X, b \in \delta(a,x)\} \cup \{a_0 \to a \,|\, a \in I\} \cup \{a \to e \,|\, a \in T\};$$

Slično kao u prethodnom zadatku dokazuje se da je $L = L(G,\sigma)$. □

Zadatak 3.124. *Konstruisati kontekstno-nezavisnu gramatiku koja generiše jezik raspoznatljiv konačnim determinističkim automatom predstavljenim grafom:*

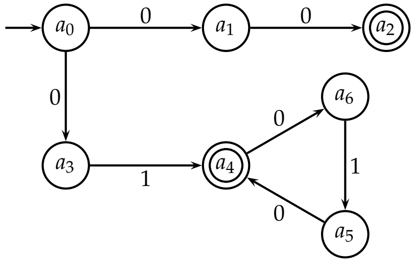


Rešenje: Na osnovu prethodnog zadatka definisaćemo gramatiku $G = (V, X, \pi)$, u kojoj je $V = \{a_0, a_1, a_2, a_3\}$, $X = \{0,1\}$ i pravila izvođenja su definisana sa:

$$\begin{array}{lll} a_0 \to 0a_1, & a_0 \to 1a_2, & a_1 \to 0a_1, \\ a_1 \to 1a_3, & a_2 \to 0a_3, & a_2 \to 1a_2, \\ a_3 \to 0a_3, & a_3 \to 1a_3, & a_3 \to e. \end{array}$$

Jasno je da je jezik datog automata $L(A) = L(G, a_0)$. □

Zadatak 3.125. *Konstruisati kontekstno-nezavisnu gramatiku koja generiše jezik koji se raspoznaje nedeterminističkim automatom predstavljenim grafom:*



Rešenje: Sa grafa prelaza ovog automata, prema Zadatku 3.122., dobijamo gramatiku $G = (V, \{0,1\}, \pi)$, sa rečnikom $V = \{a_0, a_1, a_2, a_3, a_4, a_5, a_6\}$ i pravilima izvođenja

$$\begin{array}{lll} a_0 \to 0a_1, & a_0 \to 0a_3, & a_1 \to 0a_2, \\ a_3 \to 1a_4, & a_4 \to 0a_6, & a_6 \to 1a_5, \\ a_5 \to 0a_4, & a_2 \to e, & a_4 \to e. \end{array}$$

Jasno je da je G kontekstno-nezavisna gramatika koja generiše raspoznatljiv jezik $L = L(G, a_0)$. □

3.2. Kontekstno-nezavisne gramatike

Neka je $G = (V, X, \pi)$ kontekstno-nezavisna gramatika.

Pravila oblika $\alpha \to e$, gde je $\alpha \in V \setminus X$ nazivamo *e-pravilima* u gramatici G.

Pravila oblika $\alpha \to \beta$, gde su $\alpha, \beta \in V \setminus X$, nazivamo *trivijalnim pravilima*. Jasno je da se primenom ovakvih pravila vrši samo preimenovanje pomoćnih simbola, što opravdava njihov naziv.

Kontekstno-nezavisnu gramatiku $G = (V, X, \pi)$ nazivamo *čistom gramatikom* ako je svako pravilo iz π ili oblika

$$\alpha \to \beta, \quad \text{gde je } \alpha \in V \setminus X,\ \beta \in (V \setminus X)^+,\ |\beta| > 1,$$

ili oblika

$$\alpha \to x, \quad \text{gde je } \alpha \in V \setminus X,\ x \in X.$$

Uočimo da čista gramatika nema ni e-pravila, ni trivijalnih pravila.

Za kontekstno-nezavisnu gramatiku $G = (V, X, \pi)$ kažemo da je u *Normalnoj formi Chomsky* ako je svako pravilo iz π ili oblika

$$\alpha \to \beta\gamma, \quad \text{gde su } \alpha, \beta, \gamma \in V \setminus X,$$

ili oblika

$$\alpha \to x, \quad \text{gde je } \alpha \in V \setminus X,\ x \in X.$$

Pokazaćemo da je moguće izvršiti redukciju kontekstno-nezavisne gramatike do čiste gramatike i gramatike u Normalnoj formi Chomsky.

Već smo videli da je klasa kontekstno-nezavisnih jezika zatvorena za konačnu uniju, proizvod i Kleenijevu zvezda operaciju. Nije zatvorena za konačne preseke, što znači da nije zatvorena ni za komplemente.

Zadatak 3.126. *Neka je X konačan alfabet. Ako je $L_1 \subseteq X^*$ raspoznatljiv, a $L_2 \subseteq X^*$ kontekstno-nezavisan jezik, tada je i $L_1 \cap L_2$ kontekstno-nezavisan jezik. Dokazati.*

Rešenje: Neka je $\mathcal{A} = (A, a_0, X, \delta, T)$ konačan automat koji raspoznaje jezik L_1 skupom $T \subseteq A$. Kako je klasa kontekstno-nezavisnih jezika zatvorena za uniju, to bez umanjenja opštosti dokaza možemo uzeti da je skup T jednoelementan, tj. $T = \{t\}$.

Uzmimo da je $L_2 = L(G, \sigma)$, gde je $G = (V, X, \pi)$ kontekstno-nezavisna gramatika i $\sigma \in V \setminus A$. Definišimo novu gramatiku $G' = (V', X, \pi')$ na sledeći način: uzećemo da je

$$V' = X \cup (A \times V \times A),$$

a da se π' sastoji od pravila sledećih oblika

(a) $(a, \alpha, b) \to (a, \alpha_1, c_1)(c_1, \alpha_2, c_2) \cdots (c_{m-1}, \alpha_m, b)$, gde je $\alpha \to \alpha_1\alpha_2 \cdots \alpha_m$ pravilo iz π, za $\alpha_1, \alpha_2, \ldots, \alpha_m \in V$ i $a, c_1, \ldots, c_{m-1}, b \in A$;

(b) $(a, u, b) \to u$, ako je $\delta(a, u) = b$ u automatu $\mathcal{A}$.

Neposredno se proverava da je $L_1 \cap L_2 = L(G', \sigma')$, gde je $\sigma' = (a_0, \sigma, t)$. $\square$

Zadatak 3.127. *Nekla je $G = (V, X, \pi)$ kontekstno-nezavisna gramatika. Dokazati da postoji efektivan način generisanja skupa*

$$A = \{\alpha \in V \setminus X | \alpha \overset{*}{\Rightarrow} e\}.$$

Rešenje: U dokazu Zadatka 1.42. formirali smo neopadajući niz skupova $U_i \subseteq V \setminus X$ i $U = \bigcup_{n \geqslant 1} U_n$. Dokazaćemo da je

$$U = \{\alpha \in V \setminus X | \alpha \overset{*}{\Rightarrow} e\}.$$

Posmatrajmo $\alpha \in U$. Ako $\alpha \in U_0$, onda je $\alpha \to e$ iz π, pa $\alpha \in A$. Pretpostavimo da, za proizvoljno $\alpha \in U_{n-1}$ postoji izvođenje $\alpha \overset{*}{\Rightarrow} e$. Za $\alpha \in U_n$ važi jedan od sledećih slučajeva:

(i) Ako $\alpha \in U_{n-1}$ važi indukcijska pretpostavka;

(ii) Ako postoji pravilo $\alpha \to w$ u π, za $w \in U_{n-1}^*$, postoje $\beta_1, \ldots, \beta_k \in U_{n-1}$ za koje je $w = \beta_1 \ldots \beta_k$. Prema indukcijskoj pretpostavci, za svako $i \in \{1, 2, \ldots, n\}$, postoji izvođenje $\beta_i \overset{*}{\Rightarrow} e$, što znači da postoji izvođenje $w = \beta_1 \ldots \beta_k \overset{*}{\Rightarrow} e \ldots e = e$ (vidi Zadatak 1.21.). Dakle, postoji izvođenje $\alpha \to w \overset{*}{\Rightarrow} e$, tj. važi $U \subseteq A$.

Obratnu inkluziju dokazujemo indukcijom po dužini izvođenja. Ako je $\alpha \in A$ i ako je izvođenje $\alpha \to e$ dužine jedan, onda $\alpha \in U_0 \subseteq U$. Pretpostavimo da izvođenje $\alpha \overset{*}{\Rightarrow} e$, dužine ne veće od $n-1$, daje da $\alpha \in U$. Neka je $\alpha \overset{*}{\Rightarrow} e$ izvođenje dužine n. Tada imamo

$$\alpha \Rightarrow \mu_1 \Rightarrow \cdots \Rightarrow \mu_{n-1} \Rightarrow e,$$

za $\mu_i \in U_{n-1}^*, i = \overline{1,n}$. Ako pretstavimo da je $\mu_1 = \beta_1 \ldots \beta_k \in (V \setminus X)^*$, onda je izvođenje $\beta_1 \ldots \beta_k \overset{*}{\Rightarrow} e$ dužine $n-1$. Kako je grupa jedinice trivijalna, tj. postoji samo jedno predstavljanje prazne reči $e = ee \ldots e$, to postoje izvođenja $\beta_i \overset{*}{\Rightarrow} e$, za $i = \overline{1,n}$, čija dužina ne prelazi $n-1$. Prema indukcijskoj pretpostavci $\beta_i \in U_{n-1}$, odakle je $\alpha \in U_n \subseteq U$. Dakle, važi $U = \{\alpha \in V \setminus X | \alpha \overset{*}{\Rightarrow} e\} = A$. □

Zadatak 3.128. *Dokazati da postoji algoritam kojim se može utvrditi da li jezik generisan kontekstno-nezavisnom gramatikom sadrži praznu reč ili ne.*

Rešenje: Algoritam kojim se može utvrditi da li jezik $L(G, \sigma)$, generisan gramatikom G, polazeći od $\sigma \in V \setminus X$, sadrži praznu reč zasniva se na algoritmu za nalaženje skupa U, iz prethodnog zadatka, jer je $e \in L(G, \sigma)$ ako i samo ako je $\sigma \in U$. □

Zadatak 3.129. *Dokazati da za proizvoljan kontekstno-nezavisan jezik $L \subseteq X^*$ postoji kontekstno-nezavisna gramatika $G = (V, X, \pi)$ bez e-pravila koja generiše jezik $L \setminus \{e\}$.*

Rešenje: Neka je $L = L(G_0, \sigma)$, gde je $G_0 = (V, X, \pi_0)$ kontekstno-nezavisna gramatika i $\sigma \in V \setminus X$. Uočimo skup

$$U = \{\alpha \in V \setminus X \,|\, \alpha \overset{*}{\Rightarrow} e \text{ u } G_0\}.$$

Za reč $w \in V^*$, označimo sa $D(w,U)$ skup svih reči iz V^* koje su nastale iz reči w brisanjem izvesnog broja slova iz skupa U, moguće i nijednog. Neka je $G = (V,X,\pi)$ gramatika sa skupom pravila π definisanim sa

$$\pi = \left\{(\alpha,u) \in (V \setminus X) \times V^+ \,\middle|\, (\exists w \in V^*)\ (\alpha,w) \in \pi_0 \ \&\ u \in D(w,U)\right\}.$$

Drugim rečima, za proizvoljno pravilo $\alpha \to w$ iz π_0, ako reč w sadrži k javljanja slova iz skupa U, onda to pravilo zamenjujemo sa 2^k pravila oblika $\alpha \to u$, $u \in D(w,U)$, $u \neq e$, u slučaju da je $w \notin U^*$, odnosno sa $2^k - 1$ pravila tog oblika, u slučaju da je $w \in U^*$. Jasno je da gramatika G ne sadrži e-pravila.

Dokazaćemo da je $L(G,\sigma) = L \setminus \{e\}$.

Ako je $\alpha \to u$ pravilo iz π, tada prema definiciji gramatike G imamo da je $\alpha \to w$ pravilo iz π_0, pri čemu je $u \in D(w,U)$. Uzmimo da je

$$w = v_1 v_2 \cdots v_m,$$

za neke $v_1, v_2, \dots, v_m \in V$. Iz $u \in D(w,U)$ dobijamo da je

$$u = v_{i_1} v_{i_2} \cdots v_{i_j},$$

za neke $i_1, i_2, \dots, i_j \in \{1,2,\dots,m\}$, pri čemu za svaki $i \in \{1,2,\dots,m\} \setminus \{i_1, i_2, \dots, i_j\}$ je $v_i \overset{*}{\Rightarrow} e$ u G_0. Odavde se neposredno dobija da je $\alpha \Rightarrow w \overset{*}{\Rightarrow} u$ u G_0. Prema tome, svakom pravilu $\alpha \to u$ iz π odgovara neko izvođenje $\alpha \overset{*}{\Rightarrow} u$ u G_0, što znači da je relacija π sadržana u polu-kongruenciji $\overset{*}{\Rightarrow}_{G_0}$ na V^*. Kako, prema tvrđenju Zadatka 1.19. imamo da je $\overset{*}{\Rightarrow}_G$ najmanja polukongruencija na V^* koja sadrži π, to je $\overset{*}{\Rightarrow}_G \subseteq \overset{*}{\Rightarrow}_{G_0}$, što znači da je $L(G,\sigma) \subseteq L(G_0,\sigma) = L$. Konačno, kako $e \notin L(G,\sigma)$, jer π ne sadrži e-pravila, to je $L(G,\sigma) \subseteq L \setminus \{e\}$.

Obratno, da bi smo dokazali da je $L \setminus \{e\} \subseteq L(G,\sigma)$, dovoljno je dokazati da za proizvoljne $\alpha \in V \setminus X$ i $w \in V^+$, $\alpha \overset{*}{\Rightarrow}_{G_0} w$ povlači $\alpha \overset{*}{\Rightarrow}_G w$. To ćemo dokazati indukcijom po dužini izvođenja $\alpha \overset{*}{\Rightarrow}_{G_0} w$.

Uzmimo najpre da je $\alpha \Rightarrow_{G_0} w$. Tada je $\alpha \to w$ pravilo iz π_0, jer je $\alpha \in V \setminus X$, odakle zaključujemo da $\alpha \to w$ jeste pravilo iz π, pa postoji izvođenje $\alpha \overset{*}{\Rightarrow}_G w$.

Pretpostavimo dalje da je izvođenje $\alpha \overset{*}{\Rightarrow}_{G_0} w$ dužine $m > 1$ i pretpostavimo da tvrđenje koje dokazujemo važi za sva izvođenja dužine manje od m. Uočimo najpre da je

$$\alpha \Rightarrow_{G_0} v_1 v_2 \cdots v_k \overset{*}{\Rightarrow}_{G_0} w,$$

za neke $v_1, v_2, \dots, v_k \in V$. Tada se w se može zapisati u obliku $w = u_1 u_2 \cdots u_k$, za neke $u_1, u_2, \dots, u_k \in V^*$, takve da je $v_i \overset{*}{\Rightarrow}_{G_0} u_i$, za svaki $i \in \{1,2,\dots,k\}$, pri čemu dužina ni jednog od ovih izvođenja nije veća od dužine izvođenja $v_1 v_2 \cdots v_k \overset{*}{\Rightarrow}_{G_0} w$, tj. od $m-1$. Prema tome, kad god je $u_i \neq e$, na ta izvođenja možemo primeniti indukcijsku hipotezu, čime dobijamo izvođenja $v_i \overset{*}{\Rightarrow}_G u_i$. Neka je $\{i_1, i_2, \dots, i_j\}$ skup svih elemenata iz skupa $\{1,2,\dots,k\}$ za koje je $u_i \neq e$ i neka je $v = v_{i_1} v_{i_2} \cdots v_{i_j}$. Kako je reč v nastala iz reči $v_1 v_2 \cdots v_k$ brisanjem slova iz skupa U, to je $\alpha \to v$ pravilo iz π. Sa druge strane, iz

$$v_{i_1} \overset{*}{\Rightarrow}_G u_{i_1}, \ \cdots, \ v_{i_j} \overset{*}{\Rightarrow}_G u_{i_j},$$

zbog saglasnosti relacije $\overset{*}{\Rightarrow}_G$, dobijamo da je $v = v_{i_1} v_{i_2} \cdots v_{i_j} \overset{*}{\Rightarrow}_G u_{i_1} u_{i_2} \cdots u_{i_j} \overset{*}{\Rightarrow}_G w$. Prema tome, važi $\alpha \Rightarrow_G v \overset{*}{\Rightarrow}_G w$, što je i trebalo dokazati. □

Zadatak 3.130. *Neka je data gramatika $G_0 = (V, X, \pi_0)$, nad ulaznim alfabetom $X = \{x, y\}$, $V \setminus X = \{\sigma, \lambda\}$ i sa pravilima $\sigma \to x\lambda y$, $\lambda \to x\lambda y$, $\lambda \to e$. Konstruisati gramatiku bez e-pravila koja generiše isti jezik kao G_0.*

Rešenje: Ova gramatika generiše jezik $L(G_0, \sigma) = \{x^n y^n \mid n \in \mathbb{N}\}$. Jasno je da je $U = \{\lambda\}$, a gramatika $G = (V, X, \pi)$ je dobijena metodama iz prethodnog zadatka i ima pravila $\sigma \to x\lambda y$, $\sigma \to xy$, $\lambda \to x\lambda y$, $\lambda \to xy$. Na primer, izvođenju

$$\sigma \Rightarrow x\lambda y \Rightarrow x^2\lambda y^2 \Rightarrow x^3\lambda y^3 \Rightarrow x^3 y^3$$

u G_0, odgovara izvođenje

$$\sigma \Rightarrow x\lambda y \Rightarrow x^2\lambda y^2 \Rightarrow x^2 xyy^2 = x^3 y^3$$

u gramatici G. □

Zadatak 3.131. *Dokazati da proizvoljan kontekstno-nezavisan jezik može biti generisan nekom kontekstno-nezavisnom gramatikom bez trivijalnih pravila.*

Rešenje: Neka je $L = L(G_0, \sigma)$ jezik generisan kontekstno-nezavisnom gramatikom $G_0 = (V, X, \pi_0)$, za $\sigma \in V \setminus X$. Definisaćemo gramatiku $G_1 = (V, X, \pi_1)$ na sledeći način: Uzećemo da se skup pravila π_1 sastoji od svih pravila iz π_0 i od svih parova $(\alpha, u) \in (V \setminus X) \times V^*$ takvih da postoji pomoćni simbol $\beta \in V \setminus X$, tako da je zadovoljen jedan od sledeća dva uslova:

(i) $\beta \to u$ je netrivijalno pravilo iz π_0;
(ii) α i β su povezani nekim nizom

$$\alpha \to \beta \qquad \text{ili} \qquad \alpha \to \alpha_1 \to \cdots \to \alpha_n \to \beta,\ n \in \mathbb{N},$$

trivijalnih pravila iz π_0.

Pretpostavićemo, takođe, da je $G = (V, X, \pi)$ gramatika čiji skup pravila π čine sva netrivijalna pravila iz π_1. Dokazaćemo da je $L(G_0, \sigma) = L(G, \sigma)$, pri čemu će nam gramatika G_1 služiti kao pomoćna gramatika. To znači da, za proizvoljnu reč $w \in X^*$, treba dokazati da postoji izvođenje $\sigma \overset{*}{\Rightarrow}_{G_0} w$ ako i samo ako postoji izvođenje $\sigma \overset{*}{\Rightarrow}_G w$. Kako gramatika G_1 obuhvata gramatike G_0 i G, to izvođenja u njoj nećemo posebno označavati indeksom G_1, kao što ćemo činiti u slučaju izvođenja u gramatikama G_0 i G.

Neka je $\sigma \overset{*}{\Rightarrow}_{G_0} w$. Posmatrajmo izvođenje $\sigma \overset{*}{\Rightarrow} w$ u G_1. Ako se u ovom izvođenju ne koriste trivijalna pravila, tada je jasno da je $\sigma \overset{*}{\Rightarrow}_G w$, što i treba dokazati.

Uzmimo da se u izvođenju $\sigma \overset{*}{\Rightarrow} w$ trivijalna pravila koriste k puta, gde je $k \geqslant 1$. Neka je $w_1 \Rightarrow w_2$ poslednji korak u tom izvođenju kod koga se koristi neko trivijalno pravilo, recimo pravilo $\alpha \to \beta$, $\alpha, \beta \in V \setminus X$. Tada se izvođenje $\sigma \overset{*}{\Rightarrow} w$ može zapisati u obliku $\sigma \overset{*}{\Rightarrow} w_1 \Rightarrow w_2 \overset{*}{\Rightarrow} w$, pri čemu se u izvođenju $w_2 \overset{*}{\Rightarrow} w$ ne koriste trivijalna pravila. Iz pretpostavke da je u $w_1 \Rightarrow w_2$ korišćeno pravilo $\alpha \to \beta$ dobijamo da je $w_1 = p\alpha q$ i $w_2 = p\beta q$, za neke $p, q \in V^*$.

Prema Zadatku 1.41. imamo da je $w = p'uq'$, za neke $p', u, q' \in V^*$, za koje postoje izvođenja

$$p \overset{*}{\Rightarrow} p', \qquad \beta \overset{*}{\Rightarrow} u, \qquad q \overset{*}{\Rightarrow} q'.$$

U ovim izvođenjima koriste se samo pravila koja se koriste i u izvođenju $w_2 \overset{*}{\Rightarrow} w$, što znači da se u njima ne koriste trivijalna pravila. Neka je $\beta \Rightarrow v$ prvi korak u izvođenju $\beta \overset{*}{\Rightarrow} u$. Jasno je da je, tada, $\beta \to v$ pravilo iz π_0, i to netrivijalno, jer smo videli da se u izvođenju $\beta \overset{*}{\Rightarrow} u$ ne koriste trivijalna pravila.

Prema tome, imamo da je $\alpha \to \beta$ trivijalno pravilo iz π_0 i $\beta \to v$ je netrivijalno pravilo iz π_0, odakle je $\alpha \to v$ netrivijalno pravilo iz π_1, odnosno pravilo iz π, prema definiciji gramatika G_1 i G. Korišćenjem tog pravila dobijamo da važi

$$w_1 = p\alpha q \Rightarrow pvq. \tag{3.1}$$

Sa druge strane, iz $p \overset{*}{\Rightarrow} p'$, $v \overset{*}{\Rightarrow} u$ i $q \overset{*}{\Rightarrow} q'$, prema Zadatku 1.21. dobijamo da postoji izvođenje

$$pvq \overset{*}{\Rightarrow} p'uq' = w \tag{3.2}$$

u kome se koriste samo ona pravila koja se koriste u prelazima $p \overset{*}{\Rightarrow} p'$, $v \overset{*}{\Rightarrow} u$ i $q \overset{*}{\Rightarrow} q'$. To znači da se u (3.2) ne koriste trivijalna pravila.

Dakle, iz (3.1) i (3.2) dobijamo da postoji izvođenje $w_1 \overset{*}{\Rightarrow} w$ u kome se ne koriste trivijalna pravila, čime smo dobili da se izvođenje $\sigma \overset{*}{\Rightarrow} w$ može zameniti izvođenjem u kome se koristi samo $k-1$ trivijalnih pravila, odnosno da se broj korišćenja trivijalnih pravila u tom izvođenju može smanjiti, i ako ponovimo taj postupak još $k-1$ put, dobićemo izvođenje reči w iz σ u kome se ne koristi nijedno trivijalno pravilo, što znači da je $\sigma \overset{*}{\Rightarrow}_G w$.

Obratno, uzmimo da je $\sigma \overset{*}{\Rightarrow}_G w$ i posmatrajmo ga kao izvođenje $\sigma \overset{*}{\Rightarrow} w$ u G_1. Ako se u njemu koriste samo pravila iz π_0, tada je jasno da je $\sigma \overset{*}{\Rightarrow}_{G_0} w$, što i treba dokazati.

Uzmimo da se u izvođenju pravila iz skupa $\pi_1 \setminus \pi_0$ koriste k puta. Tada ovo izvođenje možemo zapisati u obliku

$$\sigma \overset{*}{\Rightarrow} w_1 \Rightarrow w_2 \overset{*}{\Rightarrow} w,$$

pri čemu je izvođenje $w_1 \Rightarrow w_2$ zasnovano na primeni pravila $\alpha \to u$ iz $\pi_1 \setminus \pi_0$. To znači da je $w_1 = p\alpha q$ i $w_2 = puq$, za neke $p,q \in V^*$, dok prema definiciji skupa π_1 imamo da postoji netrivijalno pravilo $\beta \to u$ u π_0 i niz

$$\alpha \to \alpha_1 \to \cdots \to \alpha_n \to \beta, \qquad n \in \mathbb{N}^0,$$

trivijalnih pravila iz π_0. Sada je jasno da se izvođenje $w_1 \Rightarrow w_2$ može zameniti izvođenjem

$$w_1 = p\alpha q \Rightarrow p\alpha_1 q \Rightarrow \cdots \Rightarrow p\alpha_n q \Rightarrow p\beta q \Rightarrow puq = w_2$$

u kome se ne koriste pravila iz $\pi_1 \setminus \pi_0$. Prema tome, broj primena pravila iz $\pi_1 \setminus \pi_0$ se možemo smanjiti, sve dok ih potpuno ne eliminišemo, posle čega dobijamo izvođenje u G_0. Prema tome, $\sigma \overset{*}{\Rightarrow}_{G_0} w$, što je i trebalo dokazati. Ovim je dokaz završen. □

Zadatak 3.132. *Naći gramatiku bez trivijalnih pravila koja generiše isti jezik kao gramatika $G_0 = (V, X, \pi_0)$, u kojoj je $V \setminus X = \{\sigma, \lambda, \mu\}$, $X = \{x, y\}$ i π_0 se sastoji iz pravila*

$$\begin{array}{lll} \sigma \to \lambda, & \lambda \to \mu, & \mu \to \sigma \\ \sigma \to x^2, & \lambda \to x, & \mu \to y. \end{array}$$

Rešenje: Ova gramatika generiše jezik $L = L(G_0, \sigma) = \{x^2, x, y\}$. Kako u π_0 imamo sledeće nizove trivijalnih pravila

$$\begin{array}{ll} \mu \to \sigma, & \lambda \to \mu \to \sigma, \\ \sigma \to \lambda, & \mu \to \sigma \to \lambda, \\ \lambda \to \mu, & \sigma \to \lambda \to \mu, \end{array}$$

to netrivijalnim pravilima $\sigma \to x^2$, $\lambda \to x$ i $\mu \to y$ gramatike G_0 pridružujemo redom pravila

$$\mu \to x^2,\ \lambda \to x^2, \qquad \sigma \to x,\ \mu \to x, \qquad \lambda \to y,\ \sigma \to y,$$

čime smo dobili pravila

$$\begin{array}{lll} \sigma \to x^2, & \mu \to x^2, & \lambda \to x^2, \\ \lambda \to x, & \sigma \to x, & \mu \to x, \\ \mu \to y, & \lambda \to y, & \sigma \to y, \end{array}$$

nove gramatike $G = (V, X, \pi)$ koja nema trivijalnih pravila i koja generiše isti jezik L, tj. $L = L(G, \sigma)$. □

Zadatak 3.133. *Svaki kontekstno-nezavisan jezik u X^+ može biti generisan čistom gramatikom. Dokazati.*

Rešenje: Neka je $G = (V, X, \pi)$ kontekstno-nezavisna gramatika koja generiše jezik $L = L(G, \sigma)$, za $\sigma \in V \setminus X$. Prema prethodnom, ne umanjujući opštost dokaza, možemo uzeti da gramatika G ne sadrži e-pravila niti trivijalna pravila. Proizvoljnom slovu $x \in X$ pridružimo simbol $\beta_x \notin V$ i stavimo da je

$$V' = V \cup \{\beta_x \,|\, x \in X\}.$$

Pravila iz π su ili oblika $\alpha \to x$, gde je $\alpha \in V \setminus X$, $x \in X$, ili su oblika $\alpha \to u$, gde je $u \in V^+ \setminus V$. Pravila oblika $\alpha \to x$ nećemo dirati, dok ćemo svako pravilo iz π oblika $\alpha \to u$ zameniti pravilom oblika $\alpha \to u'$, pri čemu je u' reč dobijena iz u zamenom svakog terminalnog simbola x koji se javlja u u sa β_x. Označimo sa π' novi skup pravila dobijen iz π zadržavanjem svih pravila iz π oblika $\alpha \to x$, zamenom svih pravila oblika $\alpha \to u$ odgovarajućim pravilima oblika $\alpha \to u'$ i dodavanjem novih pravila $\beta_x \to x$, za svaki $x \in X$.

Jasno je da je $G' = (V', X, \pi')$ čista gramatika. Ostaje još samo da se dokaže da je $L(G, \sigma) = L(G', \sigma)$.

Uočimo proizvoljno neposredno izvođenje

$$p\alpha q \Rightarrow_G puq, \tag{3.3}$$

u gramatici G koje se koristi pravilom oblika $\alpha \to u$, gde su $p, q \in V^*$. Kako u G imamo pravilo $\alpha \to u'$, gde je u' reč dobijena iz u zamenom svakog terminalnog slova $x \in X$ koje se javlja u u novim pomoćnim simbolom β_x, to imamo da je $p\alpha q \Rightarrow_{G'} pu'q$, sa druge strane, reč u se može dobiti iz u' ponovnom zamenom simbola β_x sa x, pa koristeći pravila iz π' oblika $\beta_x \to x$ dobijamo da je $pu'q \overset{*}{\Rightarrow}_{G'} puq$. Prema tome, neposredno izvođenje (3.3) u G u kome se koristi pravilo oblika $\alpha \to u$ možemo zameniti u G' izvođenjem

$$p\alpha q \Rightarrow_{G'} pu'q \overset{*}{\Rightarrow}_{G'} puq,$$

pa ako to učinimo sa svim izvođenjima u G koja koriste pravila oblika $\alpha \to u$, dobićemo da svakom izvođenju $\sigma \overset{*}{\Rightarrow}_G w$, u G, $w \in X^+$, odgovara neko izvođenje $\sigma \overset{*}{\Rightarrow}_{G'} w$ u G'. Prema tome, $L(G, \sigma) \subseteq L(G', \sigma)$.

Da bi dokazali obratnu inkluziju, dovoljno je da dokažemo da svakom izvođenju $\alpha \overset{*}{\Rightarrow}_{G'} w$ u G', gde je $\alpha \in (V \setminus X)^+$ i $w \in X^+$, odgovara neko izvođenje $\alpha \overset{*}{\Rightarrow}_G w$ u G. Dokaz ćemo izvesti indukcijom po dužini izvođenja $\alpha \overset{*}{\Rightarrow}_{G'} w$.

Neka je, najpre, $\alpha \Rightarrow_{G'} w$ neposredno izvođenje i neka je $\alpha = \alpha_1 \alpha_2 \cdots \alpha_n$, za neki $n \in \mathbb{N}$ i neke $\alpha_1, \alpha_2, \ldots, \alpha_n \in V \setminus X$. Tada je, $w = w_1 w_2 \cdots w_n$, za neke $w_1, w_2, \ldots, w_n \in X^+$, pri čemu postoji izvođenje $\alpha_i \Rightarrow_{G'} w_i$, za svaki $i \in \{1, 2, \ldots, n\}$. Jasno je da je svako od tih izvođenja dobijeno primenom pravila oblika $\alpha \to x$, što znači da je $\alpha_i \Rightarrow_G w_i$, za svaki $i \in \{1, 2, \ldots, n\}$, odakle dobijamo da je

$$\alpha = \alpha_1 \alpha_2 \cdots \alpha_n \overset{*}{\Rightarrow}_G w_1 w_2 \cdots w_n = w,$$

a to je i trebalo dokazati.

Pretpostavimo dalje da je $\alpha \overset{*}{\Rightarrow}_{G'} w$ izvođenje dužine $m > 1$ i da tvrđenje koje dokazujemo važi za sva izvođenja ovog oblika dužine manje od m. Neka je $\alpha \Rightarrow_{G'} u$, za $u \in V^*$, prvi korak izvođenja $\alpha \overset{*}{\Rightarrow}_{G'} w$. Pretpostavimo da je to izvođenje dobijeno primenom pravila $\beta \to v$ iz π'. To znači da je $\alpha = p\beta q$ i $u = pvq$, za neke $p, q \in (V')^*$. Kako je $\alpha \in (V \setminus X)^+$, to je jasno da je $\beta \in V \setminus X$ i $p, q \in V^*$. Ako je $\beta \to v$, pravilo iz π, tada je $\alpha \Rightarrow_G u$, što i treba dokazati. U suprotnom je $v \in (V')^+$ i v se može zapisati u obliku $v = v_1\beta_{x_1}v_2\beta_{x_2}\cdots v_k\beta_{x_k}v_{k+1}$, za neke $v_1, v_2, \cdots, v_{k+1} \in (V \setminus X)^*$ i $x_1, x_2, \cdots, x_k \in X$. Prema definiciji pravila gramatike G', pravilo $\beta \to v$ je dobijeno iz nekog pravila $\beta \to v_1x_1v_2x_2\cdots v_kx_kv_{k+1}$ iz π. Sa druge strane, iz $u = pvq = pv_1\beta_{x_1}v_2\beta_{x_2}\cdots v_k\beta_{x_k}v_{k+1}q \overset{*}{\Rightarrow}_{G'} w$ sledi da je $w = w_1s_1w_2s_2\cdots w_ks_kw_{k+1}$, pri čemu postoje izvođenja

$$pv_1 \overset{*}{\Rightarrow}_{G'} w_1, \qquad v_i \overset{*}{\Rightarrow}_{G'} w_i, \text{ za } 2 \leqslant i \leqslant k, \qquad v_{k+1}q \overset{*}{\Rightarrow}_{G'} w_{k+1}, \tag{3.4}$$

$$\beta_{x_i} \overset{*}{\Rightarrow}_{G'} s_i, \quad \text{za } 1 \leqslant i \leqslant k, \tag{3.5}$$

dužine ne veće od dužine izvođenja $u \overset{*}{\Rightarrow}_{G'} w$. Sada na izvođenja (3.4) možemo primeniti indukcijsku pretpostavku, čime dobijamo da postoje izvođenja

$$pv_1 \overset{*}{\Rightarrow}_G w_1, \qquad v_i \overset{*}{\Rightarrow}_G w_i, \quad \text{za } 2 \leqslant i \leqslant k, \qquad v_{k+1}q \overset{*}{\Rightarrow}_G w_{k+1} \tag{3.6}$$

u G. Sa druge strane, za proizvoljan $i \in \{1, 2, \ldots, k\}$, jedino pravilo u π' koje sadrži β_{x_i} na levoj strani, jeste pravilo $\beta_{x_i} \to x_i$, odakle prema (3.5) dobijamo da je $s_i = x_i$, za svaki $i \in \{1, 2, \ldots, k\}$, što znači da je $w = w_1x_1w_2x_2\cdots w_kx_kw_{k+1}$. Prema tome, korišćenjem pravila $\beta \to v_1x_1v_2x_2\cdots v_kx_kv_{k+1}$ iz π dolazimo do izvođenja

$$\alpha = p\beta q \Rightarrow_G pv_1x_1v_2x_2\cdots v_kx_kv_{k+1}q, \tag{3.7}$$

dok iz (3.6) i iz $w = w_1x_1w_2x_2\cdots w_kx_kw_{k+1}$ dobijamo

$$pv_1x_1v_2x_2\cdots v_kx_kv_{k+1}q \overset{*}{\Rightarrow}_G w_1x_1w_2x_2\cdots w_kx_kw_{k+1} = w. \tag{3.8}$$

Dakle, iz (3.7) i (3.8) dobijamo da je $\alpha \overset{*}{\Rightarrow}_G w$, što je i trebalo dokazati. Ovim smo pokazali da je $L(G', \sigma) \subseteq L(G, \sigma)$, čime je dokaz kompletiran. □

Zadatak 3.134. *Neka je data gramatika $G = (V, X, \pi)$ sa $V \setminus X = \{\sigma, \lambda\}$, $X = \{x, y\}$ i skupom π pravila datim sa $\sigma \to x\lambda y$, $\sigma \to xy$, $\lambda \to x\lambda y$, $\lambda \to xy$. Svesti ovu gramatiku na ekvivalentnu čistu gramatiku.*

Rešenje: Primetimo da smo do ove gramatike došli u Zadatku 3.130. uklanjanjem e-pravila. Metodom koja je data u prethodnom zadatku dobijamo gramatiku $G' = (V', X, \pi')$, gde je $V' = V \cup \{\beta_x, \beta_y\}$ i skup π' se sastoji iz pravila $\sigma \to \beta_x\lambda\beta_y$, $\sigma \to \beta_x\beta_y$, $\lambda \to \beta_x\lambda\beta_y$, $\lambda \to \beta_x\beta_y$, $\beta_x \to x$, $\beta_y \to y$.

Na primer, izvođenju

$$\sigma \Rightarrow x\lambda y \Rightarrow x^2\lambda y^2 \Rightarrow x^2(xy)y^2 = x^3y^3$$

u gramatici G odgovara izvođenje

$$\sigma \Rightarrow \beta_x \lambda \beta_y \Rightarrow \beta_x^2 \lambda \beta_y^2 \Rightarrow \beta_x^2 (\beta_x \beta_y) \beta_y^2 = \beta_x^3 \beta_y^3 \overset{*}{\Rightarrow} x^3 y^3$$

u gramatici G'. $\square$

Zadatak 3.135. *Dokazati da svaki kontekstno-nezavisan jezik u X^+ jezik može biti generisan gramatikom u Normalnoj formi Chomsky.*

Rešenje: Neka je $L = L(G,\sigma)$ jezik u X^+ generisan kontekstno-nezavisnom gramatikom $G = (V, X, \pi)$. Ne umanjujući opštost dokaza možemo uzeti da je G čista gramatika. Prema tome, pravila iz π mogu biti ili oblika $\alpha \to \beta\gamma$ ili oblika $\alpha \to x$ ili oblika

$$\alpha \to \alpha_1 \alpha_2 \cdots \alpha_k, \qquad \text{gde su } \alpha, \alpha_1, \ldots, \alpha_k \in V \setminus X,\ k \geqslant 3. \tag{3.9}$$

Kako su pravila oblika $\alpha \to \beta\gamma$ i $\alpha \to x$ već u Normalnoj formi Chomsky, to ćemo se zadržati samo na pravilima oblika (3.9), i svako od takvih pravila ćemo zameniti nekim novim pravilima koja su u Normalnoj formi Chomsky.

Naime, svakom pravilu oblika (3.9) pridružićemo skup $\{\gamma_1, \gamma_2, \ldots, \gamma_{k-2}\}$ novih pomoćnih simbola, tako da skupovi ovog oblika koji odgovaraju različitim pravilima iz π oblika (3.9) budu međusobno disjunktni. Pravilo (3.9) ćemo zameniti nizom pravila

$$\alpha \to \alpha_1 \gamma_1,\ \gamma_1 \to \alpha_2 \gamma_2,\ \cdots,\ \gamma_{k-3} \to \alpha_{k-2} \gamma_{k-2},\ \gamma_{k-2} \to \alpha_{k-1} \alpha_k. \tag{3.10}$$

Kada to učinimo sa svim pravilima iz π oblika (3.9), dobićemo novi rečnik V' i novi skup pravila π', odnosno dobićemo novu gramatiku $G' = (V', X, \pi')$, koja je očigledno u Normalnoj formi Chomsky.

Ostaje da dokažemo da je $L = L(G,\sigma) = L(G',\sigma)$.

Primetimo najpre, da svakom pravilu iz π oblika (3.9) odgovara u gramatici G' izvođenje

$$\alpha \Rightarrow_{G'} \alpha_1 \gamma_1 \Rightarrow_{G'} \alpha_1 \alpha_2 \gamma_2 \Rightarrow_{G'} \cdots \Rightarrow_{G'} \alpha_1 \alpha_2 \cdots \alpha_k,$$

odakle dobijamo da svakom izvođenju u G odgovara neko izvođenje u G'. Prema tome, $L(G,\sigma) \subseteq L(G',\sigma)$.

Uočimo dalje gramatiku $G'' = (V', X, \pi \cup \pi')$. Kako ona obuhvata obe gramatike G i G', to neće biti opasnosti od zabune ako pri označavanju izvođenja u G'' izostavimo pisanje indeksa G''.

Da bi smo dokazali inkluziju $L = (G',\sigma) \subseteq L(G,\sigma)$, dovoljno je dokazati da se iz proizvoljnog izvođenja

$$\sigma = w_0 \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_m = w, \tag{3.11}$$

gde je $m \in \mathbb{N}$ i $w \in X^*$, mogu eliminisati primene svih pravila iz $\pi' \setminus \pi$, odnosno svi simboli iz $V' \setminus V$.

Pretpostavimo da se pravila iz $\pi' \setminus \pi$ u izvođenju (3.11) koriste n puta, gde je $n \in \mathbb{N}$, i dokažimo da se taj broj može smanjiti, tj. da postoji neko drugo izvođenje $\sigma \overset{*}{\Rightarrow} w$ u G'' u kome se pravila iz $\pi' \setminus \pi$ koriste manje od n puta.

Neka je $w_{i_1} \Rightarrow w_{i_1+1}$, gde je $0 \leqslant i_1 < m$, proizvoljno neposredno izvođenje pri kome se koristi pravilo $\alpha \to \alpha_1\gamma_1$ iz (3.10). To znači da važi

$$w_{i_1} = p_1\alpha q_1 \Rightarrow p_1\alpha_1\gamma_1 q_1 = w_{i_1+1},$$

za neke $p_1, q_1 \in (V')^*$. Simbol γ_1, koga smo ovom prilikom uveli u izvođenje (3.11), izgubiće se prilikom primene pravila $\gamma_1 \to \alpha_2\gamma_2$ u nekom neposrednom izvođenju $w_{i_2} \Rightarrow w_{i_2+1}$, gde je $i_1 < i_2 < m$.Tokom izvođenja $w_{i_1+1} \overset{*}{\Rightarrow} w_{i_2}$, njega ćemo samo prepisivati, dok ćemo zamene pomoćnih simbola rečima iz $(V')^*$, prema pravilima gramatike G'', u rečima $w_{i_1+1}, \dots, w_{i_2-1}$, vršiti levo i desno od tog simbola. Na taj način dobijamo da je

$$w_{i_2} = p_2\alpha_1^{(2)}\gamma_1 q_2,$$

gde su $p_2, \alpha_1^{(2)}, q_2 \in (V')^*$ reči za koje važi

$$p_1 \overset{*}{\Rightarrow} p_2, \qquad \alpha_1 \overset{*}{\Rightarrow} \alpha_1^{(2)}, \qquad q_1 \overset{*}{\Rightarrow} q_2. \tag{3.12}$$

Pri tome su sva pravila koja se koriste u sva tri izvođenja iz (3.12) tačno ona koja se koriste u izvođenju $w_{i_1+1} \overset{*}{\Rightarrow} w_{i_2}$, odakle lako zaključujemo da ukupan broj primena pravila iz $\pi' \setminus \pi$ u izvođenjima iz (3.12) nije veći od broja primena tih pravila u izvođenju $w_{i_1+1} \overset{*}{\Rightarrow} w_{i_2}$.

Nastavljajući na isti način dalje, dobićemo da se izvođenje (3.11) može zapisati u obliku

$$\begin{aligned}
\sigma = w_0 \overset{*}{\Rightarrow} w_{i_1} &= p_1\alpha q_1 \Rightarrow p_1\alpha_1\gamma_1 q_1 = w_{i_1+1}\\
\overset{*}{\Rightarrow} w_{i_2} &= p_2\alpha_1^{(2)}\gamma_1 q_2 \Rightarrow p_2\alpha_1^{(2)}\alpha_2\gamma_2 q_2 = w_{i_2+1}\\
\overset{*}{\Rightarrow} w_{i_3} &= p_3\alpha_1^{(3)}\alpha_2^{(3)}\gamma_2 q_3 \Rightarrow p_3\alpha_1^{(2)}\alpha_2^{(3)}\alpha_3\gamma_3 q_3 = w_{i_3+1}\\
\dots \quad \dots \quad &\dots \quad \dots \quad \dots\\
\overset{*}{\Rightarrow} w_{i_{k-1}} &= p_{k-1}\alpha_1^{(k-1)}\cdots\alpha_{k-2}^{(k-1)}\gamma_{k-2}q_{k-1}\\
\Rightarrow p_{k-1}\alpha_1^{(k-1)}&\cdots\alpha_{k-2}^{(k-1)}\alpha_{k-1}\alpha_k q_{k-1} = w_{i_{k-1}+1} \overset{*}{\Rightarrow} w,
\end{aligned}$$

pri čemu su $p_j, \alpha_1^{(j)}, \dots, \alpha_{j-1}^{(j)}, q_j \in (V')^*$, $1 \leqslant j \leqslant k-1$, reči za koje važi

$$p_j \overset{*}{\Rightarrow} p_{j+1},\ \alpha_1^{(j)} \overset{*}{\Rightarrow} \alpha_1^{(j+1)},\ \cdots,\ \alpha_{j-1}^{(j)} \overset{*}{\Rightarrow} \alpha_{j-1}^{(j+1)},\ q_j \overset{*}{\Rightarrow} q_{j+1}, \tag{3.13}$$

za $1 \leqslant j \leqslant k-2$. Pri tome, kao i napred, dobijamo da ukupan broj primena pravila iz $\pi' \setminus \pi$ u izvođenjima iz (3.13) nije veći od broja primena tih pravila u izvođenju $w_{i_j+1} \overset{*}{\Rightarrow} w_{i_{j+1}}$, za svaki j, $1 \leqslant j \leqslant k-2$.

Prema tome, imamo izvođenja

$$p_1 \overset{*}{\Rightarrow} p_{k-1}, \qquad \alpha_i \overset{*}{\Rightarrow} \alpha_i^{(k-1)}, \text{ za } 1 \leqslant i \leqslant k-2, \qquad q_1 \overset{*}{\Rightarrow} q_{k-1}, \tag{3.14}$$

u kojima ukupan broj primena pravila iz $\pi' \setminus \pi$ nije veći od ukupnog broja primena tih pravila u izvođenjima

$$w_{i_1+1} \overset{*}{\Rightarrow} w_{i_2},\ w_{i_2+1} \overset{*}{\Rightarrow} w_{i_3},\ \cdots,\ w_{i_{k-2}+1} \overset{*}{\Rightarrow} w_{i_{k-1}}. \tag{3.15}$$

Sada, iz (3.14) dobijamo da postoji izvođenje

$$p_1\alpha_1 \cdots \alpha_{k-2}\alpha_{k-1}\alpha_k q_1 \overset{*}{\Rightarrow} p_{k-1}\alpha_1^{(k-1)} \cdots \alpha_{k-2}^{(k-1)}\alpha_{k-1}\alpha_k q_{k-1}, \tag{3.16}$$

u kome broj primena pravila iz $\pi' \setminus \pi$ nije veći od ukupnog broja primena tih pravila u izvođenjima iz (3.15).

Dakle, imamo da važi

$$\begin{aligned} \sigma = w_0 &\overset{*}{\Rightarrow} w_{i_1} = p_1\alpha q_1 \\ &\Rightarrow p_1\alpha_1 \cdots \alpha_{k-2}\alpha_{k-1}\alpha_k q_1 \\ &\overset{*}{\Rightarrow} p_{k-1}\alpha_1^{(k-1)} \cdots \alpha_{k-2}^{(k-1)}\alpha_{k-1}\alpha_k q_{k-1} = w_{i_{k-1}+1} \\ &\overset{*}{\Rightarrow} w, \end{aligned} \tag{3.17}$$

prema (3.16), koristeći pravilo $\alpha \to \alpha_1\alpha_2 \cdots \alpha_k$ iz π. Jasno je da se u izvođenju (3.17) pravila iz $\pi' \setminus \pi$ primenjuju manji broj puta nego u (3.11), što je i trebalo dokazati. Ovim je dokaz završen. □

Zadatak 3.136. *Gramatiku čiju smo redukciju doveli u Zadatku* 3.134. *do čiste gramatike u kojoj su* $\sigma \to \beta_x\lambda\beta_y$, $\sigma \to \beta_x\beta_y$, $\lambda \to \beta_x\lambda\beta_y$, $\lambda \to \beta_x\beta_y$, $\beta_x \to x$, $\beta_y \to y$ *pravila izvođenja, svesti na gramatiku u Normalnoj formi Chomsky.*

Rešenje: Ako nastavimo redukciju dalje, na način prikazan u prethodnom zadatku, dolazimo do gramatike u Normalnoj formi Chomsky sa pravilima

$$\begin{array}{llll} \sigma \to \beta_x\gamma, & \gamma \to \lambda\beta_y, & \sigma \to \beta_x\beta_y, & \lambda \to \beta_x\delta, \\ \delta \to \lambda\beta_y, & \lambda \to \beta_x\beta_y, & \beta_x \to x, & \beta_y \to y, \end{array}$$

pri čemu uvodimo nove pomoćne simbole γ i δ.

Prema tome, izvođenje

$$\begin{aligned} \sigma &\Rightarrow \beta_x\gamma \Rightarrow x\gamma \Rightarrow x\lambda\beta_y \Rightarrow x\beta_x\delta\beta_y \Rightarrow x^2\delta\beta_y \Rightarrow x^2\lambda\beta_y^2 \\ &\Rightarrow x^2\beta_x\beta_y^3 \Rightarrow x^3\beta_y^3 \Rightarrow x^3y\beta_y^2 \Rightarrow x^3y^2\beta_y \Rightarrow x^3y^3. \end{aligned}$$

jeste jedno od izvođenja reči x^3y^3 u ovoj gramatici. □

Jedna od najvažnijih primena gramatike u Normalnoj formi Chomsky jeste dokaz tvrđenja da je svaka konteksno-nezavisna gramatika odlučiva.

Zadatak 3.137. *Za svaku konteksno-nezavisnu gramatiku* $G = (V, X, \pi)$ *i svaku reč* $u \in X^*$ *postoji algoritam koji utvrđuje da li reč pripada jeziku* $L = L(G)$ *generisanom datom gramatikom. Dokazati.*

Rešenje: Neka je $u \in X^*$ data reč. Na osnovu Zadatka 1.42. postoji algoritam kojim se utvrđuje da li gramatika G generiše praznu reč ili ne. Dakle, dovoljno je posmatrati reči $u \neq e$. Prema Zadatku 3.135. možemo, bez umanjenja opštosti, pretpostaviti da je gramatika G u Normalnoj formi Chomsky.

Pretpostavimo da se reč u u gramatici G može izvesti u k koraka. Tada je jasno da se izvođenja $\alpha \to p$, za $\alpha \in V \setminus X$ i $p \in X$, koriste tačno onoliko puta kolika je dužina reči u. Dakle, izvođenja oblika $\alpha \to \beta\gamma$, $\alpha,\beta,\gamma \in V \setminus X$, koriste se tačno $k - |u|$ puta. Kako izvođenja ovog oblika svaki put povećavaju dužinu reči za jedan, zaključujemo da je $|u| = k - |u| + 1$, odakle dobijamo $k = 2|u| - 1$. To znači da na osnovu dužine reči u možemo zaključiti koliko koraka ima eventualno izvođenje te reči u gramatici G. Tako, da bi utvrdili da li reč u može biti generisana nekim izvođenjem u gramatici G, dovoljno je da ispitamo sva moguća izvođenja dužine $2|u| - 1$. □

3.3. Lema o napumpavanju za kontekstno-nezavisne jezike

Formulisaćemo Lemu o napumpavanju za konteksno-nezavisne jezike i pokazaćemo kako se ona koristi za dokaz da dati jezik nije kontekstno-nezavisan.

Teorema 3.17. (Lema o napumpavanju) *Neka je X konačan alfabet i $L \subseteq X^*$ je beskonačan kontekstno-nezavisan jezik. Tada postoji broj $n \in \mathbb{N}$ takav da svaka reč $w \in L$ dužine $|w| \geqslant n$ može biti zapisana u obliku*

$$w = puqvr,$$

gde su $p,u,q,v,r \in X^*$ reči za koje važi:

(a) $|uv| \geqslant 1$;
(b) $|uqv| \leqslant n$;
(c) $pu^m qv^m r \in L$, *za svaki* $m \in \mathbb{N}^0$.

Zadatak 3.138. *Dokazati da jezik $L = \{x^n y^n z^n \,|\, n \in \mathbb{N}\}$ nije kontekstno-nezavisan.*

Rešenje: Pretpostavimo suprotno, da je jezik L kontekstno-nezavisan. Tada je, za broj $n \in \mathbb{N}$, reč $w = x^n y^n z^n$ dužine $|w| \geqslant n$, pa može biti zapisana u obliku $w = puqvr$, za $p,u,q,v,r \in X^*$. Kako je $|uqv| \leqslant n$, to ova podreč sadrži najviše dva različita slova x i y ili y i z. Na osnovu Leme o napumpavanju, reč $w_1 = pu^2qv^2r$ bi, takođe, trebalo da pripada jeziku L. To je nemoguće, jer u reči w_1 nemamo isti broj slova x, y i z. Ovim smo pokazali da jezik L nije konteksno-nezavisan. □

Zadatak 3.139. *Dokazati da jezik $L = \{x^n y^n x^n \,|\, n \in \mathbb{N}\}$ nije konteksno-nezavisan.*

Rešenje: Primećujemo da je jezik L specijalan slučaj jezika $\{x^n y^n z^n \,|\, n \in \mathbb{N}\}$, koji dobijamo kada slovo z zamenimo sa x, pa L ne zadovoljava uslove Leme o napumpavanju, tj. nije konteksno-nezavisan (videti prethodni zadatak) . □

Zadatak 3.140. *Dokazati da jezik* $L = \{uu \,|\, u \in \{0,1\}^*\}$ *nije kontekstno-nezavisan.*

Rešenje: Pretpostavimo suprotno, da je jezik L kontekstno-nezavisan. Tada je, za broj $n \in \mathbb{N}$, reč $w = 0^n1^n0^n1^n \in L$ dužine $|w| > n$, pa može biti zapisana u obliku $w = puqvr$, za $p,u,q,v,r \in X^*$. Kako je $|uqv| \leqslant n$, to ova podreč sadrži ili neku podreč reči 0^n1^n ili podreč od 1^n0^n. U svakom slučaju pqr sadrži blok nula (ili jedinica) dužine manje od n i blok nula (ili jedinica) dužine tačno n. Odavde je jasno da reč $w_1 = pu^0qv^0r = pqr$ ne može biti zapisana u obliku $u'u'$, pa $w_1 \notin L$, što je kontrdikcija sa trćim uslovom Leme o napumpavanju. Dakle, jezik L nije kontekstno-nezavisan. □

Zadatak 3.141. *Dokazati da jezik* $L = \{0^n1^m \,|\, m \leqslant n^2\}$ *nije kontekstno-nezavisan.*

Rešenje: Pretpostavimo suprotno, da je jezik L kontekstno-nezavisan. Tada, za broj $n \in \mathbb{N}$, reč $w = 0^t1^{t^2} \in L$, gde je $t = max\{n,2\}$, može biti zapisana u obliku $w = puqvr$, za $p,u,q,v,r \in X^*$. Razlikovaćemo dva slučaja:

1. uv sadrži bar jedno 0.

Prema drugom uslovu Leme o napumpavanju, važi $pqr = 0^i1^j$, za $i \leqslant t-1$ i $j \geqslant t^2 - n$. Kako je $t = max\{n,2\}$, to je $j \geqslant t^2 - n > t^2 - 2t + 1 = (t-1)^2 \geqslant i$, što znači da $w_1 = pu^0qv^0r = pqr \notin L$, čime smo došli do kontradikcije.

2. $uv = 1^j$, za neki $1 \leqslant j \leqslant n$.

U ovom slučaju dobijamo da reč $w_1 = pu^2qv^2r = 0^t1^{t^2+j} \notin L$, što je takođe kontradikcija sa trećim uslovom leme.

Dakle, dati jezik nije kontekstno-nezavisan. □

Zadatak 3.142. *Ako jezik* $L \subseteq X^*$*, nad alfabetom* X*, ima svojstvo da je svaki njegov podskup kontekstno-nezavisan, onda je* L *konačan jezik. Dokazati.*

Rešenje: Pretpostavimo suprotno, da je jezik L beskonačan.Tada postoji beskonačan podskup $A \subseteq L$ koji ima sledeće svojstvo: Za proizvoljnu reč $u \in A$, ni jedna reč v dužine $|u| + 1 \leqslant |v| \leqslant 2|u|$ nije u A. Ovakav skup A možemo definisati rekurzivno na sledeći način:

Pođimo od $A = \emptyset$.

U nultom koraku biramo najkraću nepraznu reč $u_0 \in L$ i smeštamo je u A.

Na kraju n-tog koraka izabrali smo reči $u_0, u_1, \ldots, u_n \in L$, takve da važi $|u_i| > 2|u_{i-1}|$, za $i = 1,2,\ldots n$ i smestili smo ih u A.

U $n+1$-om koraku biramo reč $u_{n+1} \in L$ sa svojstvom $|u_{n+1}| > 2|u_n|$ i smeštamo je u A.

Kako je jezik L beskonačan ova konstrukcija je dobro definisana i prema pretpostavci zadatka jezik A je kontekstno-nezavisan. To znači da važe uslovi Leme o napumpavanju, pa proizvoljnu reč $w \in A$ dužine $|w| > n$ možemo napisati u obliku $w = puqvr$, za $p,u,q,v,r \in X^*$, pri čemu je $0 < |uv| \leqslant |uqv| \leqslant n$ i $pu^nqv^nr \in A$, za svaki $n \geqslant 0$. Međutim, primećujemo da važi

$$|w| < |pu^2qv^2r| \leqslant |w| + n < 2|w|,$$

odakle, prema definiciji jezika A, zaključujemo da $pu^2qv^2r \notin A$, tj. da jezik A nije kontekstno-nezavisan.

Dakle, došli smo do kontradikcije, pa jezik L mora biti konačan. □

Zadatak 3.143. (a) *Dokazati da presek dva kontekstno-nezavisna jezika ne mora biti kontekstno-nezavisan jezik.*

(b) *Dokazati da komplement kontekstno-nezavisnog jezika ne mora biti kontekstno-nezavisan jezik.*

Rešenje: (a) Neka je dati alfabet $X = \{x, y\}$ i $L = \{x^n y^n x^m \,|\, m, n \in \mathbb{N}\}$.

Jezik L može se predstaviti kao proizvod jezika $\{x^n y^n \,|\, n \in \mathbb{N}\}$, za koji smo videli da je kontekstno-nezavisan, i jezika $\{x\}^+$, za koji znamo da je raspoznatljiv, pa time i kontekstno-nezavisan. Prema tome, L je kontekstno-nezavisan jezik, kao proizvod dva kontekstno-nezavisna jezika. Slično, i $L' = \{x^m y^n x^n \,|\, m, n \in \mathbb{N}\}$ jeste kontekstno-nezavisan jezik.

Međutim, kao što smo pokazali u prethodnom zadatku, jezik

$$L \cap L' = \{x^n y^n x^n \,|\, n \in \mathbb{N}\},$$

nije kontekstno-nezavisan.

(b) Ovo sledi iz (a), jer za proizvoljna dva jezika $L_1, L_2 \subseteq X^*$ važi

$$L_1 \cap L_2 = (L_1^c \cup L_2^c)^c,$$

gde je L_i^c, $i = 1, 2$ oznaka za komplement jezika L_i u X^*. □

3.4. Stabla izvođenja i parsirajuća stabla

Sada ćemo izvođenja u kontekstno-nezavisnih gramatikama predstaviti grafovima ili, preciznije stablima.

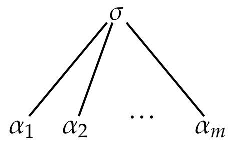
Stabla izvođenja

Neka je data kontekstno-nezavisna gramatika $G = (V, X, \pi)$, $\sigma \in V \setminus X$ i izvođenje $\sigma \overset{*}{\Rightarrow} w$ u G, gde je $w \in V^*$. Tada tom izvođenju odgovara stablo D označeno elementima iz V koje definišemo na sledeći način:

Neka je izvođenje $\sigma \overset{*}{\Rightarrow} w$ dato sa:

$$\sigma = w_0 \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n = w.$$

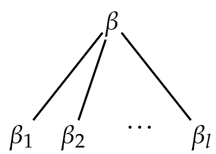
Koren stabla D označen je sa σ. Ako se u izvođenju $\sigma \Rightarrow w_1$ koristi pravilo oblika $\sigma \to \alpha_1\alpha_2\cdots\alpha_m$, gde su $\alpha_1,\alpha_2,\dots,\alpha_m \in V$, tada stablo izvođenja $\sigma \Rightarrow w_1$, u oznaci D_1, definišemo sa:



Dalje, neka je definisano stablo izvođenja

$$\sigma = w_0 \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_k,$$

gde je $1 \leqslant k < n$, koje ćemo označiti sa D_k. Pretpostavimo da je neposredno izvođenje $w_k \Rightarrow w_{k+1}$ zasnovano na primeni pravila oblika $\beta \to \beta_1\beta_2\cdots\beta_l$, gde su $\beta_1,\beta_2,\dots,\beta_l \in V$. To znači da se jedno od pojavljivanja simbola β u reči w_k zamenjuje sa $\beta_1\beta_2\cdots\beta_l$. Kako tom pojavljivanju simbola β u w_k odgovara jedno određeno pojavljivanje tog simbola kao oznake lista u D_k, to ćemo stablo D_{k+1} dobiti na taj način što ćemo tom čvoru u D_k prikačiti stablo



Na ovaj način smo induktivno definisali niz stabala $D_1,D_2,\dots,D_n$. Stablo D_n nazivamo *stablom izvođenja* $\sigma = w_0 \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n = w$.

Primetimo da, koristeći grafičko predstavljanje stabla D, grane koje polaze iz proizvoljnog čvora stabla D možemo urediti uzimajući, na primer, njihov redosled sa leva na desno. Slično možemo urediti i puteve u stablu D koji polaze iz korena. Naime, za svaka dva takva puta p_1 i p_2 postoji čvor a stabla D u kome se oni razdvajaju, odnosno račvaju, pa ako se grana puta p_1 koja izlazi iz a nalazi levo od odgovarajuće grane puta p_2, tada ćemo reći da se put p_1 nalazi levo od puta p_2.

Konačno, za dva lista l_1 i l_2 stabla D ćemo reći da se list l_1 nalazi levo od l_2 ako se put koji ide od korena do l_1 nalazi levo od puta koji ide od korena do l_2.

Ako kod stabla D izvođenja $\sigma \overset{*}{\Rightarrow} w$, $w \in V^*$, čitamo oznake listova sa leva na desno, pročitaćemo upravo reč w, za koju kažemo da je *rezultat stabla D*.

Parsirajuća stabla (stabla raščlanjenja)

Parsiranje (*raščlanjenje*) je jedan od najvažnijih zadataka u dizajnu kompajlera. To je postupak formiranja tzv. parsirajućeg stabla za datu reč u, koje se dobija iz pravila izvođenja kontekstno-nezavisne gramatike. Ima više tehnika parsiranja. Mi ćemo se zadržati na najopštijoj, ali ne uvek najefikasnijoj, *top-down* metodi parsiranja.

Neka je $G = (V, X, \pi)$ kontekstno-nezavisna gramatika. Reč $u \in V^*$ nazivamo *rečeničnom formom* u gramatici G ako postoji izvođenje $\sigma \overset{*}{\Rightarrow} u$, a *levom rečeničnom formom* ako postoji krajnje-levo izvođenje od σ do u.

Za svaku kontekstno-nezavisnu gramatiku G definisaćemo *krajnje-levi graf* $g(G)$ na sledeći način:

(a) Čvorovi u $g(G)$ su leve rečenične forme u gramatici G;

(b) Za dve leve rečenične forme u_1 i u_2, ako u π postiji izvođenje $\alpha \to u$, takvo da je $u_1 = v\alpha w$ i $u_2 = vuw$, za neki $v \in X^*$ i $w \in V^*$, onda postoji usmerena grana iz u_1 u u_2.

Ako leve rečenične forme u gramatici G imaju jedinstveno krajnje-levo izvođenje iz početnog simbola σ, tada je $g(G)$ stablo čiji je koren početni simbol σ.

Parsirajuće stablo reči $u \in X^*$ možemo odrediti jednostavnim pretraživanjem grafa $g(G)$.

Kako graf $g(G)$ može biti beskonačan, to nije sigurno da će se algoritam pretraživanja zaustaviti za sve ulazne reči.

Međutim, ukoliko kontekstno-nezavisnu gramatiku svedemo na gramatiku bez e-pravila pretraživanje će se sigurno zaustaviti. Dužina rečeničnih formi se, u ovom slučaju, ne smanjuje tokom izvođenja, tako da određivanje izvođenja $\sigma \overset{*}{\Rightarrow} u$ u $g(G)$ jednostavno možemo ograničiti na podgraf u kome dužina puteva ne prelazi $|u|$.

Kontekstno-nezavisna gramatika je *dvoznačna* ako postoji reč $u \in X^*$ koja ima dva parsirajuća stabla. Sintaksička struktura reči, koju daje parsirajuće stablo G indukuje i semantičku strukturu (značenje) te reči u $L(G)$. Kada postoje dva parsirajuća stabla, teško je odrediti koja je semantička struktura korektna. Zato uvek težimo konstrukciji jednoznačne konteksno-nezavisne gramatike.

Zadatak 3.144. *Neka je data gramatika $G = (V, X, \pi)$, u kojoj je $V \setminus X = \{\sigma, \lambda, \mu\}$, $X = \{x, y, z\}$ i pravila su*

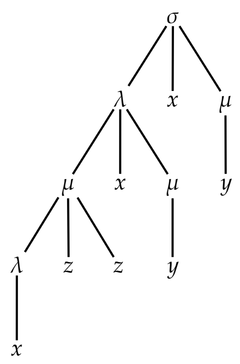
$$\sigma \to \lambda x\mu, \quad \lambda \to \mu x\mu, \quad \mu \to \lambda z^2, \quad \mu \to y, \quad \lambda \to x.$$

Ispitati da li reč $xz^2(xy)^2$ pripada jeziku $L = L(G, \sigma)$.

Rešenje: Primetimo da postji stablo izvođenja, prikazano na Slici 3.144., koje odgovara izvođenju

$$\sigma \Rightarrow \lambda x\mu \Rightarrow \mu x\mu x\mu \Rightarrow \lambda z^2 x\mu x\mu \Rightarrow \lambda z^2 xyx\mu \Rightarrow xz^2 xyx\mu \Rightarrow xz^2 xyxy = xz^2(xy)^2,$$

jer čitanjem oznaka listova, sa leva na desno, dobijamo reč $xzzxyxy$. Ovim smo dokazali da je $xz^2(xy)^2 \in L(G, \sigma)$. $\square$

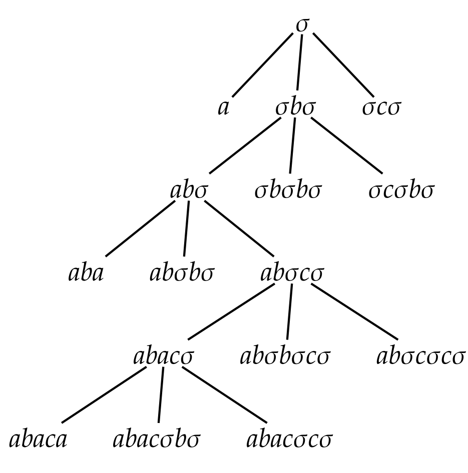


Slika 3.1 Stablo izvođenja za Zadatak 3.144..

Zadatak 3.145. *Odrediti da li reč abaca pripada jeziku $L(G,\sigma)$, ako je $G = (V,X,\pi)$ data gramatika , u kojoj je $V \setminus X = \{\sigma\}$, $X = \{a,b,c\}$, a pravila izvođenja su*

$$\sigma \longrightarrow \sigma b\sigma, \quad \sigma \longrightarrow \sigma c\sigma, \quad \sigma \longrightarrow a.$$

Rešenje: Parsirajuće stablo ćemo dobiti pretraživanjem po dubini grafa $g(G)$ (tzv. Depth-first search algoritam). Čvorove ćemo ispitivati sa leva na desno sve dok ne dobijemo traženu reč. Pri tome ispitujemo samo čvorove do kojih dužina puta od početnog simbola σ nije veća od dužine $|abaca| = 5$ i možemo odbaciti svaki čvor u kome naiđemo na dva pojavljivanja slova b ili slova c.



Napomena 5. *Traženu reč možemo naći i pretraživanjem stabla po širini (tzv. Breadth-first search algoritam).*

Pretraživanje stabla prekidamo čim nađemo traženu reč ili kada dužina puta od korena premaši 5. □

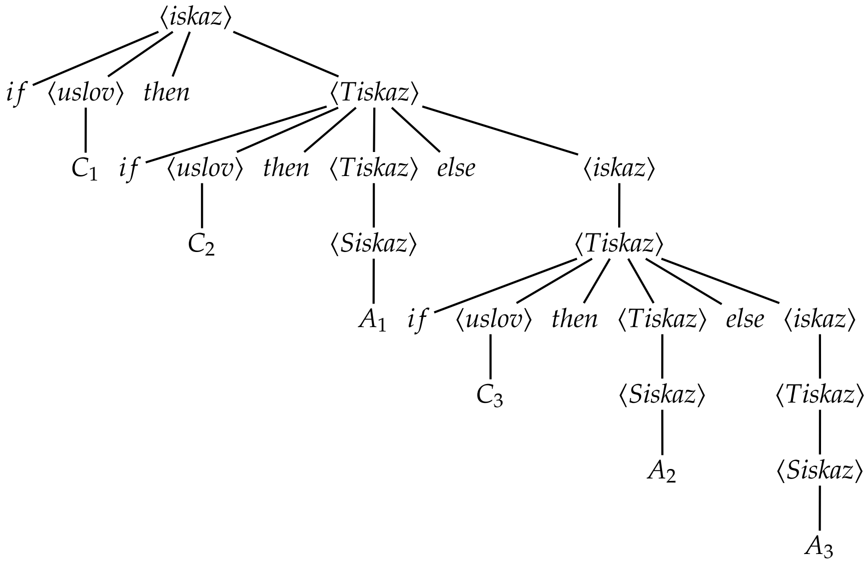
Zadatak 3.146. *Neka je $G = (V, X, \pi)$ kontekstno-nezavisna gramatika u kojoj su simboli oblika $\langle \cdots \rangle$ pomoćni simboli, dok su ostali simboli terminalni, a pravila izvođenja su:*

$$\begin{aligned}
\langle iskaz \rangle &\to if\ \langle uslov \rangle\ then\ \langle Tiskaz \rangle + \langle Tiskaz \rangle \\
\langle Tiskaz \rangle &\to if\ \langle uslov \rangle\ then\ \langle Tiskaz \rangle\ else\ \langle iskaz \rangle + \langle Siskaz \rangle \\
\langle uslov \rangle &\to C_1 + C_2 + C_3 \\
\langle Siskaz \rangle &\to A_1 + A_2 + A_3.
\end{aligned}$$

Naći dva različita parsirajuća stabla za rečenicu:

if C_1 then if C_2 then A_1 else if C_3 then A_2 else A_3.

Rešenje: Videćemo da je moguće naći dva parsirajuća stabla za datu rečenicu:



Primetimo da iz ovog parsirajućeg stabla ne možemo odrediti šta se dešava ako uslov C_1 ne važi. Očigledno je da se ne dobija ni jedan od jednostavnih iskaza (Siskaz) A_1, A_2 ili A_3.

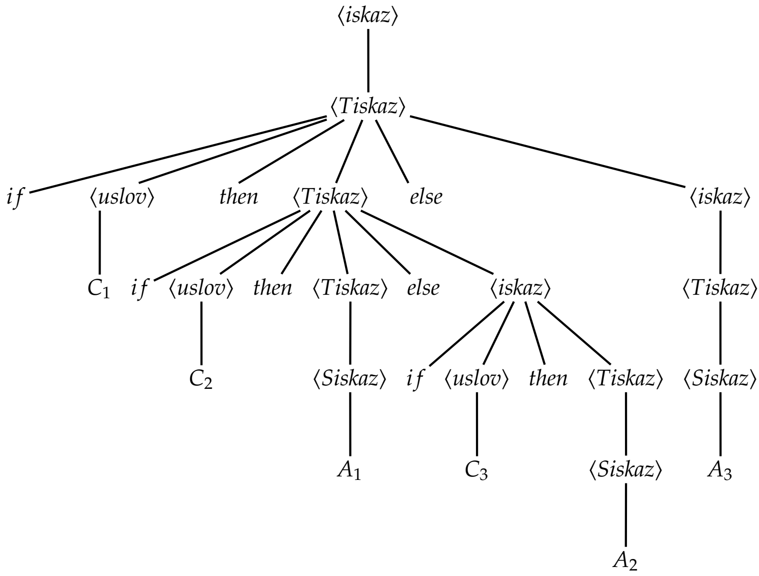
Videćemo da se u drugom parsirajućem stablu, prikazanom na Slici 3.2, u slučaju kada uslov C_1 ne važi dobija jednostavan iskaz A_3.

Dakle, data gramatika je dvoznačna i vidimo da je teško odrediti značenje rečenice iz dva parsirajuća stabla. Zato je poželjno konstruisati jednoznačnu gramatiku uvek kada je to moguće. □

Zadatak 3.147. (a) *Dokazati da je konteksno-nezavisna gramatika $G = (V, X, \pi)$ aritmetičkih izraza sa dve promenljive $X = \{a, b\}$ i pravilima izvođenja*

$$\sigma \to \sigma + \sigma, \ \ \sigma \to \sigma - \sigma, \ \ \sigma \to \sigma \cdot \sigma, \ \ \sigma \to \sigma : \sigma, \ \ \sigma \to (\sigma), \ \ \sigma \to a + b,$$

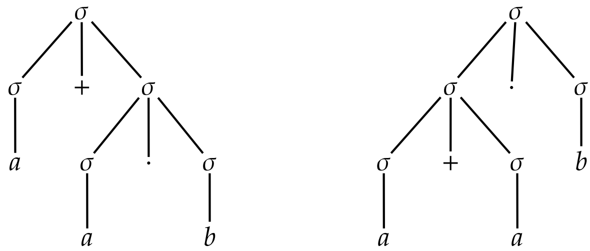
dvoznačna gramatika.



Slika 3.2 Drugo parsirajuće stablo za Zadatak 3.146.

(b) *Naći ekvivalentnu jednoznačnu gramatiku jezika $L(G)$.*

Rešenje: (a) Uzmimo proizvoljan, jednostavan aritmetički izraz $a + a \cdot b$. Pokazaćemo da on ima dva parsirajuća stabla:



Ako vrednost izraza čitamo prema redosledu dobijenom u parsirajućem stablu, u prvom stablu dobijamo vrednost $a + (a \cdot b) = a + ab$, dok je drugom dobijena vrednost $(a + a) \cdot b = 2a + b$, što znači da su dobijene vrednosti različite.

(b) Obično dvoznačnost proizilazi iz paralelne zamene pomoćnog simbola u rečeničnoj formi. Ovde σ možemo zameniti sa $\sigma + \sigma$ ili $\sigma \cdot \sigma$ i iz obe rečenične forme može da se izvede $\sigma + \sigma \cdot \sigma$. Obično se ovaj problem prevazilazi korišćenjem različitih pomoćnih simbola za predstavljanje operanda različitih operatora.

Zato ćemo uvesti nove pomoćne simbole α za predstavljanje operanda za operacije "+ i -", i β za predstavljanje operanada za "· i :" . Pri formiranju pravila izvođenja vodićemo računa o uobičajenim pravilima koja važe za aritmetičke izraze:

Operacije "$\cdot$ i :" imaju prioritet nad operacijama "+ i -";
Prilikom primene operacija istog prioriteta operacija koja je levo ima prioritet nad onom koja je sa desne strane u aritmetičkom izrazu.

Uzimajući u obzir ova pravila možemo konstruisati sledeća pravila izvođenja:

$$\sigma \to \sigma + \alpha, \ \ \sigma \to \sigma - \alpha, \ \ \sigma \to \alpha,$$
$$\alpha \to \alpha \cdot \beta, \ \ \alpha \to \alpha : \beta, \ \ \alpha \to \beta,$$
$$\beta \to (\sigma), \ \ \beta \to a, \ \ \beta \to b.$$

Na ovaj način definisali smo jednoznačnu kontekstno-nezavisnu gramatiku ekvivalentnu datoj gramatici. □

Zadatak 3.148. * *Neka je* $L = \{u \in \{a,b\}^* \,|\, |u|_a = |u|_b\}$.

a) *Konstruisati kontekstno-nezavisnu gramatiku koja generiše dati jezik.*
b) *Naći jednoznačnu kontekstno-nezavisnu gramatiku koja generiše dati jezik.*

Rešenje: a) Jednostavno se pokazuje da gramatika G, sa pravilima izvođenja

$$\sigma \to a\sigma b\sigma, \ \ \sigma \to b\sigma a\sigma, \ \ \sigma \to e,$$

generiše dati jezik, tj. da je $L = L(G,\sigma)$. Ova gramatika nije jednoznačna, jer se σ pojavljuje dva puta u rečeničnim formama $a\sigma b\sigma$ i $b\sigma a\sigma$.

Postoje, na primer, dva leva izvođenja reči $abab$ i to:

$$\sigma \Rightarrow a\sigma b\sigma \Rightarrow ab\sigma a\sigma b\sigma \Rightarrow aba\sigma b\sigma \Rightarrow abab\sigma \Rightarrow abab \quad \text{i}$$
$$\sigma \Rightarrow a\sigma b\sigma \Rightarrow ab\sigma \Rightarrow aba\sigma b\sigma \Rightarrow abab\sigma \Rightarrow abab.$$

b) Označimo sa $c(u) = |u|_a - |u|_b$, za proizvoljnu reč $u \in \{a,b\}^*$. Definisaćemo nove pomoćne simbole α i β.

Želimo da α generiše skup

$$A = \{u \in \{a,b\}^* \,|\, c(u) = 0, \text{ za svaki prefiks } v \text{ od } u \text{ je } c(v) \geqslant 0\},$$

a da neterminalni simbol β generiše skup

$$B = \{u \in \{a,b\}^* \,|\, c(u) = 0, \text{ za svaki prefiks } v \text{ od } u \text{ je } c(v) \leqslant 0\}.$$

Konstruisaćemo jednoznačnu kontekstno-nezavisnu gramatiku G_α za skup A sa pravilima $\alpha \to a\alpha b\alpha + e$ i pokazaćemo da je $A = L(G_\alpha, \alpha)$.

Jasno je da, za proizvoljnu reč $u \in L(G_\alpha, \alpha)$, važi da je $c(u) = 0$. Indukcijom po dužini reči u pokazaćemo da, za svaki prefiks v od u, važi $c(v) \geqslant 0$.

Za praznu reč $u = e$ dužine $|u| = 0$ ovo tvrđenje je trivijalno. Neka su date reči $u,v \in L(G_\alpha, \alpha)$ dužine $|u|,|v| > 0$. Tada postoje izvođenja $\alpha \overset{*}{\Rightarrow} u$ i $\alpha \overset{*}{\Rightarrow} v$ i pretpostavimo da je $c(w_1) \geqslant 0$, za svaki prefiks w_1 od u i $c(w_2) \geqslant 0$, za svaki prefiks w_2 od v. U izvođenju

$$\alpha \Rightarrow a\alpha b\alpha \stackrel{*}{\Rightarrow} aub\alpha \stackrel{*}{\Rightarrow} aubv,$$

za prefiks p reči au je, prema indukcijskoj pretpostavci, jeste vrednost $c(p) \geqslant 1$ i $c(aub) = 0$. Tako, za proizvoljan prefiks q od v, imamo $c(aubv) = c(v) \geqslant 0$. Dakle, $L(G_\alpha, \alpha) \subseteq A$.

Obratnu inkluziju dokazaćemo indukcijom po dužini reči w.

Za $w = e$ dokaz je trivijalan. Posmatrajmo proizvoljnu, nepraznu reč w. Očito je da ona počinje slovom a i završava sa b. Pretpostavimo da tvrđenje važi za sve reči dužine manje od $|w|$. Zapišimo reč w u obliku $w = aubv$, pri čemu je au najduži prefiks od w koji ima sledeću osobinu:

P_1: Svaki neprazan prefiks p od au, uključujući i au, ima $c(p) > 0$.

Za proizvoljan prefiks q od u važi $c(q) = c(aq) - 1 \geqslant 0$ i prema definiciji au imamo da je $c(aub) = 0$, tj. $c(u) = 0$, što znači da $u \in A$. Dobijamo, takođe, da je $c(v) = 0$ i za svaki prefiks r od v važi $c(r) = c(aubr) \geqslant 0$. Dakle, $u, v \in A$, pa prema indukcijskoj pretpostavci $u, v \in L(G_\alpha, \alpha)$. Kako postoje izvođenja $\alpha \stackrel{*}{\Rightarrow} u$ i $\alpha \stackrel{*}{\Rightarrow} v$ imamo

$$\alpha \Rightarrow a\alpha b\alpha \stackrel{*}{\Rightarrow} aub\alpha \stackrel{*}{\Rightarrow} aubv = w. \tag{3.18}$$

Ovim smo dokazali da je $A = L(G_\alpha, \alpha)$.

Ostalo je da dokažemo jednoznačnost gramatke G_α. Pretpostavimo da se w može dobiti iz α izvođenjem (3.18.). Tada je au najduži prefiks od w koji ima osobinu P_1, jer uslov $c(u) = 0$ povlači da je $c(aub) = 0$ i, za svaki prefiks p od u iz $c(p) \geqslant 0$ sledi $c(ap) \geqslant 1$. Tako dolazimo do zahteva da je prvo slovo a u w jednoznačno upareno sa b koje je odmah iza u. To sledi iz jedinstvenosti polaznog argumenta u parsirajućem stablu izvođenja reči w iz α u G_α.

Analogno, definišemo gramatiku G_β sa pravilima izvođenja $\beta \to b\beta a\beta + e$ i pokazujemo da je $B = L(G_\beta, \beta)$.

Sada smo spremni da definišemo jednoznačnu konteksno nezavisnu gramatiku G_1 u kojoj su pravila izvođenja:

$$\sigma \to a\alpha b\sigma + b\beta a\sigma + e, \;\; \alpha \to a\alpha b\alpha + e, \;\; \beta \to b\beta a\beta + e.$$

Očito je da je $L(G_1) \subseteq L$.

Obratnu inkluziju, odnosno tvrđenje da ako je $w \in L$ onda w ima jedinstveno parsirajuće stablo u G_1, dokazujemo indukcijom po dužini reči w. Tvrđenje važi za $w = e$. Pretpostavimo da reč $w \in \{a, b\}^+$ počinje slovom a. Zapišimo w u obliku $w = aubv$, gde je au najduži prefiks od w koji ima osobinu P_1. Na isti način kao u prethodnom delu zaključujemo da postoji jedinstveno parsirajuće stablo izvođenja $\alpha \stackrel{*}{\Rightarrow} u$, pri čemu je $c(u) = c(v) = 0$. Prema indukcijskoj pretpostavci i izvođenje $\alpha \stackrel{*}{\Rightarrow} v$ ima jedinstveno parsirajuće stablo. Dakle, postoji izvođenje:

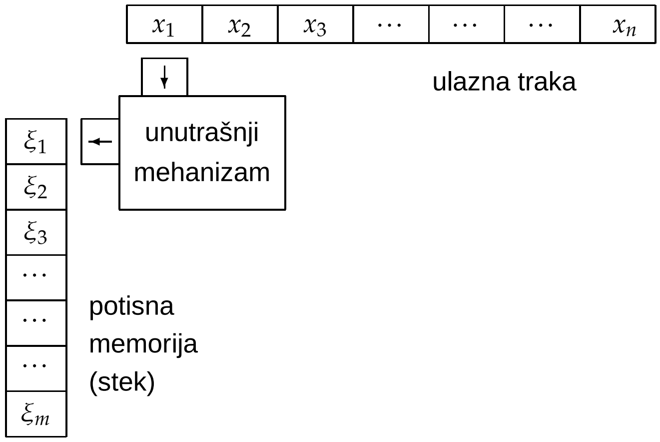
$$\alpha \Rightarrow a\alpha b\sigma \stackrel{*}{\Rightarrow} aub\sigma \stackrel{*}{\Rightarrow} aubv = w.$$

Jasno je da ovo izvođenje ima jedinstveno parsirajuće stablo, a na osnovu prvog koraka vidimo da slovo a treba upariti sa slovom b koje se nalazi iza u. Pokazaćemo da a ne može biti upareno ni sa jednim b u reči u. Ako je $u = u_1bu_2$, za neke u_1, u_2, onda, prema P_1, važi $c(u_1) > 0$. To znači da je $c(au_1b) > 0$, pa u_1 ne možemo izvesti iz α. Dalje, pokazujemo da a ne može biti upareno ni sa jednim b iz v. Za $v = v_1bv_2$ imamo $c(ub) = c(aub) - 1 = -1$, što znači da ubv_1 ima prefiks sa negativnom vrednošću c. Dakle, $ubv_1 \notin A$, pa ne može biti izvedeno iz α. Na isti način izvodimo dokaz i kada w počinje slovom b. Dakle, G_1 je jednoznačna gramatika koja generiše jezik L. $\square$

3.5. Potisni automati

Pre nego što damo formalnu matematičku definiciju pojma potisnog automata, objasnićemo šta se zamišlja pod realnim modelom potisnog automata.

Realni model potisnog automata prikazan je na sledećoj slici:



Kao što se vidi sa slike, potisni automat se u ovom modelu sastoji od ulazne trake, potisne memorije i unutrašnjeg mehanizma.

Na ulaznoj traci upisana je ulazna reč $x_1x_2\cdots x_n$, gde su $x_1, x_2, \ldots, x_n$ slova ulaznog alfabeta X. Ta reč se čita slovo po slovo, glavom za čitanje označenom na slici simbolom $\downarrow$. Posle učitavanja svakog slova, ono se briše, a ostatak reči se pomera za jedno mesto ulevo, tako da je potisni automat spreman da učita sledeće slovo ulazne reči.

U potisnoj memoriji upisana je reč $\xi_1\xi_2\cdots\xi_m$, gde su $\xi_1, \xi_2, \ldots, \xi_m$ slova alfabeta memorije M. Glava za upisivanje, označana na slici sa $\leftarrow$, deluje samo na znak na vrhu memorije, u ovom slučaju na znak ξ_1. Taj znak može se zameniti proizvoljnom reči $w = \alpha_1\alpha_2\cdots\alpha_k \in M^*$. Reč w se u memoriju upisuje počev od poslednjeg slova α_k, do prvog slova α_1, pri čemu upisivanje svakog narednog slova potiskuje ostatak sadržaja memorije za jedno mesto naniže.

Zbog toga se ovakva memorija naziva *potisna memorija* ili *stek*, a ovakav automat i zove *potisnim automatom* ili *automatom sa potiskujućom memorijom*. Zamena simbola ξ_1 praznom reči znači prosto brisanje tog simbola, pri čemu se ostatak sadržaja memorije podiže za jedno mesto naviše. Do proizvoljnog slova sadržanog u memoriji se može doći samo brisanjem svih slova koja se nalaze iznad njega.

Na kraju, unutrašnji mehanizam se karakteriše unutrašnjim stanjima potisnog automata, i u toku rada tog automata vrši se permanentna promena stanja.

Prema tome, možemo reći da se u toku rada, potisni automat u svakom trenutku karakteriše trojkom koju čine stanje u kome se trenutno nalazi, trenutni sadržaj memorije i trenutni sadržaj ulazne trake. Ovu trojku nazivamo konfiguracijom tog automata. Sada ćemo preći na formalnu definiciju potisnog automata. Pod *potisnim automatom* $\mathcal{A}$ podrazumevamo sedmorku

$$\mathcal{A} = (A, X, M, a_0, \xi_0, \mu, \delta),$$

gde je

A – konačan skup, *skup stanja* od $\mathcal{A}$;

X – neprazan konačan skup, *ulazni alfabet* od $\mathcal{A}$;

M – neprazan konačan skup, *alfabet steka* (*potisne memorije*) od $\mathcal{A}$;

a_0 – element iz A, *početno stanje* od $\mathcal{A}$;

ξ_0 – element iz M, *početni simbol steka* od $\mathcal{A}$;

μ – preslikavanje iz $A\times M$ u $\mathcal{F}(A\times M^*)$, gde $\mathcal{F}(A\times M^*)$ označava skup svih konačnih podskupova od $A\times M^*$, uključujući tu i prazan podskup;

δ – preslikavanje iz $A\times M\times(A\cup\{e\})$ u $\mathcal{F}(A\times M^*)$.

Preslikavanja μ i δ nazivamo *funkcijama prelaza* potisnog automata $\mathcal{A}$. Kako su slike pri ovim preslikavanjima podskupovi od $A\times M^*$, to je jasno da se potisni automati "ponašaju *nedeterministički*", što će se bolje videti iz daljeg teksta.

Proizvoljnu trojku $(a,\alpha,u)\in A\times M^*\times X^*$ nazivaćemo *konfiguracijom* potisnog automata $\mathcal{A}$. U napred datom modelu potisnog automata, ova konfiguracija bi odgovarala trenutku u kome se potisni automat $\mathcal{A}$ nalazi u stanju a, u steku se nalazi zapamćena reč α, a na ulaznoj traci na očitavanje čeka ulazna reč u. Rad potisnog automata $\mathcal{A}$ sastoji se u nizu prelazaka iz konfiguracije u konfiguraciju, koji su određeni funkcijama prelaza μ i δ. Te funkcije se takođe mogu shvatiti kao izvesni *program*, ili, tačnije rečeno, *skup instrukcija* po kome radi potisni automat $\mathcal{A}$. Taj skup instrukcija čine sve relacije oblika

$$(b,\beta)\in\mu(a,\xi)$$

koje važe za $(b,\beta)\in A\times M^*$ i $(a,\xi)\in A\times M$, i sve relacije oblika

$$(b,\beta)\in\delta(a,\xi,x)$$

koje važe za $(b,\beta) \in A \times M^*$ i $(a,\xi,x) \in A \times M \times (X \cup \{e\})$.

Možemo razlikovati tri vrste prelaza u potisnom automatu:

Stacionarni prelazi:

Neka je data konfiguracija (a,α,u), pri čemu je $\alpha \in M^+$, što znači da je stek neprazan. Označimo sa ξ prvo slovo reči α, tj. uzmimo da je $\alpha = \xi\alpha'$, za neki $\alpha' \in M^*$. Ako je $(b,\beta) \in \mu(a,\xi)$, tada je moguć prelaz sa konfiguracije (a,α,u) na konfiguraciju $(b,\beta\alpha',u)$, što simbolički označavamo

$$(a,\alpha,u) \to (b,\beta\alpha',u).$$

Drugim rečima, relaciju

$$(b,\beta) \in \mu(a,\xi),$$

možemo smatrati instrukcijom programa potisnog automata $\mathcal{A}$ koja kaže

> *pređi iz stanja a u b, zameni prvi simbol steka ξ rečju β*
> *i ostavi ulaznu reč u nepromenjenom.*

Dakle, prelazi ovog tipa ne zavise od ulaza. Ovakve prelaze nazivamo *stacionarnim prelazima*, jer se prilikom njih ne menja sadrzaj ulazne trake.

Progresivni prelazi:

Neka je data konfiguracija (a,α,u), pri čemu je $\alpha \in M^+$ i $u \in X^+$, što znači da su i stek i ulazna traka neprazni. Označimo sa ξ prvo slovo reči α, tj. uzmimo da je $\alpha = \xi\alpha'$, za neki $\alpha' \in M^*$, i označimo sa x prvo slovo ulazne reči u, tj. uzmimo da je $u = xu'$, za neki $u' \in X^*$. Ako je $(b,\beta) \in \delta(a,\xi,x)$, tada je moguć prelaz sa konfiguracije (a,α,u) na konfiguraciju $(b,\beta\alpha',u')$, što simbolički označavamo

$$(a,\alpha,u) \to (b,\beta\alpha',u').$$

Drugim rečima, relaciju

$$(b,\beta) \in \delta(a,\xi,x)$$

možemo smatrati instrukcijom programa potisnog automata $\mathcal{A}$ koja kaže

> *pređi iz stanja a u b, zameni prvi simbol steka ξ rečju β*
> *i izbriši prvo slovo x ulazne reči u.*

Prelaze ovakvog tipa nazivamo *progresivnim prelazima*, jer se prilikom njih briše prvi simbol sa ulazne trake.

Prelazi sa praznim ulazom:

Neka je data konfiguracija (a,α,e), pri čemu je $\alpha \in M^+$. Označimo sa ξ prvo slovo reči α, tj. uzmimo da je $\alpha = \xi\alpha'$, za neki $\alpha' \in M^*$. Ako je $(b,\beta) \in \delta(a,\xi,e)$, tada je moguć prelaz sa konfiguracije (a,α,u) na konfiguraciju $(b,\beta\alpha',e)$, što simbolički označavamo

$$(a,\alpha,e) \to (b,\beta\alpha',e).$$

Drugim rečima, relaciju

$$(b,\beta) \in \delta(a,\xi,e)$$

možemo smatrati instrukcijom programa potisnog automata $\mathcal{A}$ koja kaže

> *pređi iz stanja a u b, zameni prvi simbol steka ξ rečju β*
> *i ostavi ulaznu traku praznom*

(kakva je bila i pre prelaza). Ovakve prelaze ćemo nazivati *prelazima sa praznim ulazom*.

Možemo definisati relaciju $\to$ na skupu svih konfiguracija potisnog automata A, pri čemu je

$$(a_1,\alpha_1,u_1) \to (a_2,\alpha_2,u_2)$$

ako i samo ako je moguć prelaz iz konfiguracije (a_1,α_1,u_1) u konfiguraciju (a_2,α_2,u_2), koji je bilo stacionaran, bilo progresivan, bilo prelaz sa praznim ulazom.

Sa $\xrightarrow{*}$ ćemo označavati refleksivno-tranzitivno zatvorenje relacije $\to$. To znači da za dve konfiguracije (a,α,u) i (b,β,v) potisnog automata $\mathcal{A}$ važi

$$(a,\alpha,u) \xrightarrow{*} (b,\beta,v) \tag{3.19}$$

ako i samo ako je ili $(a,\alpha,u) = (b,\beta,v)$, ili postoji niz konfiguracija

$$\Big\{(a_i,\alpha_i,u_i)\Big\}_{i=1}^{n}, \quad n \in \mathbb{N},$$

takav da važi

$$(a,\alpha,u) = (a_1,\alpha_1,u_1) \to (a_2,\alpha_2,u_2) \to \cdots \to (a_n,\alpha_n,u_n) = (b,\beta,v). \tag{3.20}$$

Izraz oblika (3.19), odnosno niz prelaza (3.20), nazivamo *izračunavanjem* u $\mathcal{A}$, a pojedinačne prelaze u (3.20) nazivamo *koracima* u tom izračunavanju.

Primetimo da sa konfiguracije oblika (a,e,u) nije moguće preći ni na jednu drugu konfiguraciju potisnog automata $\mathcal{A}$. To znači da ako u nekom izračunavanju u $\mathcal{A}$ dođemo do konfiguracije tog oblika, izračunavanje se prekida. Drugim rečima, ukoliko se u toku rada potisnog automata stek isprazni, što se može desiti, jer prvi simbol steka možemo zameniti i praznom reči, što znači obrisati, tada se rad potisnog automata prekida.

Raspoznavanje jezika potisnim automatima

Pojam raspoznavanja, odnosno generisanja jezika potisnim automatom može se definisati na dva načina – raspoznavanje skupom stanja i raspoznavanje praznim stekom, i pokazuje se da su ta dva načina ekvivalentna.

Potisni automat $\mathcal{A}$ *raspoznaje jezik $L \subseteq X^*$ skupom $T \subseteq A$* ako je

$$L = \Big\{u \in X^* \,\Big|\, (\exists a \in T)(\exists \alpha \in M^*)\ (a_0,\xi_0,u) \xrightarrow{*} (a,\alpha,e)\Big\},$$

odnosno ako za $u \in X^*$ važi

$$u \in L \iff (\exists a \in T)(\exists \alpha \in M^*)\ (a_0,\xi_0,u) \xrightarrow{*} (a,\alpha,e).$$

Drugi način raspoznavanja jezika potisnim automatom je raspoznavanje praznim stekom. Naime, potisni automat $\mathcal{A}$ *raspoznaje jezik L praznim stekom* ako je

$$L = \Big\{u \in X^* \,\Big|\, (\exists a \in A)\ (a_0,\xi_0,u) \xrightarrow{*} (a,e,e)\Big\},$$

odnosno ako za $u \in X^*$ važi

$$u \in L \iff (\exists a \in A)\ (a_0,\xi_0,u) \xrightarrow{*} (a,e,e).$$

Teorema 3.18. *Neka je $L \subseteq X^*$ jezik nad konačnim alfabetom X. Tada postoji potisni automat koji raspoznaje L skupom stanja ako i samo ako postoji potisni automat koji raspoznaje L praznim stekom. Dokazati.*

Za jezik $L \subseteq X^*$ nad konačnim alfabetom X i potisni automat $\mathcal{A}$ koji raspoznaje L bilo nekim podskupom skupa stanja, bilo praznim stekom, kažemo da *A raspoznaje L* i da je L jezik *raspoznatljiv potisnim automatom*.

Zadatak 3.149. *Posmatrajmo potisni automat $\mathcal{A} = (A,X,M,q,e,\delta)$, sa skupom stanja $A = \{p,q\}$, ulaznim alfabetom $X = \{a,b,c\}$, alfabetom steka $M = \{a,b\}$, početnim simbolom steka $\xi_0 = e$ i progresivnim prelazima definisanom na sledeći način:*

$$\begin{array}{lll} \delta(q,\xi,a) = \{(q,a\xi)\}, & \delta(q,\xi,b) = \{(q,b\xi)\}, & \delta(q,\xi,c) = \{(p,\xi)\}, \\ \delta(p,a,a) = \{(p,e)\}, & \delta(p,b,b) = \{(p,e)\}, & \textit{za simbol } \xi \in M \cup \{\xi_0\} \textit{ sa vrha steka.} \end{array}$$

Odrediti jezik $L(\mathcal{A})$ koji raspoznaje ovaj potisni automat i naći izračunavanja reči abcba, abcb i abcbab. Koja od ovih reči pripada jeziku $L(\mathcal{A})$?

Rešenje: Primetimo da ovaj potisni automat radi na sledeći način: U inicijalnom stanju q on smešta slova a i b sa ulazne trake u stek, sve dok na ulaznoj traci ne pročita slovo c. Posle toga, unutrašnji mehanizam prelazi u stanje p, u kome $\mathcal{A}$ upoređuje svaki simbol sa ulazne trake sa simbolom na vrhu steka. Ako su oni jednaki, automat prihvata ulaz.

Date progresivne prelaze možemo, formalno, predstaviti na sledeći način:

$$\begin{array}{ll} (q,\alpha,au') \to (q,a\alpha,u'), & (q,\alpha,bu') \to (q,b\alpha,u'), \\ (p,a\alpha',au') \to (p,\alpha',u'), & (p,b\alpha',bu') \to (p,\alpha',u'), \\ (q,\alpha,cu') \to (p,\alpha,u'), & \text{za } \alpha \in,\ \alpha' \in M^*,\ u' \in X^*. \end{array}$$

Neka je reč $u \in \{a,b\}^*$ prefiks na ulaznoj traci, pre prvog pojavljivanja reči c i v sufiks posle prvog pojavljivanja c na ulaznoj traci. U trenutku kada potisni automat pređe u stanje p, reč u je smeštena u stek u obratnom redosledu, tj. pri poređenju sufiksa v sa sadržajem steka, prvo slovo iz v poredi se sa poslednjim slovom reči u i tako dalje. To znači da je $v = \overline{u}$, pa je

$$L(\mathcal{A}) = \{uc\overline{u} \,|\, u \in \{a,b\}^*\}.$$

Kako je $\xi_0 = e$ početni simbol steka, tražene reči mogu se dobiti sledećim izračunavanjima:

$$(q,e,abcba) \to (q,a,bcba) \to (q,ba,cba) \to (p,ba,ba) \to (p,a,a) \to (p,e,e);$$
$$(q,e,abcb) \to (q,a,bcb) \to (q,ba,cb) \to (p,ba,b) \to (p,a,e);$$
$$(q,e,abcbab) \to (q,a,bcbab) \to (q,ba,cbab) \to (p,ba,bab) \to (p,a,ab) \to (p,e,b).$$

Primećujemo da $\mathcal{A}$ raspoznaje reč $abcba$ praznim stekom, dok se druga i treća reč ne raspoznaju ovim potisnim automatom. □

Zadatak 3.150. *Konstruisati potisni automat koji raspoznaje jezik*

$$L = \{a^i b^j c^k \,|\, i,j,k \geqslant 0,\ i+k=j\}.$$

Rešenje: Koristićemo tri različita stanja potisnog automata da bi obezbedili redosled čitanja slova a, b i c i stek da bi ispunili zahtev da je $i+k=j$. Dakle, konstruisaćemo automat $\mathcal{A}$ sa skupom stanja $A = \{p,q,r\}$, gde će p biti inicijalno, a r finalno stanje. U stanju p, automat svako slovo a sa ulazne trake smešta u stek. Pri prvom učitavanju slova b prelazi u stanje q i svako b sa ulazne trake ili uparuje sa slovom a sa vrha steka ili ga smešta u stek. Pri prvom čitanju slova c sa ulazne trake automat prelazi u stanje r i uparuje ga sa b iz steka, kao i svako sledeće c dok se stek ne isprazni.

Dakle, traženi potisni automat je $\mathcal{A} = (A,X,M,p,\xi_0,\delta,T)$, sa skupom stanja $A = \{p,q,r\}$, ulaznim alfabetom $X = \{a,b,c\}$, alfabetom steka $M = \{a,b\}$, inicijalnim stanjem p , početnim simbolom steka ξ_0 i skupom finalnih stanja $T = \{r\}$. U automatu $\mathcal{A}$ ćemo definisati progresivne prelaze na sledeći način:

$$\delta(p,\xi,a) = \{(p,a\xi)\},\quad \delta(p,a,b) = \{(q,e)\},\quad \text{za } \xi \in \{\xi_0,a\}$$
$$\delta(q,\xi,b) = \{(q,b\xi)\},\quad \delta(q,b,c) = \{(r,e)\},\quad \text{za } \xi \in \{\xi_0,b\}$$
$$\delta(r,b,c) = \{(r,e)\}.$$

Jasno je da, ovako definisan potisni automat, raspoznaje jezik L praznim stekom. □

Zadatak 3.151. *Konstruisati potisni automat koji raspoznaje jezik*

$$L = \{x^i y^j \,|\, 2i \neq 3j\}.$$

Rešenje: Posmatrajmo najpre jezik $L_1 = \{a^i b^j \,|\, i,j \geqslant 0, 2i = 3j\}$. Primećujemo da reč $x^i y^j$ pripada jeziku L_1 samo ako je $i = 3k$ i $j = 2k$, za neki $k \geqslant 0$. Za svako slovo x na ulaznoj traci možemo smestiti u stek dve njegove kopije, korišćenjem pomoćnog stanja i stacionarnih prelaza. Slično, svako ulazno slovo y možemo upariti sa slovom x na vrhu steka, a zatim pomoću još dva pomoćna stanja brišemo još dva slova x, za isti ulaz y. Konstruišimo potisni automat $\mathcal{A}_1 = (A_1, X, M_1, a, \xi_0^1, \delta_1, \mu_1, T_1)$ koristeći ovu ideju. Neka je skup stanja ovog automata $A_1 = \{a, a', b, b', b''\}$, ulazni alfabet je $X = \{x, y\}$, alfabet steka $M_1 = \{x\}$, ξ_0^1 početni simbol steka, $T_1 = \{b\}$ finalno stanje. Progresivne prelaze definisaćemo sa:

$$\delta(a, \xi_0^1, e) = (b, e, e), \quad \delta(b'', x, y) = (b, e),$$
$$\delta(a', \xi, x) = (a, x\xi), \quad \text{za } \xi \in \{\xi_0^1, x\},$$

a stacionarne prelaze na sledeći način:

$$(a, \alpha, xu') \to (a', x\alpha, xu'), \quad \text{tj. } \mu(a, \xi) = (a', x\xi),$$
$$\text{za } \alpha = \xi\alpha',\ \xi \in \{\xi_0^1, x\},\ \alpha' \in \{x\}^*,\ u' \in X^*$$
$$(a, \alpha, yu') \to (b', \alpha', yu'), \quad \text{tj. } \mu(a, x) = (b', e),\ \text{za } \alpha = x\alpha',\ \alpha' \in \{x\}^*,\ u' \in X^*,$$
$$(b', \alpha, yu') \to (b'', \alpha', y), \quad \text{tj. } \mu(b', x) = (b'', e),\ \text{za } \alpha = x\alpha',\ \alpha' \in \{x\}^*,\ u' \in X^*,$$
$$(b, \alpha, yu') \to (b', \alpha', yu'), \quad \text{tj. } \mu(b, x) = (b', e),\ \text{za } \alpha = x\alpha',\ \alpha' \in \{x\}^*,\ u' \in X^*.$$

Potisni automat $\mathcal{A}_1$ raspoznaje jezik L_1 praznim stekom.

Vratimo se jeziku L. Iskoristićemo prethodnu ideju da svako slovo y uparimo sa jednim i još pola x. Posle toga ćemo imati dva završna stanja a_x, ako je $2i > 3j$ i b_y, ako je $2i < 3j$. Da bi stigli do stanja a_x, neophodno je da iz steka brišemo bar jedno slovo x koje nije vezano za ulaz y. Da bi stigli do b_y treba da učitamo najmanje jedno slovo y više. U ovom drugom slučaju može se javiti problem, jer automat može da blokira u stanjima b' i b'' pokušavajući da izvuče x iz praznog steka. Primetimo da, ako je u stanju b' ili b'' stek prazan, važi $2i < 3j$, pa možemo doći do stanja b_y, direktno iz a, odnosno b', bez brisanja simbola x iz steka. Konstruišimo sada automat $\mathcal{A} = (A, X, M, a, \xi_0, \delta, \mu, T)$ na sledeći način: Neka je skup stanja ovog automata $A = \{a, a', a_x, b, b', b'', b_y\}$, ulazni alfabet je $X = \{x, y\}$, alfabet steka $M = \{x\}$, ξ_0 početni simbol steka i $T = \{a_x, b_y\}$ skup finalnih stanja. Prelaze, na osnovu prethodnog, možemo definisati na sledeći način:

$$\delta(a, \xi, e) = (b, e), \quad \text{za } \xi \in \{\xi_0, x\},$$
$$\delta(b, e, x) = (a_x, e), \quad \delta(a_x, e, x) = (a_x, e),$$
$$\delta(b, e, y) = (b_y, e), \quad \delta(b_y, e, y) = (b_y, e),$$
$$(a, \alpha, xu') \to (a', x\alpha, xu'), \quad \text{tj. } \mu(a, \xi) = (a', x\xi),$$
$$\text{za } \alpha = \xi\alpha',\ \xi \in \{\xi_0, x\},\ \alpha' \in \{x\}^*,\ u' \in X^*$$
$$\delta(a', \xi, x) = (a, x\xi), \quad \text{za } \xi \in \{\xi_0, x\},$$
$$(a, \alpha, yu') \to (b', \alpha', yu'), \quad \text{tj. } \mu(a', x) = (b', e),\ \text{za } \alpha = x\alpha',\ \alpha' \in \{x\}^+,\ u' \in X^*,$$
$$(a, x, yu') \to (b_y, e, u'), \quad \text{tj. } \delta(a, x, y) = (b_y, e),\ \text{za } u' \in X^*$$
$$(b', \alpha, yu') \to (b'', \alpha', yu'), \quad \text{tj. } \mu(b', x) = (b'', e),\ \text{za } \alpha = x\alpha',\ \alpha' \in \{x\}^+,\ u' \in X^*,$$
$$(b', x, yu') \to (b_y, e, u'), \quad \text{tj. } \delta(b', x, y) = (b_y, e),\ \text{za } u' \in X^*,$$
$$\delta(b'', x, y) = (b, e),$$

$$(b,\alpha,yu') \to (b',\alpha',yu'), \quad \text{tj. } \mu(b,x) = (b',e), \text{ za } \alpha = x\alpha',\ \alpha' \in \{x\}^+ \text{ i } u' \in X^*,$$
$$(b,x,yu') \to (b_y,e,u'), \quad \text{tj. } \delta(b,x,y) = (b_y,e), \text{ za } u' \in X^*.$$

Nije teško pokazati da ovako definisan potisni automat raspoznaje traženi jezik L. □

Zadatak 3.152. *Konstruisati potisni automat koji raspoznaje jezik*

$$L = \{u \in \{x,y\}^* \,|\, 2|u|_x \neq 3|u|_y\}$$

Rešenje: Primetimo, ovde, da za svako slovo x sa ulaza u stek treba smestiti dve kopije simbola x, dok za svako ulazno slovo y u stek smeštamo tri kopije simbola y. Svaki stek simbol x treba uparitisa stek simbolom y. Uparivanje vršimo ako je novo ulazno slovo različito od simbola sa vrha steka, dok u protivnom slovo smeštamo u stek. Treba voditi računa da, ako u steku nema dovoljno simbola sa kojima bi se uparivanje vršilo, ostale kopije ulaznog slova smeštamo u stek.

Neka je a inicijalno stanje traženog automata. Kada je na ulaznoj traci učitano slovo x ili ga uparujemo sa simbolom y iz steka ili ga smeštamo u stek, kada automat prelazi u stanje a_1. U stanju a_1 ili brišemo iz steka drugi simbol y ili stavljamo drugu kopiju simbola x u stek i vraćamo se u stanje a. Stanja b_1 i b_2 koristićemo za rad sa simbolom y na sličan način.

Formirajmo sada potisni automat $\mathcal{A} = (A,X,M,a,\xi_0,\delta,\mu,T)$. Neka je skup $A = \{a,a_1,b_1,b_2,a_x,b_y\}$, $X = \{x,y\}$, $M = \{x,y\}$, $T = \{a_x,b_y\}$, a prelaze ćemo definisati na sledeći način:

$$(a,\xi\alpha',xu') \to (a_1,x\alpha,xu'), \ \text{tj. } \mu(a,\xi) = (a_1,x\xi),$$
$$\text{za } \alpha = \xi\alpha',\ \xi \in \{\xi_0,x\},\ \alpha' \in \{x,y\}^*,\ u' \in X^*$$
$$(a,y\alpha',xu') \to (a_1,\alpha',xu'), \ \text{tj. } \mu(a,x) = (a_1,e),\ u' \in X^*$$
$$\delta(a_1,\xi,x) = (a,x\xi), \qquad \text{za } \xi \in \{\xi_0,x\},$$
$$\delta(a_1,y,x) = (a,e),$$
$$\delta(a,x,e) = (a_x,e),$$
$$\delta(a_x,x,x) = \delta(a_x,x,e) = (a_x,e),$$
$$\delta(a,y,e) = (b_y,e),$$
$$\delta(b_y,y,y) = \delta(b_y,y,e) = (b_y,e),$$
$$(a,\xi\alpha',yu') \to (b_1,y\alpha,yu'), \ \text{tj. } \mu(a,\xi) = (b_1,y\xi),$$
$$\text{za } \alpha = \xi\alpha',\ \xi \in \{\xi_0,y\},\ \alpha' \in \{x,y\}^*,\ u' \in X^*,$$
$$(a,x\alpha',yu') \to (b_1,\alpha',yu'), \ \text{tj. } \mu(a,y) = (b_1,e),\ u' \in X^*,$$
$$(b_1,\xi\alpha',yu') \to (b_2,y\alpha,yu'), \text{tj. } \mu(a,\xi) = (b_1,y\xi),$$
$$\text{za } \alpha = \xi\alpha',\ \xi \in \{\xi_0,y\},\ \alpha' \in \{x,y\}^*,\ u' \in X^*$$
$$(b_1,x\alpha',yu') \to (b_2,\alpha',yu'), \ \text{tj. } \mu(b_1,y) = (b_2,e),\ u' \in X^*,$$

$$\begin{aligned}
\delta(b_2,\xi,y) &= (a,y\xi), && \text{za } \xi \in \{\xi_0,y\},\\
\delta(b_2,y,x) &= (a,e).
\end{aligned}$$

Da bi pokazali da ovaj potisni automat raspoznaje upravo jezik L, primetimo da, prema prethodnim pravilima, stek uvek sadrži najviše jedan stek simbol. Ako $u \in L$ i ako je $2|u|_x - 3|u|_y > 0$ stižemo do završnog stanja a_x, dok za $2|u|_x - 3|u|_y < 0$ stižemo u b_y. □

Zadatak 3.153. *Konstruisati potisni automat koji raspoznaje jezik*

$$L = \{u \in \{x,y\}^* \,|\, |u|_x \leqslant |u|_y \leqslant 2|u|_x\}.$$

Rešenje: Pri konstrukciji ovog automata problem je što svako slovo x treba upariti sa najmanje jednim i najviše dva slova y. Tu se javlja problem, jer ako u stek stavimo samo jedno slovo x, ne možemo da odredimo da li je $|u|_y \leqslant 2|u|_x$. Takođe, ako u stek stavimo dva simbola xx, nema načina da proverimo da li je $|u|_x \leqslant |u|_y$. Jedini način da se to reši je, da podelimo sva pojavljivanja slova x u dve grupe: svako x iz prve grupe uparujemo sa jednim y i svako x iz druge grupe sa po dva y.

Dakle, konstrisaćemo potisni automat $\mathcal{A} = (A,X,M,a,\xi_0,\delta,\mu,T)$, sa dva stanja $A = \{a,b\}$, alfabetom $X = \{x,y\}$, skupom stek simbola $M = \{x,y\}$, i inicijalnim i završnim stanjem koja se poklapaju, tj.$T = \{a\}$, dok su prelazi definisani na sledeći način:

$$\begin{aligned}
&\delta(a,e,x) = (b,x), && \text{za } \xi \in \xi_0\\
&\delta(a,x,x) = (a,xx),\\
&\delta(a,x,y) = (a,e),\\
&\delta(a,y,x) = (b,e),\\
&\delta(a,\xi,y) = (a,y\xi), && \text{za } \xi \in \{\xi_0,y\}\\
&\mu(b,\xi) = (a,x\xi), && \text{za } \xi \in \{\xi_0,x\}\\
&\delta(b,\xi,y) = (a,e), && \text{za } \xi \in \{\xi_0,x\}.
\end{aligned}$$

Ovaj automat raspoznaje traženi jezik. □

Potisni automati i konteksno nezavisni jezici

Zadatak 3.154. *Neka je X konačan alfabet i neka je $L \subseteq X^*$ neprazan jezik. Dokazati da je L kontekstno-nezavisan ako i samo ako može biti raspoznat nekim potisnim automatom.*

Rešenje: Prikazaćemo samo glavne crte ovog dokaza. Pretpostavimo prvo da je L kontekstno-nezavisan jezik. Razlikovaćemo slučajeve $e \notin L$ i $e \in L$.

Slučaj $e \notin L$: Bez umanjenja opštosti možemo pretpostaviti da je $L = L(G,\sigma)$ za neku kontekstno-nezavisnu gramatiku $G = (V,X,\pi)$ koja je u Normalnoj formi Chomsky.

Konstruišimo potisni automat $\mathcal{A} = (A, X, M, a_0, \xi_0, \mu, \delta, T)$ na sledeći način:

$$
\begin{aligned}
&A = \{a_0, a_1, t\}, \quad M = (V \setminus X) \cup \{\xi_0\}, \text{ gde } \xi_0 \notin V \text{ i } X \cap V = \emptyset,\\
&\mu(a_0, \xi_0) = \{(a_1, \sigma\xi_0)\},\\
&\mu(a_1, \lambda) = \{(a_1, \alpha\beta) \,|\, \lambda \to \alpha\beta \text{ je pravilo iz } \pi\},\\
&\mu(a, \lambda) = \emptyset, \quad \text{u ostalim slučajevima},\\
&\delta(a_1, \xi_0, e) = \{(t, \xi_0)\},\\
&\delta(a_1, \lambda, x) = \{(a_1, e)\}, \quad \text{ako je } \lambda \to x \text{ pravilo iz } \pi,\\
&\delta(a, \lambda, x) = \emptyset, \quad \text{u ostalim slučajevima.}
\end{aligned} \tag{3.21}
$$

Tada potisni automat $\mathcal{A}$ raspoznaje L skupom $T = \{t\}$.

Slučaj $e \in L$: U ovom slučaju posmatramo jezik $L' = L \setminus \{e\}$. Tada automat A definisan za L' raspoznaje L, ako u (3.21) dodamo pravilo

$$\delta(a_0, \xi_0, e) = \{(t, \xi_0)\}.$$

Obratno, pretpostavimo da jezik L može biti raspoznat nekim potisnim automatom

$$\mathcal{A} = (A, X, M, a_0, \xi_0, \mu, \delta).$$

Prema Teoremi 3.18. možemo, bez umanjenja opštosti, uzeti da se radi o raspoznavanju praznim stekom.

Definišimo gramatiku $G = (V, X, \pi)$ na sledeći način. Uzmimo najpre da je

$$V \setminus X = (A \times M \times A) \cup \{\sigma\}.$$

Elemente iz $V \setminus X$ zapisivaćemo u obliku $[p, \alpha, q]$, da ih ne bi pomešali sa konfiguracijama potisnig automata A.

Skup π pravila definisaćemo na sledeći način. Najpre ćemo za svaki $a \in A$ staviti da je

$$\sigma \to [a_0, \xi_0, a]$$

pravilo iz π. Dalje, svakoj "instrukciji" $(a', \beta_1\beta_2 \cdots \beta_k) \in \mu(a, \alpha)$, za $k \geqslant 1$, pridružićemo skup pravila

$$[a, \alpha, a_k] \to [a', \beta_1, a_1][a_1, \beta_2, a_2] \cdots [a_{k-1}, \beta_k, a_k],$$

gde je $a_1, \ldots, a_k$ proizvoljan niz elemenata iz A, dok ćemo "instrukciji" oblika $(a', e) \in \mu(a, \alpha)$ pridružiti pravilo

$$[a, \alpha, a'] \to e.$$

Na kraju, proizvoljnoj "instrukciji" $(a', \beta_1\beta_2 \cdots \beta_k) \in \delta(a, \alpha, x)$, za $k \geqslant 1$, pridružujemo skup pravila

$$[a, \alpha, a_k] \to x[a', \beta_1, a_1][a_1, \beta_2, a_2] \cdots [a_{k-1}, \beta_k, a_k],$$

gde je $a_1,\dots,a_k$ proizvoljan niz elemenata iz A, dok ćemo "instrukciji" oblika $(a',e) \in \delta(a,\alpha,x)$ pridružiti pravilo

$$[a,\alpha,a'] \to x.$$

Kada sva ova pravila sakupimo i njima formiramo skup π, dobijamo gramatiku G koja je očigledno kontekstno-nezavisna (mada ne u Normalnoj formi Chomsky), i za koju je $L = L(G,\sigma)$. □

Zadatak 3.155. *Neka je $G = (V,\{x,y\},\pi)$ konteksno-nezavisna gramatika sa pravilima izvođenja:*

$$\sigma \to x\sigma, \;\; \sigma \to x\sigma b\sigma y\sigma, \;\; \sigma \to e.$$

Konstruisati potisni automat $\mathcal{A}$ koji raspoznaje jezik $L = L(G,\sigma)$.

Rešenje: Da bi primenili konstrukciju automata datu u rešenju prethodnog zadatka treba datu gramatiku G svesti na ekvivalentnu gramatiku u Normalnoj formi Chomsky.

Jednostavno je videti da je skup pravila izvođenja π ekvivalentan sledećem skupu pravila:

$$\sigma \to \alpha\sigma, \;\; \sigma \to \alpha\beta, \;\; \beta \to \sigma\gamma, \;\; \gamma \to \lambda\sigma,$$
$$\sigma \to x, \;\; \alpha \to x, \;\; \beta \to y, \;\; \gamma \to y, \;\; \lambda \to y.$$

Konstruišimo potisni automat $\mathcal{A} = (A,X,M,a_0,\xi_0,\mu,\delta,T)$ na sledeći način:

$A = \{a_0,a_1,t\}, \quad X = \{x,y\}, \quad M = (V \setminus X) \cup \{\xi_0\}$, gde $\xi_0 \notin V$ i $X \cap V = \emptyset$,
$\mu(a_0,\xi_0) = \{(a_1,\sigma\xi_0)\}$,
$\delta(a_0,\xi_0,e) = \{(t,\xi_0)\}$,
$\mu(a_1,\sigma) = \{(a_1,\alpha\sigma) \,|\, \sigma \to \alpha\sigma\}$,
$\mu(a_1,\sigma) = \{(a_1,\alpha\beta) \,|\, \sigma \to \alpha\beta\}$,
$\mu(a_1,\beta) = \{(a_1,\sigma\gamma) \,|\, \beta \to \sigma\gamma\}$,
$\mu(a,\mu) = \emptyset$, u ostalim slučajevima, $a \in A$, $\mu \in M$,
$\delta(a_1,\xi_0,e) = \{(t,\xi_0)\}$,
$\delta(a_1,\sigma,x) = \{(a_1,e)\}$, za pravilo $\sigma \to x$,
$\delta(a_1,\alpha,x) = \{(a_1,e)\}$, za pravilo $\alpha \to x$,
$\delta(a_1,\beta,y) = \{(a_1,e)\}$, za pravilo $\beta \to y$,
$\delta(a_1,\gamma,y) = \{(a_1,e)\}$, za pravilo $\gamma \to y$,
$\delta(a_1,\lambda,y) = \{(a_1,e)\}$, za pravilo $\lambda \to y$,
$\delta(a,\mu,x) = \emptyset$, u ostalim slučajevima, za $a \in A$, $\mu \in M$,
$\delta(a,\mu,y) = \emptyset$, u ostalim slučajevima, za $a \in A$, $\mu \in M$.

Tada potisni automat $\mathcal{A}$ raspoznaje $L = L(G,\sigma)$ skupom $T = \{t\}$. □

Zadaci za samostalni rad

1. Dokazati da postoji algoritam pomoću koga za svaku konteksno-nezavisnu gramatiku $G = (V, X, \pi)$ i svaki konačan jezik $L \subseteq X^*$ možemo odlučiti da li je $L \subseteq L(G)$ i da li je $L \cap L(G) = \emptyset$.
2. Jezik $L = \{x^m y^n x^m y^n \mid m, n \in \mathbb{N}\}$ nije kontekstno-nezavisan. Dokazati.
3. Dokazati da jezik $L = \{a^i b^j c^k \mid k = max\{i, j\}\}$ nije kontekstno-nezavisan.
4. Neka je $G = (V, X, \pi)$ konteksno nezavisna gramatika sa pravilima

$$\sigma \to a\sigma aa + \alpha, \ \alpha \to bb\alpha cc + \beta, \ \beta \to bc.$$

   (a) Konstruisati parsirajuće stablo reči $a^3b^3c^3a^6$.
   (b) Dokazati da ova gramatika nije jednoznačna.
5. Neka je $G = (V, X, \pi)$ konteksno nezavisna gramatika sa pravilima

$$\sigma \to \alpha a\sigma b\beta + e, \ \alpha \to a\alpha + a, \ \beta \to b\beta + e.$$

   (a) Dokazati da je ova gramatika jednoznačna.
   (b) Konstruisati njoj ekvivalentnu nejednoznačnu gramatiku.
6. Konstruisati potisni automat koji raspoznaje jezik

   (i) $L = \{u \in \{x, y\}^* \mid 3|u|_x \leqslant 5|u|_y \leqslant 4|u|_x\}$;
   (ii) $L = \{u \in \{x, y\}^* \mid 2|u|_x + 5 \leqslant 3|u|_y\}$;
   (iii) $L = \{a^i b^j c^k \mid i, j, k \geqslant 0, \ i + 2k = j\}$.

7. Za svaku od sledećih konteksno-nezavisnih gramatika $G = (V, X, \pi)$ konstruisati potisni automat koji raspoznaje jezik $L = L(G, \sigma)$:

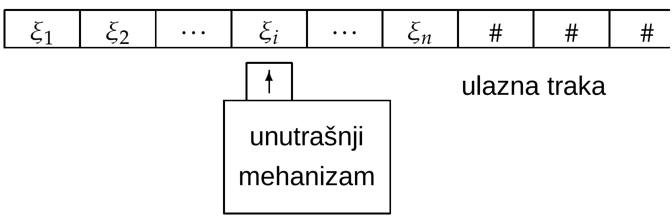
   (a) $X = \{a, b\}, \ \pi = \{\sigma \to a\sigma b\sigma, \ \sigma \to b\sigma a\sigma, \ \sigma \to e\}$,
   (b) $X = \{x, y\}, \ \pi = \{\sigma \to \sigma\sigma, \ \sigma \to x\sigma y, \ \sigma \to e\}$.

Glava 4
Turingove mašine

Postoji više različitih načina za definisanje Turingovih mašina. Međutim, svi oni su međusobno ekvivalentni, pa ćemo ovde dati osnovni model Turingove mašine. Njegovom jednostavnom modifikacijom dobijaju se i ostali modeli.

4.1. Konstrukcija Turingovih mašina

Pre nego što damo formalnu definiciju Turingove mašine opisaćemo neformalno njen rad.



Turingova mašina se sastoji iz *ulazne trake* koja je podeljena na ćelije i u svakoj od njih zabeležen je po jedan simbol iz konačnog alfabeta Σ, koji se naziva skup *simbola trake*. Ulazna traka ima početak sa leve strane, a sa desne strane je neograničena, tj. ima beskonačno mnogo ćelija. U svakom diskretnom trenutku vremena, *glava mašine* pokazuje na jednu ćeliju, odnosno na simbol trake koji je u njoj zabeležen. Na početku rada mašine, na ulaznoj traci su u prvih n ćelija, za neki prirodan broj n, zabeleženi simboli iz *ulaznog alfabeta* X, gde je $X \subseteq \Sigma$. Preostali deo trake je prazan, tako da među simbolima trake postoji i simbol # koji označava praznu ćeliju i važi $\# \notin X$. *Unutrašnji mehanizam* mašine određen je *stanjima* mašine, pri čemu je skup A svih stanja mašine konačan. Rad mašine kontrolisan je *instrukcijama* koje u svakom diskretnom trenutku vremena, zavisno od trenutnog stanja mašine i simbola na koji glava mašine trenutno pokazuje, određuju šta će se desiti u narednom trenutku. Sam rad mašine sastoji se u izvršavanju sledećih operacija:

1. prelazak iz jednog stanja u drugo;
2. zamena simbola u ćeliji drugim simbolom, pri čemu je moguće da se on zameni i istim simbolom, tj. da sadržaj ćelije ostane nepromenjen;
3. pomeranje glave mašine za jedno mesto ulevo ili jedno mesto udesno.

U pomeranju glave ključnu ulogu igraju dva posebna simbola L i R, pri čemu je, jasno, sa L određeno pomeranje glave ulevo, a sa R udesno. Pri tome može da postoji najviše jedna instrukcija koja za trenutno stanje i simbol na traci ukazuje na dalji rad mašine. Mašina završava sa radom kada za trenutno stanje i simbol na traci ne postoji nijedna instrukcija koja predviđa dalje ponašanje mašine.

Indirektno, smatraćemo da Turingova mašina ima mogućnost brisanja. Naime, alfabetu Σ se može dodati novi pomoćni simbol B (čita se *prazno* ili *blanko*) koji je moguće upisati umesto razmatranog simbola, čime bi on bio tretiran kao obrisan.

Sada ćemo dati formalnu definiciju Turingove mašine. Pod *Turingovom mašinom* podrazumevamo petorku $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$, gde je

A – neprazan konačan skup, *skup stanja* mašine $\mathcal{A}$;
a_0 –element iz A, *početno stanje* mašine $\mathcal{A}$;
X – neprazan konačan skup, *ulazni alfabet* mašine $\mathcal{A}$;
Σ –neprazan konačan skup koji sadrži X i simbol $\# \notin X$, *alfabet trake* mašine $\mathcal{A}$;
δ – parcijalno preslikavanje iz $A \times \Sigma$ u $A \times \Sigma \setminus \{\#\} \times \{L, R\}$, *funkcija prelaza* mašine $\mathcal{A}$.

Pod pojmom *konfiguracije* Turingove mašine A podrazumevaćemo proizvoljnu trojku $(a, \alpha, i) \in A \times (\Sigma \setminus \{\#\})^* \times \mathbb{N}$. U datom neformalnom modelu mašine, ova konfiguracija bi odgovarala trenutku kada se mašina $\mathcal{A}$ nalazi u stanju a, u nepraznom delu trake je upisana reč α, čitano sleva na desno, a i je redni broj ćelije na koju glava pokazuje u tom trenutku, takođe brojano sleva na desno, od početka trake. Ako je $\delta(a, \xi) = (b, \xi', D)$, za neki $(a, \xi) \in A \times \Sigma$ i $D \in \{L, R\}$, tada izraz

$$\delta(a, \xi) = (b, \xi', D)$$

nazivamo *instrukcijom* mašine A. Rad Turingove mašine $\mathcal{A}$ sastoji se u nizu prelazaka iz jedne konfiguracije u drugu, pri čemu su ti prelasci određeni instrukcijama te mašine.

Neka je $(a, \xi_1 \xi_2 \cdots \xi_n, i)$ proizvoljna konfiguracija, za $\xi_1, \xi_2, \ldots, \xi_n \in \Sigma \setminus \{\#\}$. Razlikujemo više tipova prelaza:

1. Neka je $i \in [1, n]$ i $\delta(a, \xi_i) = (b, \xi, R)$. Tada se sa date konfiguracije prelazi na konfiguraciju $(a, \xi_1 \ldots \xi_{i-1} \xi \xi_{i+1} \ldots \xi_n, i+1)$, što pišemo

$$(a, \xi_1 \xi_2 \ldots \xi_n, i) \underset{\mathcal{A}}{\longrightarrow} (a, \xi_1 \ldots \xi_{i-1} \xi \xi_{i+1} \ldots \xi_n, i+1).$$

Drugim rečima, instrukcija $\delta(a, \xi_i) = (b, \xi, R)$ kaže sledeće: pređi iz stanja a u stanje b, zameni simbol ξ_i u i-toj ćeliji simbolom ξ i pomeri glavu za jedno mesto udesno, nad ćeliju sa rednim brojem $i+1$.

2. Neka je $i \in [2, n]$ i $\delta(a, \xi_i) = (b, \xi, L)$. Tada se sa date konfiguracije prelazi na konfiguraciju $(b, \xi_1 \ldots \xi_{i-1} \xi \xi_{i+1} \ldots \xi_n, i-1)$, što pišemo

$$(a, \xi_1 \xi_2 \ldots \xi_n, i) \underset{\mathcal{A}}{\longrightarrow} (b, \xi_1 \ldots \xi_{i-1} \xi \xi_{i+1} \ldots \xi_n, i-1).$$

Drugim rečima, instrukcija $\delta(a,\xi_i) = (b,\xi,L)$ kaže sledeće: pređi iz stanja a u stanje b, zameni simbol ξ_i u i-toj ćeliji simbolom ξ i pomeri glavu za jedno mesto ulevo, nad ćeliju sa rednim brojem $i-1$.

3. Neka je $i = n+1$. Tada glava pokazuje na praznu ćeliju, tj. na ćeliju u kojoj je zabeležen znak #, pa razlikujemo sledeće podslučajeve:

3.1. Ako je $\delta(a,\#) = (b,\xi,R)$, tada se sa date konfiguracije prelazi na konfiguraciju $(b,\xi_1\dots\xi_n\xi,n+2)$, što pišemo

$$(a,\xi_1\xi_2\dots\xi_n,n+1)\underset{\mathcal{A}}{\longrightarrow}(b,\xi_1\dots\xi_n\xi,n+2).$$

Znači, instrukcija $\delta(a,\#) = (b,\xi,R)$ kaže: zabeleži simbol ξ u $(n+1)$-vu ćeliju i pomeri glavu za jedno mesto udesno, nad $(n+2)$-gu ćeliju.

3.2. Ako je $\delta(a,\#) = (b,\xi,L)$, tada se sa date konfiguracije prelazi na konfiguraciju $(b,\xi_1\dots\xi_n\xi,n)$, što pišemo

$$(a,\xi_1\xi_2\dots\xi_n,n+1)\underset{\mathcal{A}}{\longrightarrow}(b,\xi_1\dots\xi_n\xi,n).$$

Prema tome, u ovom slučaju instrukcija $\delta(a,\#) = (b,\xi,L)$ kaže: upiši simbol ξ u $(n+1)$-vu ćeliju i pomeri glavu za jedno mesto ulevo, nad n-tu ćeliju.

Konačan niz konfiguracija (a_k,α_k,i_k), $k \in [0,n]$, gde je $n \in \mathbb{N}^0$, nazivaćemo *izračunavanjem* u Turingovoj mašini $\mathcal{A}$, ako za svako $k \in [0,n-1]$ važi

$$(a_k,\alpha_k,i_k)\underset{\mathcal{A}}{\longrightarrow}(a_{k+1},\alpha_{k+1},i_{k+1}).$$

Ove pojedinačne prelaze nazivamo *koracima* tog izračunavanja. Broj koraka u ovom izračunavanju, tj. broj n, nazivamo *dužinom* tog izračunavanja.

Ako su (a,α,i) i (b,β,j) konfiguracije za koje postoji izračunavanje (a_k,α_k,i_k), $k \in [0,n]$, takvo da je

$$(a,\alpha,i) = (a_0,\alpha_0,i_0) \quad \text{i} \quad (b,\beta,j) = (a_n,\alpha_n,i_n),$$

tada pišemo

$$(a,\alpha,i)\overset{*}{\underset{\mathcal{A}}{\longrightarrow}}(b,\beta,j),$$

pri čemu, bez opasnosti od zabune, i taj izraz nazivamo izračunavanjem. Jasno, to izračunavanje je dužine 0 ako i samo ako je $(a,\alpha,i) = (b,\beta,j)$. Ukoliko je jasno o kojoj se Turingovoj mašini radi, u oznakama $\underset{\mathcal{A}}{\longrightarrow}$ i $\overset{*}{\underset{\mathcal{A}}{\longrightarrow}}$ izostavljamo $\mathcal{A}$, tj. pišemo samo $\rightarrow$ i $\overset{*}{\rightarrow}$. Drugim rečima, na skupu $A\times\Sigma^*\times\mathbb{N}$ svih konfiguracija definisana je relacija $\rightarrow$ tako da su dve konfiguracije (a,α,i) i (b,β,j) u relaciji $\rightarrow$ ako i samo ako se iz konfiguracije (a,α,i) može preći u (b,β,j) u jednom koraku. Takođe, definisana je i relacija $\overset{*}{\rightarrow}$, koja predstavlja refleksivno-tranzitivno zatvorenje relacije $\rightarrow$.

Napomena 6. Kod ovako uvedenog pojma Turingove mašine postoji mogućnost da glava posle čitanja simbola sa trake ne ide obavezno levo ili desno, već može ostati i na istom mestu. Naime, ako želimo da mašina, koja se nalazi u stanju a i čita simbol ξ sa trake, pređe u stanje a_1, ispiše ξ_1 na traci i glava ostane na istom mestu, skupu stanja mašine $\mathcal{A}$ dodajemo novo stanje b, a instrukcijama mašine $\mathcal{A}$ dodajemo sledeće instrukcije:

$$\begin{aligned}&\delta(a,\xi) = (b,\xi_1,R),\\ &\delta(b,\mu) = (a_1,\mu,L), \quad \text{za svaki simbol trake } \mu.\end{aligned}$$

Takođe, postoje i tzv. Turingove mašine kod kojih ulazna traka ima više staza. To su, zapravo, Turingove mašine kod kojih alfabet trake, a i ulazni alfabet, jesu Descartesovi proizvodi nekih skupova. Mašina ima onoliko staza koliko činilaca ima u tim Descartesovim proizvodima. Turingova mašina sa k staza, za neki $k \in \mathbb{N}$, jeste mašina kod koje alfabet trake jeste neki skup k-torki.

Analogno pojmu raspoznavanja jezika automatima uvodi se i pojam raspoznavanja jezika Turingovim mašinama. Kažemo da Turingova mašina $\mathcal{A} = (A,a_0,X,\Sigma,\delta)$ *raspoznaje jezik* $L \subseteq X^*$ *pomoću skupa* $T \subseteq A$ ako je

$$L = \{u \in X^* \,|\, (a_0,u,1) \xrightarrow{*} (a,\alpha,i), \ \text{gde je } a \in T, \alpha \in \Sigma^*, i \in \mathbb{N}\},$$

tj.

$$u \in L \Leftrightarrow (a_0,u,1) \xrightarrow{*} (a,\alpha,i) \ \text{ gde je } a \in T, \alpha \in \Sigma^*, i \in \mathbb{N},$$

a za jezik L u tom slučaju kažemo da je *raspoznatljiv Turingovom mašinom*. Skup T nazivamo skupom *završnih stanja* Turingove mašine $\mathcal{A}$.

Neformalno, Turingova mašina $\mathcal{A}$ raspoznaje jezik L ako je L skup svih reči ulaznog alfabeta koje vode do nekog završnog stanja. Često se kaže i da Turingova mašina $\mathcal{A}$ *prihvata reč* $u \in X^*$ *skupom* T ako je $(a_0,u,1) \xrightarrow{*} (a,\alpha,i)$, za neko stanje $a \in T$.

Primetimo da se ovde govori samo o prihvatanju reči nad alfabetom X, dok simboli iz skupa $\Sigma \setminus X$ imaju pomoćnu ulogu, koja se može uporediti sa ulogom pomoćnih simbola kod gramatika.

Možemo da uvedemo i pojam nedeterminističke Turingove mašine. Kao što se i očekuje, to će biti Turingova mašina kod koje δ nije parcijalno preslikavanje, već je relacija, pa za neke $a \in A$ i $\xi \in \Sigma$ ima više izbora za $\delta(a,\xi)$, tj. $\delta(a,\xi)$ je podskup skupa $A \times \Sigma \setminus \{\#\} \times \{L,R\}$. Tako ćemo ranije definisanu Turingovu mašinu nazivati i *determinističkom Turingovom mašinom*, a petorku $\mathcal{A} = (A,a_0,X,\Sigma,\delta)$ nazivamo *nedeterminističkom Turingovom mašinom* ako A, a_0, X i Σ imaju isto značenje kao u definiciji determinističke Turingove mašine, dok je δ relacija između skupova $A \times \Sigma$ i $A \times \Sigma \setminus \{\#\} \times \{L,R\}$.

Jezik $L \subseteq X^*$ je raspoznatljiv nedeterminističkom Turingovom mašinom $\mathcal{A} = (A,a_0,X,\Sigma,\delta)$ pomoću skupa $T \subseteq A$ ako je

$$L = \{u \in X^* \,|\, (a_0,u,1) \xrightarrow{*} (a,\alpha,i) \ \text{gde je } a \in T, \alpha \in \Sigma^*, i \in \mathbb{N}\},$$

tj.

$$u \in L \Leftrightarrow (a_0, u, 1) \xrightarrow{*} (a, \alpha, i) \text{ gde je } a \in T, \alpha \in \Sigma^*, i \in \mathbb{N}.$$

Skup T je skup *završnih stanja* Turingove mašine $\mathcal{A}$. Neformalno, nedeterministička Turingova mašina $\mathcal{A}$ raspoznaje jezik L ako je L skup svih reči ulaznog alfabeta koje vode iz početnog do nekog od završnih stanja, za neki izbor niza prelaza. Često se kaže i da Turingova mašina $\mathcal{A}$ *prihvata reč $u \in X^*$ skupom T* ako postoji način da se, polazeći od a_0 i reči u na ulaznoj traci, dođe do nekog završnih stanja.

Kao i u slučaju automata bez izlaza, i ovde važi da determinističke i nedeterminističke Turingove mašine imaju iste mogućnosti u raspoznavanju jezika.

Zadatak 4.156. *Dokazati da je jezik raspoznatljiv determinističkom Turingovom mašinom ako i samo ako je raspoznatljiv nedeterminističkom Turingovom mašinom.*

Rešenje: Jasno je da se svaka deterministička Turingova mašina može smatrati nedeterminističkom, pa je direktni smer u dokazu zadatka trivijalan.

Dokazaćemo obrat. Pretpostavimo da nedeterministička Turingova mašina $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ raspoznaje jezik L pomoću skupa T. Opisaćemo determinističku Turingovu mašinu $\mathcal{A}' = (A', a_0, X, \Sigma', \delta')$ koja raspoznaje L.

Naime, skup stanja $\mathcal{A}'$ se poklapa sa skupom stanja A, a takođe će i skup završnih stanja biti T. Za fiksirane $a \in A$ i $\xi \in \Sigma$, takve da je $\delta(a,\xi)$ definisano, označimo sa $k(a,\xi)$ broj definisanih prelaza iz stanja a za ulaz ξ, i numerišimo redom prelaze $\delta(a,\xi) = (a', \mu, D)$, gde je $D \in \{R, L\}$, brojevima od 1 do $k(a,\xi)$. Uvedimo oznaku $r = \max\{k(a,\xi) \mid a \in A,\ \xi \in \Sigma\}$. Svaka transformacija u $\mathcal{A}'$ može se okarakterisati konačnim nizom brojeva između 1 i r koji predstavljaju brojeve prelaza koji su upotrebljeni u toj transformaciji. Jasno je da svaki ovakav niz brojeva ne mora da predstavlja transformaciju, jer je moguće da za izvestan par (a,ξ) ima manje od r izbora za prelaz, tj. $k(a,\xi) < r$.

Neka je sada Σ' skup koji se sastoji od nekih trojki oblika $[x, i, \xi]$, gde je $x \in X$, $i \in [0, r]$, $\xi \in \Sigma$, pri čemu skup Σ' ne mora da obuhvata sve trojke tog oblika. Naime, kako alfabet trake čine trojke, to možemo smatrati da je traka podeljena na tri staze. U prvoj će se nalaziti ulazna reč. Na drugoj su na početku nule, a nadalje $\mathcal{A}'$ generiše niz brojeva skupa $\{1, 2, \ldots, r\}$, gde svaki od tih brojeva predstavlja broj prelaza koji je upotrebljen. Na trećoj stazi je na početku kopija ulazne reči, a u toku rada se na njoj simulira rad mašine $\mathcal{A}$, za ulaze sa prve staze i prelaze određene nizom u drugoj stazi. Pri tome X identifikujemo sa trojkama oblika $[x, 0, x]$, za svaki $x \in X$. Dakle, ako je $\mathcal{A}'$ u stanju a i na ulazu se javlja trojka (x, i, ξ), i ako sa $\delta_i(a,\xi)$ označimo i-ti prelaz u našoj numeraciji nedeterminističke mašine $\mathcal{A}$, definisan za par (a,ξ), tada za $\delta_i(a,\xi) = (a', \xi', R)$ automat $\mathcal{A}'$ prelazi u stanje a', na treću stazu upisuje simbol ξ' i pomera glavu jedno mesto udesno. Analogno se razmatra i slučaj $\delta_i(a,\xi) = (a', \xi', L)$.

Jasno je da ukoliko $\mathcal{A}$ prihvata reč u tada će nas odgovarajući izbor transformacija, tj. niza na drugoj stazi, dovesti do završnog stanja i kod mašine

$\mathcal{A}'$. Obratno, ako mašina $\mathcal{A}$ ne prihvata reč u tada nas nijedan izbor niza na drugoj stazi, koji predstavlja izbor odgovarajućih prelaza, ne može dovesti do završnog stanja u $\mathcal{A}'$. □

Zadatak 4.157. *Konstruisati Turingovu mašinu koja raspoznaje sve binarne reči koje se završavaju nulom.*

Rešenje: Neka je $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ Turingova mašina u kojoj je

$$\begin{aligned} A &= \{a_0, a_1, a_2\}, \\ X &= \{0,1\}, \\ \Sigma &= \{0,1,\#\}. \end{aligned}$$

Skup završnih stanja je $T = \{a_2\}$, a funkcija prelaza δ neka je data na sledeći način:

$$\begin{array}{ll} \text{1.a} & \delta(a_0,0) = (a_0,0,R) \\ \text{1.b.} & \delta(a_0,1) = (a_0,1,R) \\ \text{1.c.} & \delta(a_0,\#) = (a_1,\#,L) \\ \text{2.} & \delta(a_1,0) = (a_2,0,R) \end{array}$$

Stanje a_0 skenira desno slovo ulazne reči, dok a_1 proverava poslednje slovo. Ovako definisana Turingova mašina prepoznaje traženi jezik. □

Zadatak 4.158. *Konstruisati Turingovu mašinu koja briše poslednje slovo ulazne binarne reči.*

Rešenje: Neka je $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ Turingova mašina, sa skupovima

$$\begin{aligned} A &= \{a_0, a_1, a_2, a_3, a_4\}, \\ X &= \{0,1\}, \\ \Sigma &= \{0,1,\#\}. \end{aligned}$$

Skup završnih stanja je $T = \{a_4\}$, a funkcija prelaza δ je data na sledeći način:

$$\begin{array}{ll} \text{1.a} & \delta(a_0,0) = (a_1,0,R) \\ \text{1.b.} & \delta(a_0,1) = (a_1,1,R) \\ \text{2.a.} & \delta(a_1,0) = (a_1,0,R) \\ \text{2.b.} & \delta(a_1,1) = (a_1,1,R) \\ \text{2.c.} & \delta(a_1,\#) = (a_2,\#,L) \\ \text{3.a.} & \delta(a_2,0) = (a_3,\#,L) \\ \text{3.b.} & \delta(a_2,1) = (a_3,\#,L) \\ \text{4.a.} & \delta(a_3,0) = (a_3,0,L) \\ \text{4.b.} & \delta(a_3,1) = (a_3,1,L) \\ \text{4.c.} & \delta(a_3,\#) = (a_4,\#,L) \end{array}$$

Stanje a_0 skenira desno slovo ulazne reči, dok a_1 proverava poslednje slovo. Ovako definisana Turingova mašina prepoznaje traženi jezik. □

Zadatak 4.159. *Opisati Turingovu mašinu $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ koja raspoznaje jezik $L = \{x^n y^n \mid n \in \mathbb{N}\}$.*

Rešenje: Neka je

$$A = \{a_0, a_1, a_2, a_3, a_4\},$$
$$X = \{x, y\},$$
$$\Sigma = \{x, y, \#, \alpha, \beta\}.$$

Skup završnih stanja je $T = \{a_4\}$, a funkcija prelaza δ je data na sledeći način. Ukoliko prvi ulazni simbol jeste x, mašina ga menja u α, dok se u suprotnom zaustavlja. Zatim se kreće udesno, do prvog y, i menja ga u β. Onda se vraća ulevo do poslednjeg α, pa prvi desni x, ako ga ima, menja u α i ponovo odlazi udesno u potragu za novim y. Dakle, u svakom krugu mašina menja prvi levi x u α i prvi levi y u β, a dolazi do završnog stanja samo ukoliko nema više x-ova levo od prvog levog y i iza poslednjeg desnog y se nalazi #. Funkcija prelaza je data sa:

1.a $\delta(a_0, x) = (a_1, \alpha, R)$
1.b. $\delta(a_0, \beta) = (a_3, \beta, R)$
2.a. $\delta(a_1, x) = (a_1, x, R)$
2.b. $\delta(a_1, \beta) = (a_1, \beta, R)$
2.c. $\delta(a_1, y) = (a_2, \beta, L)$
3.a. $\delta(a_2, \beta) = (a_2, \beta, L)$
3.b. $\delta(a_2, \alpha) = (a_0, \alpha, R)$
3.c. $\delta(a_2, x) = (a_2, x, L)$
4.a. $\delta(a_3, \beta) = (a_3, \beta, R)$
4.b. $\delta(a_3, \#) = (a_4, \#, R)$

Sledeća šema prikazuje ponašanje mašine $\mathcal{A}$ u slučaju kada je ulazna reč x^3y^3.

konfiguracija	pravilo
$(a_0, xxxyyy, 1)$	start
$(a_1, \alpha xxyyy, 2)$	1
$(a_1, \alpha xxyyy, 3)$	2.a
$(a_1, \alpha xxyyy, 4)$	2.a
$(a_2, \alpha xx\beta yy, 3)$	2.c
$(a_2, \alpha xx\beta yy, 2)$	3.c
$(a_2, \alpha xx\beta yy, 1)$	3.c
$(a_0, \alpha xx\beta yy, 2)$	3.b
$(a_1, \alpha\alpha x\beta yy, 3)$	1
$(a_1, \alpha\alpha x\beta yy, 4)$	2.a
$(a_1, \alpha\alpha x\beta yy, 5)$	2.b
$(a_2, \alpha\alpha x\beta\beta y, 4)$	2.c
$(a_2, \alpha\alpha x\beta\beta y, 3)$	3.c

konfiguracija	pravilo
$(a_2, \alpha\alpha x\beta\beta y, 2)$	3.c
$(a_0, \alpha\alpha x\beta\beta y, 3)$	3.b
$(a_1, \alpha\alpha\alpha\beta\beta y, 4)$	1.a
$(a_1, \alpha\alpha\alpha\beta\beta y, 5)$	2.b
$(a_1, \alpha\alpha\alpha\beta\beta y, 6)$	2.b
$(a_2, \alpha\alpha\alpha\beta\beta\beta, 5)$	2.c
$(a_2, \alpha\alpha\alpha\beta\beta\beta, 4)$	3.a
$(a_2, \alpha\alpha\alpha\beta\beta\beta, 3)$	3.a
$(a_3, \alpha\alpha\alpha\beta\beta\beta, 4)$	4.a
$(a_3, \alpha\alpha\alpha\beta\beta\beta, 5)$	4.a
$(a_3, \alpha\alpha\alpha\beta\beta\beta, 6)$	4.a
$(a_3, \alpha\alpha\alpha\beta\beta\beta, 7)$	4.a
$(a_4, \alpha\alpha\alpha\beta\beta\beta\#, 8)$	4.b

Ovde su,takođe, data i pravila koja se koriste u svakom od koraka. □

Zadatak 4.160. *Konstruisati Turingovu mašinu* $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ *koja raspoznaje jezik* $L = \{x^n y^n z^n \,|\, n \in \mathbb{N}\}$.

Rešenje: Stavimo

$$\begin{aligned} A &= \{a_0, a_x, a_y, a_z, a_\beta, a_\gamma, a_f\},\\ X &= \{x, y, z\},\\ \Sigma &= \{x, y, z, \#, \alpha, \beta, \gamma\}. \end{aligned}$$

Skup završnih stanja je $T = \{a_f\}$, a funkciju prelaza δ definišimo na osnovu sledećeg. Ukoliko prvi ulazni simbol jeste x, mašina ga menja u α, dok se u suprotnom zaustavlja. Zatim se kreće udesno, do prvog y i menja ga u β. Daljim kretanjem udesno, traži prvi z i menja ga u γ. Onda se vraća ulevo do poslednjeg α, pa prvi desni x, ako ga ima, menja u α i ponovo odlazi udesno u potragu za novim y, menjajući ga u β, a, potom, traži novi z. Dakle, u svakom krugu mašina menja prvi levi x u α i prvi levi y u β i prvi levi z u γ, a dolazi do završnog stanja samo ukoliko nema više x-ova levo od prvog levog y, nema više y-a levo od prvog levog z i iza poslednjeg desnog z se nalazi #. Funkcija prelaza je data sa:

$$\begin{array}{ll}
\text{1.a.} & \delta(a_0, x) = (a_x, \alpha, R)\\
\text{1.b.} & \delta(a_0, x) = (a_\beta, \beta, R)\\
\text{2.a.} & \delta(a_x, x) = (a_x, x, R)\\
\text{2.b.} & \delta(a_x, y) = (a_y, \beta, R)\\
\text{2.c.} & \delta(a_x, \beta) = (a_x, \beta, R)\\
\text{3.a.} & \delta(a_y, y) = (a_y, y, R)\\
\text{3.b.} & \delta(a_y, c) = (a_z, \gamma, L)\\
\text{3.c.} & \delta(a_y, \gamma) = (a_y, \gamma, R)\\
\text{4.a.} & \delta(a_c, x) = (a_c, x, L)\\
\text{4.b.} & \delta(a_c, y) = (a_c, y, L)\\
\text{4.c.} & \delta(a_c, \alpha) = (a_0, \alpha, R)\\
\text{4.d.} & \delta(a_c, \beta) = (a_c, \beta, L)\\
\text{4.e.} & \delta(a_c, \gamma) = (a_c, \gamma, L)\\
\text{5.a.} & \delta(a_\beta, \beta) = (a_\beta, \beta, R)\\
\text{5.b.} & \delta(a_\beta, \gamma) = (a_\gamma, \gamma, R)\\
\text{6.a.} & \delta(a_\gamma, \gamma) = (a_\gamma, \gamma, R)\\
\text{6.b.} & \delta(a_\gamma, \#) = (a_f, \#, R)
\end{array}$$

Ovako definisana Turingova mašina raspoznaje dati jezik. □

Zadatak 4.161. *Opisati Turingovu mašinu koja raspoznaje jezik*

$$L = \{a^n b^n c^{mn} \,|\, n \in \mathbb{N}\}.$$

Rešenje: Kako je postupak konstrukcije tražene Turingove mašine sličan postupku iz prethodnog zadatka, opisaćemo neformalno algoritam za njenu konstrukciju:

1. Mašina se nalazi u inicijalnom stanju. Najpre se kreće nadesno da bi proverila da li ulazna reč ima traženi raspored slova a,b,c, tj. da li ulazna reč pripada jeziku $a^*b^*c^*$. U protivnom se mašina zaustavlja.
2. Vraća se do prvog slova ulazne reči.
3. Prvo slovo a briše (zamenjuje simbolom #) i kreće se na desno do prvog levog slova b. Zatim se naizmenično kreće od najlevljeg b do najlevljeg c i pri svakoj zameni b sa β odgovarajuće c zamenjuje sa #. Ako pri zameni nekog b ne ostane ni jedno c mašina prestaje da radi. Postupak se nastavlja sve dok svi b-ovi ne budu zamenjeni. Na kraju ovog koraka ulazna reč je transformisana u oblik $a^{m-1}\beta^n c^{k-n}$.
4. Mašina svako β zamenjuje sa b transformišući dobijenu reč u oblik $a^{m-1}b^n c^{k-n}$ i vraća se do krajnjeg levog a. Zatim nastavlja **3.** i **4.** sve dok ima slova a, a onda ispituje šta je sa c-ovima. Ako ni jedno slovo c nije preostalo mašina raspoznaje dati jezik.

Formalnu konstrukciju Turingove mašine prepuštamo čitaocu. □

Zadatak 4.162. *Konstruisati Turingovu mašinu $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ koja raspoznaje jezik $L = \{w\overline{w} \mid w \in \{0,1\}^*\}$.*

Rešenje: Neka je

$$A = \{a_i, a_0, a_1, a_{\overline{0}}, a_{\overline{1}}, \overline{a}, a_f\},$$
$$X = \{0,1\},$$
$$\Sigma = \{0,1,\#\}.$$

Skup završnih stanja je $T = \{a_f\}$, a funkciju prelaza δ definišimo na osnovu sledećeg. Prvi ulazni simbol mašina menja u #. Zatim se kreće udesno, ostavljajući na svom mestu sve simbole 0 i 1 dok ne dodje do znaka #. Onda se vraća ulevo menjajući poslednji simbol znakom #, ako je on jednak prvom simbolu koji je zamenjen #. Mašina ponovo odlazi udesno ponavljajući isti postupak. Zaustavlja se ako su svi simboli zamenjeni sa #, kada se nalazi u stanju a_f. Dakle, funkcija prelaza je data sa:

$$
\begin{array}{ll}
\text{1.a.} & \delta(a_i,0) = (a_0,\#,R)\\
\text{1.b.} & \delta(a_i,1) = (a_1,\#,R)\\
\text{1.c.} & \delta(a_i,\#) = (a_f,\#,R)\\
\text{2.a.} & \delta(a_0,0) = (a_0,0,R)\\
\text{2.b.} & \delta(a_0,1) = (a_0,1,R)\\
\text{2.c.} & \delta(a_0,\#) = (a_{\overline{0}},\#,L)\\
\text{3.a.} & \delta(a_1,0) = (a_1,0,R)\\
\text{3.b.} & \delta(a_1,1) = (a_1,1,R)\\
\text{3.c.} & \delta(a_1,\#) = (a_{\overline{1}},\#,L)\\
\text{4.} & \delta(a_{\overline{0}},0) = (a_b,\#,L)\\
\text{5.} & \delta(a_{\overline{1}},1) = (a_b,\#,L)\\
\text{6.a.} & \delta(\overline{a},0) = (a_b,0,L)\\
\text{6.b.} & \delta(\overline{a},1) = (a_b,1,L)\\
\text{6.c.} & \delta(\overline{a},\#) = (a_i,\#,R)
\end{array}
$$

Ovako definisana Turingova mašina raspoznaje dati jezik. □

Zadatak 4.163. *Konstruisati Turingovu mašinu* $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ *koja raspoznaje jezik* $L = \{u \in \{x,y\}^*, \mid |u|_x = |u|_y\}$.

Rešenje: Neka je

$$A = \{a_0, a_1, a_2, a_3, a_4, a_5\},$$
$$X = \{x, y\},$$
$$\Sigma = \{x, y, \alpha, \#\}.$$

Inicijalno stanje je a_0, završno stanje je a_4, a funkciju prelaza δ definišaćemo tako što nalazimo početak i kraj ulazne reči i ako ne nađemo ni jedan simbol x moramo biti sigurni da nema ni jednog y:

1.a. $\delta(a_0, x) = (a_1, \alpha, R)$
1.b. $\delta(a_0, y) = (a_2, \alpha, R)$
1.c. $\delta(a_0, \alpha) = (a_4, \#, L)$
1.d. $\delta(a_0, \#) = (a_0, \alpha, R)$
2.a. $\delta(a_1, x) = (a_1, x, R)$
2.b. $\delta(a_1, y) = (a_3, \alpha, L)$
2.c. $\delta(a_1, \alpha) = (a_1, \alpha, R)$
2.d. $\delta(a_1, \#) = (a_5, \#, L)$
3.a. $\delta(a_2, x) = (a_3, x, L)$
3.b. $\delta(a_2, y) = (a_2, y, R)$
3.c. $\delta(a_2, \alpha) = (a_2, \alpha, R)$
3.d. $\delta(a_2, \#) = (a_5, \#, L)$
4.a. $\delta(a_3, x) = (a_3, x, L)$
4.b. $\delta(a_3, y) = (a_3, y, L)$
4.c. $\delta(a_3, \alpha) = (a_3, \alpha, L)$
4.d. $\delta(a_3, \#) = (a_0, \#, R)$

Ovako definisana Turingova mašina raspoznaje reči u kojima je broj slova x jednak broju slova y. □

Zadatak 4.164. *Konstruisati Turingovu mašinu* $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ *koja raspoznaje jezik* $L = \{u \in \{x,y\}^*, \mid u = \overline{u}\}$ *(L je skup palindroma).*

Rešenje: Neka je

$$A = \{a_0, a_1, a_2, a_3, a_4, a_5\},$$
$$X = \{x, y\},$$
$$\Sigma = \{x, y, \#\}.$$

Inicijalno stanje je a_0, završno stanje je a_5, a funkciju prelaza δ definišaćemo tako što brišemo prvo i poslednje slovo sa ulazne trake, ukoliko su ta slova jednaka, pa krug kreće od početka:

1.a. $\delta(a_0, x) = (a_1, \#, R)$
1.b. $\delta(a_0, y) = (a_2, \#, R)$
1.c. $\delta(a_0, \#) = (a_5, \#, L)$
2.a. $\delta(a_1, x) = (a_1, x, R)$

$$\begin{array}{ll}
2.b. & \delta(a_1,y) = (a_1,y,R) \\
2.c. & \delta(a_1,\#) = (a_3,\#,L) \\
3.a. & \delta(a_2,x) = (a_2,x,R) \\
3.b. & \delta(a_2,y) = (a_2,y,R) \\
3.c. & \delta(a_2,\#) = (a_4,\#,L) \\
4. & \delta(a_3,x) = (a_5,\#,L) \\
5. & \delta(a_4,y) = (a_5,\#,L) \\
6.a. & \delta(a_5,x) = (a_5,x,L) \\
6.b. & \delta(a_5,y) = (a_5,y,L) \\
6.c. & \delta(a_5,\#) = (a_0,\#,R)
\end{array}$$

Ova Turingova mašina raspoznaje sve palindrome. □

Zadatak 4.165. (a) *Konstruisati Turingovu mašinu koja binarnom zapisu ulaznog broja N dodaje* 1. *(Da bi obezbedili još jedno mesto na poziciji glave ulazne trake, pretpostavite, inicijano, u stanju a_0, da je ispred binarnog zapisa broja N znak* \$*).*
(b) *U ovoj Turingovoj mašini naći izračunavanje koje ulaznu reč* \$111 *uvećava za* 1.

Rešenje: Neka je

$$\begin{aligned} A &= \{a_0,a_1,a^0,a^1,a_2,a_3\}, \\ X &= \{0,1\}, \\ \Sigma &= \{\$,0,1,\#\}. \end{aligned}$$

Skup završnih stanja je $T = \{a_3\}$, a mašina se nalazi u stanjima a^0 i a^1 kada se na bit vece težine prenosi 0, tj. 1, redom. Dakle, funkcija prelaza je data sa:

$$\begin{array}{ll}
1. & \delta(a_0,\$) = (a_1,\$,R) \\
2.a. & \delta(a_1,0) = (a_1,0,R) \\
2.b. & \delta(a_1,1) = (a_1,1,R) \\
2.c. & \delta(a_1,\#) = (a^0,\#,L) \\
3.a. & \delta(a^0,\$) = (a_3,\$,R) \\
3.b. & \delta(a^0,0) = (a^0,0,L) \\
3.c. & \delta(a^0,1) = (a^0,1,L) \\
4.a. & \delta(a^1,\$) = (a_2,1,L) \\
4.b. & \delta(a^1,0) = (a^0,1,L) \\
4.c. & \delta(a^1,1) = (a^1,0,L) \\
5. & \delta(a_2,1) = (a_3,\#,R)
\end{array}$$

Za datu ulaznu reč \$111 Turingova mašina vrši sledeće izračunavanje:

konfiguracija	pravilo
$(a_0,\$111,\$)$	start
$(a_1,\$111,2)$	1
$(a_1,\$111,3)$	2.b
$(a_1,\$111,4)$	2.b

$(a_1,\$111,5)$	2.b
$(a^1,\$111,4)$	2.c
$(a^1,\$110,3)$	4.c
$(a^1,\$100,2)$	4.c
$(a^1,\$000,1)$	4.c
$(a_2,\#1000,1)$	4.a
$(a_3,\#1000,2)$	5

Dakle, Turingova mašina je izračunala vrednost za 1 veću od datog binarnog zapisa, tj.111+1=1000. □

Zadatak 4.166. *Opisati Turingovu mašinu* $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ *koja utvrđuje da li su u ulaznoj reči nad alfabetom* $\{(,)\}$ *zagrade uparene.*

Rešenje: Neka je

$$A = \{a_0, a_1, a_2, a_3\},$$
$$X = \{(,)\},$$
$$\Sigma = \{(,), \varrho, \#\}.$$

Inicijalno stanje je $\{a_0\}$, skup završnih stanja je $T = \{a_3\}$, a funkciju prelaza δ definišimo na osnovu sledećeg: Mašina se kreće udesnotražeći desnu zagradu i markira tu poziciju. Zatim se kreće ulevo, pronalazeći levu zagradu, koja odgovara pronađenoj desnoj zagradi i markira i njenu poziciju. Mašina se zaustavlja ako nema više levih ili ako ima suviše desnih zagrada. Dakle, funkcija prelaza je data sa:

1.a. $\delta(a_0, () = (a_0, (, R)$
1.b. $\delta(a_0,)) = (a_1, \varrho, L)$
1.c. $\delta(a_0, \varrho) = (a_0, \varrho, R)$
1.d. $\delta(a_0, \#) = (a_2, \#, L)$
2.a. $\delta(a_1, () = (a_0, \varrho, R)$
2.b. $\delta(a_1, \varrho) = (a_1, \varrho, L)$
3.a. $\delta(a_2, \varrho) = (a_2, \varrho, L)$
3.b. $\delta(a_2, \#) = (a_3, \#, L)$

Ovako definisana Turingova mašina utvrđuje da li su zagrade uparene, ispitujući unatrag, u stanju a_2, da li su sve pozicije markirane . □

Zadatak 4.167. *Konstruisati Turingovu mašinu koja, za broja* $m, n \in \mathbb{N}$, *izračunava vrednost funkcije* $f(m, n) = \max(m - n, 0)$.

Rešenje: Kako je teško zadati dva ulaza, pretpostavićemo da se na ulaznoj traci tražene Turingove mašine $\mathcal{A}$, nalazi binarna reč $0^m 1 0^n$, a da mašina završava sa radom kada je na traci reč $0^{f(m,n)}$. Mašina najpre prvu levu nulu zamenjuje # simbolom. Zatim se kreće udesno tražeći 1, a onda opet desno traži 0 i zamenjuje je sa 1. Vraćanjem nalevo $\mathcal{A}$ pronalazi prvu levu nulu i zamenjuje je #. Ponavljanje se prekida u dva slučaja:

1. Kada, kretanjem udesno u potrazi za nulom, mašina naiđe na #. Tada je n nula u reči 0^m10^n zamenjeno jedinicom i $m-1$ od m nula zamenjeno simbolom #. $\mathcal{A}$ zamenjuje $n+1$-u jedinici jednom nulom i sa n simbola #, te na ulaznoj traci ostaje reč 0^{m-n}. U ovom slučaju je $m \geqslant n$ i vrednost tražene funkcije je $f(m,n) = m-n$.

2. Ako na početku ciklusa, $\mathcal{A}$ ne može da nađe više levih nula, jer je prvih m nula promenjeno u #, to znači da je $n \geqslant m$, pa je $f(m,n) = 0$. Tada sve simbole 0 i 1 mašina menja sa #, te završava rad kada je ulazna traka prazna.

Na osnovu prethodnog, konstruišimo Turingovu mašinu $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$, u kojoj je

$$\begin{gathered} A = \{a_0, a_1, a_2, a_3, a_4, a_5, a_6\}, \\ X = \{0,1\}, \\ \Sigma = \{0,1,\#\}. \end{gathered}$$

Skup završnih stanja je $T = \{a_6\}$, a funkciju prelaza definišemo sa:

$$\begin{array}{ll}
1.a. & \delta(a_0,0) = (a_1,\#,R) \\
1.b. & \delta(a_0,1) = (a_5,\#,R) \\
2.a. & \delta(a_1,0) = (a_1,0,R) \\
2.b. & \delta(a_1,1) = (a_2,1,R) \\
3.a. & \delta(a_2,0) = (a_3,1,L) \\
3.b. & \delta(a_2,1) = (a_2,1,R) \\
3.c. & \delta(a_2,\#) = (a_4,\#,L) \\
4.a. & \delta(a_3,0) = (a_3,0,L) \\
4.b. & \delta(a_3,1) = (a_3,1,L) \\
4.c. & \delta(a_3,\#) = (a_0,\#,R) \\
5.a. & \delta(a_4,0) = (a_4,0,L) \\
5.b. & \delta(a_4,1) = (a_4,\#,L) \\
5.c. & \delta(a_4,\#) = (a_6,0,R) \\
6.a. & \delta(a_5,0) = (a_5,\#,R) \\
6.b. & \delta(a_5,1) = (a_5,\#,R) \\
6.c. & \delta(a_5,\#) = (a_6,\#,R)
\end{array}$$

Dakle, ova Turingova mašina završava rad kada je na ulaznoj traci dobijena reč $0^{f(m,n)}$, tj. traka je na kraju rada prazna za $f(m,n) = 0$. □

4.2. Turingove mašine i jezici tipa 0

Zadatak 4.168. *Neka je X proizvoljan alfabet i L jezik nad alfabetom X koji je raspoznatljiv Turingovom mašinom. Dokazati da je tada L jezik tipa* 0.

Rešenje: Pretpostavimo da Turingova mašina $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ raspoznaje jezik L skupom stanja T. Konstruisaćemo gramatiku $G = (V, X, \pi)$ koja generiše dve kopije neke reči iz Σ^* i onda simulira rad mašine $\mathcal{A}$ na jednoj od kopija. Ako $\mathcal{A}$ prihvata razmatranu reč, tada G transformiše drugu kopiju u reč iz X^*. Ako $\mathcal{A}$ ne prihvata početnu reč, tada izvođenje ne vodi do reči iz X^*. Ne umanjujući opštost dokaza, možemo pretpostaviti da je $\delta(a,\xi)$ nedefinisano za sve $a \in T$ i $\xi \in \Sigma$.

Uzmimo da je $V = \{[x,\xi] \mid x \in X \cup \{e\},\ \xi \in \Sigma\} \cup A \cup \{\sigma,\rho,\eta\}$, i neka su pravila izvođenja data sa:

1. $\sigma \to a_0\rho$,
2. $\rho \to [x,x]\rho$,
3. $\rho \to \eta$,
4. $\eta \to [e,\#]\eta$,
5. $\eta \to e$,
6. $a[x,\xi] \to [x,\mu]a'$, za sve $a \in A$ i $\xi \in \Sigma$ takve da je $\delta(a,\xi) = (a,\mu,R)$,
7. $[y,\nu]a[x,\xi] \to [y,\nu][x,\mu]a'$, za sve $\xi,\mu,\nu \in \Sigma$ i $x,y \in X \cup \{e\}$ i $a \in A$ takve da je $\delta(a,\xi) = (a',\mu,L)$,
8. $[x,\xi]a \to axa$, $a[x,\xi] \to axa$, $a \to e$, za sve $x \in X \cup \{e\}$, $\xi \in \Sigma$, $a \in T$.

Pretpostavimo da $\mathcal{A}$ prihvata reč $u = x_1x_2\dots x_n$, gde je $x_1,x_2,\dots,x_n \in X$. Koristeći pravila 1. i 2. dobijamo

$$\sigma \overset{*}{\underset{G}{\Rightarrow}} a_0[x_1,x_1][x_2,x_2]\dots[x_n,x_n]\rho.$$

Neka je $m \in \mathbb{N}$ broj ćelija koje $\mathcal{A}$ koristi za izračunavanje. Koristeći pravilo 3, a zatim m puta pravilo 4, i konačno pravilo 5, imamo

$$\sigma \overset{*}{\underset{G}{\Rightarrow}} a_0[x_1,x_1][x_2,x_2]\dots[x_n,x_n][e,\#]^m.$$

Nadalje koristimo pravila 6. i 7. sve dok ne dođemo do nekog završnog stanja. Pri tome prva komponenta u paru $[x,\xi]$, $x \in X \cup \{e\}$, $\xi \in \Sigma$, ostaje neizmenjena.

Indukcijom po dužini izračunavanja

$$(a_0,x_1x_2\dots x_n,1) \xrightarrow[\mathcal{A}]{*} (a,\xi_1\xi_2\dots\xi_s,r),$$

dokazaćemo da iz njega sledi

$$\begin{aligned} &a_0[x_1,x_1][x_2,x_2]\dots[x_n,x_n][e,\#]^m \overset{*}{\underset{G}{\Rightarrow}} \\ &\qquad \overset{*}{\underset{G}{\Rightarrow}} [x_1,\xi_1][x_2,\xi_2]\dots[x_{r-1},\xi_{r-1}]a[x_r,\xi_r]\dots[x_{n+m},\xi_{n+m}], \end{aligned}$$

gde su

$$x_1,x_2,\dots,x_n \in X,\ x_{n+1} = x_{n+2} = \dots = x_{n+m} = e,$$

$$\xi_1,\xi_2,\dots,\xi_{n+m} \in \Sigma, \quad \text{pri čemu je } \xi_{s+1} = \xi_{s+2} = \dots = \xi_{n+m} = \#.$$

Tvrđenje evidentno važi za izračunavanja dužine 0. Pretpostavimo da je tvrđenje tačno za sva izračunavanja dužine $k-1$ i uzmimo da je

$$(a_0,x_1x_2\dots x_n,1) \xrightarrow[\mathcal{A}]{*} (a,\xi_1\xi_2\dots\xi_s,r),$$

izračunavanje dužine k. To izračunavanje možemo podeliti na izračunavanje

$$(a_0, x_1x_2\ldots x_n, 1) \xrightarrow[\mathcal{A}]{*} (a', \mu_1\mu_2\ldots\mu_t, r'),$$

dužine $k-1$, i korak

$$(a', \mu_1\mu_2\ldots\mu_t, r') \xrightarrow[\mathcal{A}]{} (a, \xi_1\xi_2\ldots\mu_s, r').$$

Prema indukcijskoj hipotezi, postoji izvođenje

$$\begin{aligned} &a_0[x_1,x_1][x_2,x_2]\ldots[x_n,x_n][e,\#]^m \overset{*}{\underset{G}{\Rightarrow}} \\ &\qquad \overset{*}{\underset{G}{\Rightarrow}} [x_1,\mu_1]\ldots[x_{r'-1},\mu_{r'-1}]a'[x_{r'},\mu_{r'}]\ldots[x_{n+m},\mu_{n+m}]. \end{aligned}$$

Stavimo

$$D = \begin{cases} L, & \text{ako je } r = r'-1, \\ R, & \text{ako je } r = r'+1. \end{cases}$$

Tada svakako za neki D važi $\delta(a', \mu_{r'}) = (a, \xi_{r'}, D)$. Prema pravilima 6. ili 7. je

$$a'[x_{r'},\mu_{r'}] \xrightarrow[G]{} [x_{r'},\xi_{r'}]a$$

ili

$$[x_{r'-1},\mu_{r'-1}]a'[x_{r'},\mu_{r'}] \xrightarrow[G]{} a[x_{r'-1},\mu_{r'-1}][x_{r'},\xi_{r'}]$$

zavisno od vrednosti D. Jasno je da je $\mu_i = \xi_i$ za sve $i \neq r'$. Odatle je

$$\begin{aligned} &a_0[x_1,x_1][x_2,x_2]\ldots[x_n,x_n][e,\#]^m \overset{*}{\underset{G}{\Rightarrow}} \\ &\qquad \overset{*}{\underset{G}{\Rightarrow}} [x_1,\xi_1]\ldots[x_{r-1},\xi_{r-1}]a[x_r,\xi_r]\ldots[x_{n+m},\xi_{n+m}], \end{aligned}$$

što je i trebalo dokazati. Sada, za $a \in T$ imamo

$$[x_1,\xi_1]\ldots[x_{r-1},\xi_{r-1}]a[x_r,\xi_r]\ldots[x_{n+m},\xi_{n+m}] \overset{*}{\underset{G}{\Rightarrow}} x_1x_2\ldots x_n.$$

Dakle, $L(G,\sigma)$ sadrži sve reči koje raspoznaje mašina $\mathcal{A}$, tj. $L \subseteq L(G,\sigma)$.

Preostaje da se dokaže da $\mathcal{A}$ prihvata sve reči iz jezika $L(G,\sigma)$. Posmatrajmo izvođenje u G koje vodi do reči $u \in L \subseteq X^*$, pri čemu je $u = x_1x_2\ldots x_n$, za neke $x_1, x_2, \ldots, x_n \in X$. Početni deo tog izvođenja mora biti oblika

$$\sigma \xrightarrow[G]{} a_0\rho \xrightarrow[G]{*} a_0[x_1,x_1]\ldots[x_n,x_n].$$

Dokazaćemo indukcijom po dužini izvođenja da iz

$$\begin{aligned} &a_0[x_1,x_1][x_2,x_2]\ldots[x_n,x_n] \overset{*}{\underset{G}{\Rightarrow}} \\ &\qquad \overset{*}{\underset{G}{\Rightarrow}} [x_1,\xi_1]\ldots[x_{r-1},\xi_{r-1}]a[x_r,\xi_r]\ldots[x_n,\xi_n] \end{aligned} \tag{4.1}$$

sledi

$$(a_0, x_1x_2 \dots x_n, 1) \xrightarrow[\mathcal{A}]{*} (a, \xi_1\xi_2 \dots \xi_n, r),$$

za neke $\xi_1, \xi_2, \dots, \xi_n \in \Sigma$.

Neka je (4.1) izvođenje dužine 1, tj. neposredno izvođenje

$$a_0[x_1, x_1][x_2, x_2] \dots [x_n, x_n] \underset{G}{\Rightarrow} [x_1, \xi_1]a[x_2, x_2] \dots [x_n, x_n].$$

Ovo izvođenje je moguće samo ako je zadovoljeno $\delta(a_0, x_1) = (a, \xi_1, R)$. Tada je jasno da važi

$$(a_0, x_1x_2 \dots x_n, 1) \xrightarrow[\mathcal{A}]{} (a, \xi_1x_2 \dots x_m, 2),$$

što je i trebalo dokazati. Pretpostavimo da gornje tvrđenje važi za sva izvođenja dužine $k-1$ i dokažimo da važi i za izvođenje (4.1) dužine k. Tada se to izvođenje može podeliti na izvođenje

$$a_0[x_1, x_1][x_2, x_2] \dots [x_n, x_n] \underset{G}{\Rightarrow} [x_1, \xi'_1] \dots [x_{r'-1}, \xi'_{r'-1}]a'[x_{r'}, \xi'_{r'}] \dots [x_n, \xi'_n]$$

dužine $k-1$ i neposredno izvođenje

$$\begin{aligned}[x_1, \xi'_1] \dots [x_{r'-1}, \xi'_{r'-1}]a'[x_{r'}, \xi'_{r'}] \dots [x_n, \xi'_n] &\underset{G}{\Rightarrow} \\ \underset{G}{\Rightarrow} [x_1, \xi_1] \dots [x_{r-1}, \xi_{r-1}]a[x_r, \xi_r] \dots [x_n, \xi_n].&\end{aligned}$$

Prema indukcijskoj hipotezi je

$$(a_0, x_1x_2 \dots x_n, 1) \xrightarrow[\mathcal{A}]{*} (a, \xi'_1\xi'_2 \dots \xi'_n, r').$$

Takođe, iz poslednjeg neposrednog izvođenja sledi

$$\delta(a', x_{r'}) = \begin{cases} (a, \xi_{r'+1}, R), & \text{ako je } r = r' + 1, \\ (a, \xi_{r'-1}, L), & \text{ako je } r = r' - 1. \end{cases}$$

Odatle je

$$(a, \xi'_1\xi'_2 \dots \xi'_n, r') \rightarrow (a, \xi_1\xi_2 \dots \xi_n, r).$$

Prema tome,

$$(a_0, x_1x_2 \dots x_n, 1) \xrightarrow[\mathcal{A}]{*} (a, \xi_1\xi_2 \dots \xi_n, r),$$

što je i trebalo dokazati.

Vratimo se ponovo izvođenjima u gramatici G. Ona su sledećeg oblika

$$\begin{aligned}\sigma \xrightarrow[G]{} a_0\rho \xrightarrow[G]{*} a_0[x_1, x_1] \dots [x_n, x_n]& \\ \overset{*}{\underset{G}{\Rightarrow}} [x_1, \xi_1] \dots [x_{r-1}, \xi_{r-1}]a[x_r, \xi_r] \dots [x_n, \xi_n].&\end{aligned}$$

Izvođenje do reči iz X^* je moguće samo ako primenimo u izvesnom koraku pravilo 8, a ono je primenljivo samo ako su nas već upotrebljena izvođenja dovela do stanja $a \in T$, a onda prema već dokazanom važi

$$(a_0, x_1x_2\dots x_n, 1) \xrightarrow[\mathcal{A}]{*} (a, \xi_1\xi_2\dots\xi_n, r).$$

Jasno je da primenom pravila 8. konačan broj puta dobijamo

$$[x_1, \xi_1]\dots[x_{r-1}, \xi_{r-1}]a[x_r, \xi_r]\dots[x_n, \xi_n] \overset{*}{\underset{G}{\Rightarrow}} x_1\dots x_n.$$

Kako je $u = x_1x_2\dots x_n \in L(G, \sigma)$, to niz izvođenja koji vodi do reči iz X^* svakako postoji, i u njemu se pojavljuje i izvođenje (4.1) kod koga je $a \in T$. Odatle, prema već dokazanom važi

$$(a_0, x_1x_2\dots x_n, 1) \xrightarrow[\mathcal{A}]{*} (a, \xi_1\xi_2\dots\xi_n, r),$$

što znači da mašina $\mathcal{A}$ prihvata reč u. Dakle, $L(G, \sigma) \subseteq L$. □

Zadatak 4.169. *Neka je L jezik tipa* 0 *nad proizvoljnim alfabetom X. Tada postoji Turingova mašina koja raspoznaje jezik L. Dokazati.*

Rešenje: Neka je $L = L(G, \sigma)$, gde je $G = (V, X, \pi)$ gramatika tipa 0.

Konstruisaćemo nedeterminističku Turingovu mašinu $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ koja raspoznaje L. Uzmimo $\Sigma = V \cup \{\#, \rho, \mu\}$, pri čemu $\#, \rho, \mu$ ne pripadaju alfabetu V. Smatraćemo, takođe, da $\mathcal{A}$ može da upisuje prazan simbol #.

Opisaćemo u grubim crtama rad mašine $\mathcal{A}$.

Na početku se na ulaznoj traci mašine nalazi reč $w \in X^*$. Mašina $\mathcal{A}$ upisuje ρ na prvom mestu trake i pomera reč w jedno mesto udesno. Zatim iza reči w dopisuje $\rho\sigma\rho$. Dakle, neprazan deo trake je reč $\rho w \rho\sigma\rho$. Sada $\mathcal{A}$ simulira izvođenja u G na drugom delu trake, tj. ako je

$$\sigma \to u_1 \Rightarrow u_2 \Rightarrow \dots \Rightarrow u_{n-1} \Rightarrow u_n$$

izvođenje u G, tada $\mathcal{A}$ menja σ u u_1, zatim u_1 u u_2, ..., zatim u_{n-1} u u_n.

Naime, ako je sadržaj trake oblika

$$\rho w \rho \xi_1\xi_2\dots\xi_k\rho,$$

tada $\mathcal{A}$ bira reč

$$\xi_i\xi_{i+1}\dots\xi_{i+r-1}$$

takvu da postoji pravilo $\xi_i\xi_{i+1}\dots\xi_{i+r-1} \to \alpha$ u π, gde je α neka reč iz V^*, i menja podreč $\xi_i\xi_{i+1}\dots\xi_{i+r-1}$ sa α. Pri tome se po potrebi vrši pomeranje reči $\xi_{i+r}\dots\xi_k$ ulevo ili udesno kako bi se napravilo dovoljno prostora za reč α ako ona nije dužine r. Na kraju dolazimo do reči $u_n \in V^*$ takve da nijedna njena podreč nije leva strana nekog izvođenja iz π. Sadržaj trake je $\rho w \rho u_n \rho$.

Mašina prihvata reč w ako je $u_n = w$, tj. ako postoji izvođenje $\sigma \overset{*}{\Rightarrow} w$, odakle je jasno da $\mathcal{A}$ raspoznaje L. □

Zadaci za samostalni rad

1. Konstruisati Turingovu mašinu $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ koja raspoznaje jezik $L = \{u \in \{0,1\}^*, | u$ se završava sa 00$\}$.
2. Konstruisati Turingovu mašinu $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ koja raspoznaje jezik $L = \{u \in \{0,1\}^*, | u$ sadrži bar jednu 0 i jednu 1$\}$.
3. Konstruisati Turingovu mašinu koja broji jedinice u ulaznoj binarnoj reči i na poslednjem mestu ulazne trake upisuje 1 ako je njihov broj paran i 0, ako je broj jedinica neparan.
4. Konstruisati Turingovu mašinu $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ koja raspoznaje jezik $L = \{x^{2^n} \in \{x\}^*, | n \geqslant 0\}$.
5. Konstruisati Turingovu mašinu $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ koja raspoznaje jezik $L = \{x^n y^{2n} \in \{x, y\}^*, | n \geqslant 0\}$.
6. Konstruisati Turingovu mašinu $\mathcal{A} = (A, a_0, X, \Sigma, \delta)$ koja raspoznaje jezik $L = \{x^{|u|_x} y^{|u|_y} \,|\, u \in \{x, y\}^*\}$.

Glava 5
Automati sa izlazom

5.1. Predstavljanje i konstrukcija automata sa izlazom

Automati sa izlazom predstavljaju matematičku apstrakciju mašine koja radi u diskretnoj vremenskoj skali i koja tokom tog rada, pod uticajem ulaznih signala, menja svoja unutrašnja stanja i emituje odgovarajuće izlazne signale. O stanjima mašine razmišljamo kao o nekim njenim unutrašnjim atributima, koji, zajedno sa ulazom, određuju izlaz u datom trenutku.

Kada u ovoj glavi budemo govorili *automat*, mislićemo na pojam *Mealyevog automata*, ili *automata Mealyevog tipa*, koji se definiše kao uređena petorka $\mathcal{A} = (A, X, Y, \delta, \lambda)$ za koju važi:

– A je neprazan skup koji nazivamo *skup stanja* automata $\mathcal{A}$;
– X je neprazan skup koji nazivamo *skup ulaza* (*ulaznih signala*, *ulaznih simbola*) automata $\mathcal{A}$;
– Y je neprazan skup koji nazivamo *skup izlaza* (*izlaznih signala*, *izlaznih simbola*) automata $\mathcal{A}$;
– $\delta : A \times X \to A$ je preslikavanje koje nazivamo *funkcija prelaza* (*funkcija narednog stanja*) automata $\mathcal{A}$;
– $\lambda : A \times X \to Y$ je preslikavanje koje nazivamo *funkcija izlaza* automata $\mathcal{A}$.

Princip rada ovako definisanog automata možemo shvatiti na sledeći način: Automat $\mathcal{A}$ se u određenom trenutku nalazi u stanju $a \in A$, a na njegov ulaz dospeva ulazni signal $x \in X$. Pod dejstvom tog signala automat menja stanje i u sledećem trenutku prelazi u stanje $\delta(a, x) \in A$, i istovremeno se na izlaz automata šalje izlazni signal $\lambda(a, x) \in Y$.

Specijalizacijom ovako definisanog pojma automata dobijamo razne druge zanimljive tipove automata. Ako automat $\mathcal{A} = (A, X, Y, \delta, \lambda)$ zadovoljava uslov

$$\delta(a, x) = \delta(a', x') \;\Rightarrow\; \lambda(a, x) = \lambda(a', x'),$$

za sve $a, a' \in A$, $x, x' \in X$, što je ekvivalentno uslovu da postoji preslikavanje $\mu : A \to Y$ takvo da se preslikavanje λ može izraziti preko μ i δ sa

$$\lambda(a, x) = \mu(\delta(a, x)),$$

za sve $a \in A$, $x \in X$, tada $\mathcal{A}$ zovemo *Mooreov automat*, ili *automat Mooreovog tipa*, i pišemo $\mathcal{A} = (A, X, Y, \delta, \mu)$. Preslikavanje μ nazivamo *funkcija znaka*, a $\mu(a)$ nazivamo *znak* stanja $a \in A$. Razlika između Mealyevog i Mooreovog automata leži u tome da se kod Mealyevog automata istovremeno vrši prelazak u naredno stanje i šalje izlazni signal, dok se kod Mooreovog automata najpre vrši prelaz u naredno stanje, a tek onda šalje izlazni signal koji zavisi samo od stanja u koje je automat prešao (taj signal je znak tog stanja), dok ne zavisi direktno od ulaznog signala. Drugim rečima, zavisnost izlaznog signala od ulaznog je posredna i ispoljava se samo kroz njegov uticaj na promenu stanja.

Ako posebno ističemo stanje iz koga automat uvek započinje svoj rad govorimo o *inicijalnom* (*Mealyevom*) *automatu* koji definišemo kao uređenu šestorku $\mathcal{A} = (A, a_0, X, Y, \delta, \lambda)$, gde je $a_0 \in A$ je fiksirano stanje koje nazivamo *početno* (*inicijalno*) *stanje*. Slično definišemo *inicijalni Mooreov automat*, u oznaci $\mathcal{A} = (A, a_0, X, Y, \delta, \mu)$.

Ako su skupovi stanja, ulaza i izlaza automata konačni, tada ga nazivamo *konačan automat*.

Neka je dat Mealyev automat $\mathcal{A} = (A, X, Y, \delta, \lambda)$. Tada se funkcije prelaza i izlaza mogu proširiti redom do preslikavanja $\delta : A \times X^* \to A^*$ i $\lambda : A \times X^* \to Y^*$ na sledeći način: Za $a \in A$ i $u \in X^+$, $u = x_1 x_2 \cdots x_n$, $x_1, x_2, \ldots, x_n \in X$, stavljamo da je

$$\delta(a,u) = a_1 a_2 \cdots a_n, \tag{5.1}$$

gde je

$$a_1 = \delta(a, x_1),\ a_2 = \delta(a_1, x_2), \ldots,\ a_n = \delta(a_{n-1}, x_n), \tag{5.2}$$

i

$$\lambda(a,u) = y_1 y_2 \cdots y_n, \tag{5.3}$$

pri čemu je

$$y_1 = \lambda(a, x_1),\ y_2 = \lambda(a_1, x_2), \ldots,\ y_n = \lambda(a_{n-1}, x_n). \tag{5.4}$$

Takođe

$$\delta(a,e) = a, \qquad \lambda(a,e) = e, \tag{5.5}$$

gde smo prazne reči i u X^* i u Y^* označili istim slovom e.

Skup X nazivamo *ulazni alfabet*, polugrupu X^+ i monoid X^* nazivamo *ulazna polugrupa* i *ulazni monoid* a njihove elemente *ulazne reči* automata $\mathcal{A}$, dok skup Y nazivamo *izlazni alfabet*, polugrupu Y^+ i monoid Y^* *izlazna polugrupa* i *izlazni monoid*, a njihove elemente *izlazne reči* automata $\mathcal{A}$. Princip rada Mealyevog automata sada možemo prikazati i na sledeći način: na ulaz automata dolaze jedan za drugim ulazni signali $x_1, x_2, \ldots, x_n \in X$, tj. ulazna reč $x_1 x_2 \cdots x_n \in X^+$, i pod njenim uticajem automat $\mathcal{A}$ prelazi iz stanja a u stanje a_n preko niza *međustanja* $a_1, a_2, \ldots, a_{n-1}$, a na izlaz se jedan za drugim šalju izlazni signali $y_1, y_2, \ldots, y_n \in Y$, odnosno izlazna reč $y_1 y_2 \cdots y_n \in Y^+$. Naravno, ako na ulaz automata dospe prazna reč, tada automat ostaje u istom stanju i nema izlaznog signala (na izlaz se šalje prazna reč).

Zadnje stanje a_n u nizu datom u (5.1), odnosno (5.2), označavamo sa au. Ovom oznakom zadato je preslikavanje

$$(a,u) \mapsto au \tag{5.6}$$

koje slika $A \times X^*$ u A, koje je takođe proširenje funkcije prelaza. Primetimo da preslikavanje δ definisano sa (5.1) daje više informacija o radu automata nego preslikavanje definisano sa (5.6). Naime, preslikavanjem $(a,u) \mapsto au$ određeno je samo zadnje stanje au u koje se iz stanja a dospeva pod uticajem ulazne reči u, dok je preslikavanjem δ određen i niz međustanja preko kojih se stiže iz stanja a u stanje au. Sa druge strane, u slučaju kada nam taj niz međustanja nije bitan, zbog jednostavnijeg pisanja radije koristimo drugo preslikavanje.

Lako se dokazuje da važi sledeća lema.

Lema 5.1. *Neka je dat automat $\mathcal{A} = (A,X,Y,\delta,\lambda)$. Tada za proizvoljno stanje $a \in A$ i reči $u,v \in X^*$ važi:*

(a) $\delta(a,uv) = \delta(a,u)\delta(au,v)$;
(b) $\lambda(a,uv) = \lambda(a,u)\lambda(au,v)$;
(c) $a(uv) = (au)v$.

Automati sa izlazom predstavljaju se na isti način kao i automati bez izlaza. Konačne automate je takođe veoma zgodno zadavati tablicama. *Prelazno–izlazna tablica* Mealyevog automata $\mathcal{A} = (A,X,Y,\delta,\lambda)$ je tablica sa vrstama koje odgovaraju svakom ulaznom simbolu i kolonama koje odgovaraju stanjima. Na mestu u tablici koje odgovara vrsti određenoj ulaznim simbolom $x \in X$ i koloni određenoj stanjem $a \in A$ upisuje se uređeni par $(\delta(a,x),\lambda(a,x))$. To je prikazano u sledećoj tablici

$$\begin{array}{c|ccc} A & \dots & a & \dots \\ \hline \vdots & & \vdots & \\ x & \dots & (\delta(a,x),\lambda(a,x)) & \dots \\ \vdots & & \vdots & \end{array}$$

Drugi pogodan način zadavanja automata, kako smo videli kod automata bez izlaza, jeste njihovo zadavanje pomoću grafova. Neka je dat Mealyev automat $\mathcal{A} = (A,X,Y,\delta,\lambda)$. *Prelazno-izlazni graf* automata $\mathcal{A}$ je označeni graf čiji skup čvorova je skup stanja A, a grana između dva čvora $a,b \in A$ je označena uređenim parom (x,y), ako se iz stanja $a \in A$ pod uticajem ulaznog signala $x \in X$ prelazi u stanje $b (= \delta(a,x) \in A)$, pri čemu se emituje izlazni signal $y (= \lambda(a,x) \in Y)$.

Pri zadavanju Mooreovih automata, umesto parova $(\delta(a,x),\lambda(a,x))$ u tablicu se upisuje samo $\delta(a,x)$, dok se preslikavanje μ zadaje tako što se iznad vrste u kojoj su poređana stanja dodaje još jedna vrsta u koju se upisuju njihovi znakovi, pri čemu se znak $\mu(a)$ stanja $a \in A$ piše upravo iznad a. To je prikazano u sledećoj tablici:

$$\begin{array}{c|ccc} & \dots & \mu(a) & \dots \\ \cline{2-4} A & \dots & a & \dots \\ \hline \vdots & & \vdots & \\ x & \dots & \delta(a,x) & \dots \\ \vdots & & \vdots & \end{array}$$

Prelazno-izlazni graf Mooreovog automata $\mathcal{A} = (A, X, Y, \delta, \mu)$ ima nešto drugačiji izgled nego graf Mealyevog automata. Naime, kod grafa Mooreovog automata čvorovi su označeni uređenim parovima oblika (a, y), gde je $a \in A$, $y \in Y$ i $\mu(a) = y$, a grane su označene samo odgovarajućim ulaznim simbolima.

Zadatak 5.170. *Neka je dat automat $\mathcal{A} = (A, X, Y, \delta, \lambda)$, gde je*

$$A = \{a, b, c\}, \quad X = \{x, x', x''\} \quad i \quad Y = \{y, y'\},$$

i funkcije prelaza i izlaza su definisane sa:

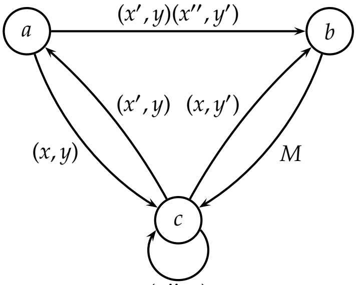
$$\delta(a, x) = \delta(b, x) = \delta(b, x') = \delta(b, x'') = \delta(c, x'') = c,$$
$$\delta(a, x') = \delta(a, x'') = \delta(c, x) = b, \ \delta(c, x') = a$$
$$\lambda(a, x) = \lambda(a, x') = \lambda(b, x) = \lambda(c, x') = \lambda(c, x'') = y$$
$$\lambda(a, x'') = \lambda(b, x') = \lambda(b, x'') = \lambda(c, x) = y'.$$

Predstaviti automat $\mathcal{A}$ prelazno-izlaznom tablicom i prelazno-izlaznim grafom.

Rešenje: Prelazno-izlazna tablica ovog automata je sledeća

A	a	b	c
x	(c, y)	(c, y)	(b, y')
x'	(b, y)	(c, y')	(a, y)
x''	(b, y')	(c, y')	(c, y)

dok je njegov prelazno-izlazni graf dat sa



gde je $M = \{(x, y), (x', y'), (x'', y')\}$. □

Zadatak 5.171. *Neka je dat Mooreov automat $\mathcal{A} = (A, X, Y, \delta, \mu)$, gde je*

$$A = \{a_1, a_2, a_3\}, \quad X = \{x_1, x_2\}, \quad Y = \{y_1, y_2\},$$

i funkcije δ i μ su zadate sa:

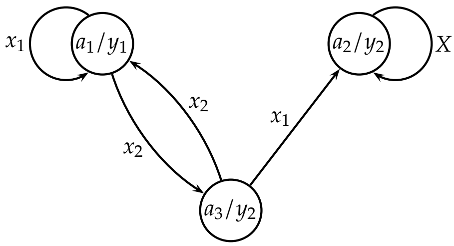
$$\delta(a_1, x_1) = \delta(a_3, x_2) = a_1, \quad \delta(a_1, x_2) = a_3,$$
$$\delta(a_2, x_1) = \delta(a_2, x_2) = \delta(a_3, x_1) = a_2$$
$$\mu(a_1) = y_1, \quad \mu(a_2) = \mu(a_3) = y_2.$$

Predstaviti automat $\mathcal{A}$ prelazno-izlaznom tablicom i prelazno-izlaznim grafom.

Rešenje: Ovaj automat zadaje se tablicom

A	y_1	y_2	y_2
	a_1	a_2	a_3
x_1	a_1	a_2	a_2
x_2	a_3	a_2	a_1

i može se predstaviti sledećim prelazno-izlaznim grafom

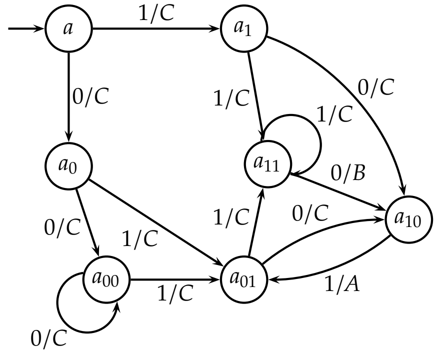


odakle jasno vidimo vrednosti funkcije μ. □

Zadatak 5.172. *Konstruisati Mealyev automat sa ulazima* $\{0,1\}$ *i izlazima* $\{A,B,C\}$ *takav da se na izlazu javlja A ako se ulazna reč završava sa* 101*, B ako se ulazna reč završava sa* 110 *i C inače.*

Rešenje: Prema uslovima zadatka jasno je da izlaz zavisi samo od sufiksa ulazne reči, koji ima dužinu tri. Dakle, za slova i ulazne reči dužine dva, na izlazu se uvek javlja C. Kada su ulazne reči dužine ne manje od tri, bitna su nam samo poslednja dva slova, jer učitavanjem narednog slova dobijamo sufiks dužine tri.

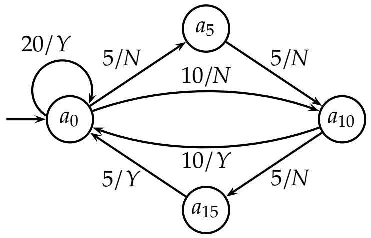
Pri konstrukciji traženog automata indeksiraćemo njegova stanja nulom, jedinicom ili binarnim rečima dužine dva, koje predstavljaju dva poslednja slova ulazne reči koja se čita slovo po slovo. Slovo koje se sledeće učitava je prva koordinata u oznaci grane, dok je druga koordineta izlazno slovo.



Prema tome, datim prelazno-izlaznim grafom je predstavljen traženi Mealyev automat. □

Zadatak 5.173. *Konstruisati automat koji prihvata novčanice od 5,10 i 20 dinara i na izlazu pise Y (YES) kada je ubačena suma novca deljiva sa 20 i N (NO) za sumu koja nije deljiva sa 20.*

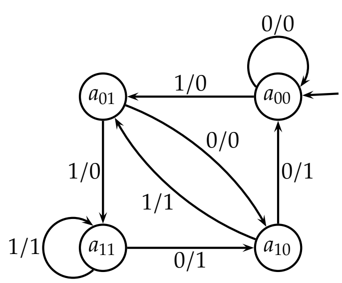
Rešenje: Neka je graf prelaza traženog automata predstavljen na sledeći način:



Oznake stanja indeksirane su sumom ubačenih novčanica, a vrednost ovih novčanica su prve koordinate u oznaci grana. □

Zadatak 5.174. *Konstruisati Mealyev automat čiji je ulazni i izlazni alfabet* $\{0,1\}$ *i koji na izlazu daje ulazni signal učitan dva koraka ranije.*

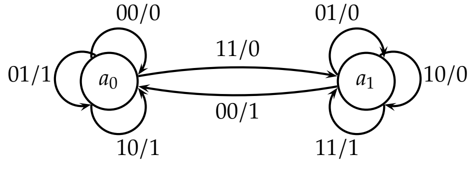
Rešenje: Graf prelaza traženog automata je:



Stanja automata smo označili sa a_{ij}, pri čemu i i j predstavljaju dva prethodna signala. □

Zadatak 5.175. *Konstruisati automat koji sabira dva broja zadata u binarnom sistemu, tako što se cifre učitavaju naizmenično počev od poslednje, a dužine brojeva se izjednačavaju dodavanjem odgovarajućeg broja nula na mesta najveće težine.*

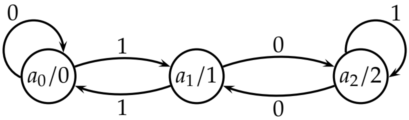
Rešenje: Stanja ovog automata označićemo sa a_i, gde i označava binarnu vrednost koja se, pri narednom unosu, prenosi na mesto veće težine.



Na taj način smo dobili graf automata koji sabira dva binarna broja. □

Zadatak 5.176. *Konstruisati Mooreov automat koji određuje ostatak pri deljenju broja u binarnom zapisu sa 3.*

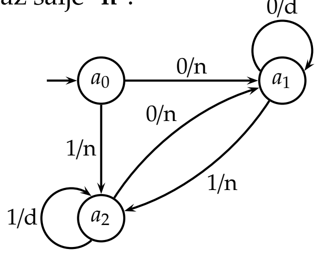
Rešenje: Neka je dat broj x u binarnom zapisu. Ako u datom binarnom zapisu dodamo nulu na mesto najmanje težine dobijamo binarni zapis broja $y = 2x$, što znači da je $y \equiv 2x\,(mod 3)$. Dodavanjem jedinice na mesto najmanje težine u binarnom zapisu broja x, dobijamo binarni oblik broja $z = 2x + 1$, te je, jasno $z \equiv 2x + 1\,(mod 3)$. Mooreov automat učitava binarni broj cifru po cifru, smeštajući je na mesto najmanje težine. Dakle, ulazni alfabet je $X = \{0,1\}$, a izlazni $Y = \{0,1,2\}$. Graf prelaza ovog automata može se predstaviti na sledeći način:



Primećujemo da indeks u oznaci stanja predstavlja ostatatak pri deljenju datog broja sa 3. □

Zadatak 5.177. *Konstruisati Mealyev automat koji daje niz izlaznih signala sastavljen od "***d***" i "***n***", tako da za svaki učitan prefiks ulazne reči šalje "***d***" ako je taj prefiks iz skupa* $L = (0+1)^*(00+11)$, *dok inače šalje "***n***".*

Rešenje: Primetimo da traženi Mealyev automat šalje izlazni signal "**d**", ako ulazna reč sadrži bar dve uzastopne nule ili bar dve uzastopne jedinice, dok u protivnom na izlaz šalje "**n**".



Dakle, traženi automat ima tri stanja i predstavljen je datim grafom. □

Zadatak 5.178. *Da li je automat iz Zadatka 5.174. automat Mooreovog tipa?*

Rešenje: Na osnovu grafa prelaza, predstavljenog u Zadatku 5.174. formiraćemo i tablicu prelaza tog automata:

A	a_{00}	a_{01}	a_{10}	a_{11}
0	$(a_{00},0)$	$(a_{10},0)$	$(a_{00},1)$	$(a_{10},1)$
1	$(a_{01},0)$	$(a_{11},0)$	$(a_{01},1)$	$(a_{11},1)$

Odavde vidimo da je

$$\delta(a_{00},0) = \delta(a_{10},0) = a_{00} \quad \text{i} \quad \lambda(a_{00},0) = 0 \neq 1 = \lambda(a_{10},0),$$

te je jasno da automat nije Mooreov. □

5.2. Preslikavanja indukovana automatima

Svakom stanju a automata $\mathcal{A} = (A, X, Y, \delta, \lambda)$ možemo pridružiti preslikavanje $\phi_a : X^* \to Y^*$ definisano sa

$$\phi_a(u) = \lambda(a, u), \quad \text{za } u \in X^*,$$

koje nazivamo *preslikavanje indukovano stanjem a automata* $\mathcal{A}$. Ako je dat inicijalni automat $\mathcal{A} = (A, a_0, X, Y, \delta, \lambda)$, tada preslikavanje ϕ_{a_0} indukovano inicijalnim stanjem a_0 automata A nazivamo *preslikavanje indukovano inicijalnim* (*Mealyevim*) *automatom*.

Štaviše, ako su dati slobodni monoidi X^* i Y^* i preslikavanje $\phi : X^* \to Y^*$, tada za ϕ kažemo da može biti indukovano inicijalnim Mealyevim automatom ako postoji neki inicijalni Mealyev automat $\mathcal{A} = (A, a_0, X, Y, \delta, \lambda)$ takav da je $\phi = \phi_{a_0}$, a za automat $\mathcal{A}$ kažemo da *predstavlja* ili *realizuje* preslikavanje ϕ.

Šta praktično predstavljaju preslikavanja indukovana automatima? Uzmimo da automat $\mathcal{A}$ počne sa radom iz stanja a. Kao rezultat njegovog rada imamo da se ulaznim rečima pridružuju odgovarajuće izlazne reči, i to pridruživanje je određeno upravo preslikavanjem ϕ_a.

Preslikavanje ϕ iz slobodnog monoida X^* u slobodni monoid Y^* nazivamo *automatovnim preslikavanjem* ako zadovoljava sledeće uslove:

(A1) ϕ očuvava dužinu reči, tj. $|\phi(u)| = |u|$, za svaki $u \in X^*$;

(A2) svaki prefiks proizvoljne reči $u \in X^*$ se preslikavanjem ϕ slika u prefiks reči $\phi(u)$.

Potsetimo da, za reč u i prirodan broj $k \leqslant |u|$, sa $r_k(u)$ označavamo *sufiks* reči u dužine k.

Zadatak 5.179. *Neka je $\phi : X^* \to Y^*$ automatovno preslikavanje. Svakoj reči $u \in X^*$ pridružimo preslikavanje $\phi_u : X^* \to Y^*$ definisano sa*

$$\phi_u(v) = r_{|v|}(\phi(uv)), \quad \text{za } v \in X^*. \tag{5.7}$$

Dokazati da tada važi:

(a) *Za proizvoljne $u, v \in X^*$, $\phi_u(v)$ je jedinstveno rešenje jednačine*

$$\phi(uv) = (\phi(u))w \tag{5.8}$$

u Y^*, po promenljivoj w.

(b) *ϕ_u je automatovno preslikavanje, za svaki $u \in X^*$, i $\phi_e = \phi$.*
(c) *$\phi_{uv} = (\phi_u)_v$, za sve $u, v \in X^*$.*

Rešenje: (a) Kako je u prefiks od uv, to prema osobini (A1) automatovnih preslikavanja dobijamo da je $\phi(u)$ prefiks od $\phi(uv)$, tj. da je $\phi(uv) = (\phi(u))w$, za neki $w \in Y^*$. Prema tome, jednačina (5.8) ima rešenje u Y^* i to rešenje je jedinstveno. Konačno, zbog toga što ϕ očuvava dužinu reči, imamo

$$|\phi(uv)| = |uv| = |u| + |v| \qquad \text{i} \qquad |(\phi(u))w| = |\phi(u)| + |w| = |u| + |w|,$$

odakle sledi $|v| = |w|$. Prema tome,

$$w = r_{|v|}(\phi(uv)) = \phi_u(v).$$

(b) Uzmimo proizvoljan $u \in X^*$. Iz (5.7) se jasno vidi da ϕ_u očuvava dužinu reči. Neka su $v, v' \in X^*$ reči takve da je v' prefiks od v, tj. $v = v'v''$, za neki $v'' \in X^*$. Tada prema (a) imamo da je

$$\phi(uv) = \phi(uv'v'') = (\phi(uv'))(\phi_{uv'}(v'')) = (\phi(u))(\phi_u(v'))(\phi_{uv'}(v'')), \tag{5.9}$$

odakle zbog jedinstvenosti rešenja jednačine (5.8) sledi

$$\phi_u(v) = (\phi_u(v'))(\phi_{uv'}(v'')).$$

Prema tome, $\phi_u(v')$ je prefiks od $\phi_u(v)$, što je i trebalo dokazati. Ovim smo dokazali da je ϕ_u automatovno preslikavanje.

Dalje, ako u (5.7) stavimo da je $u = e$, onda neposredno sledi da je $\phi_e = \phi$.

(c) Za proizvoljan $w \in X^*$ je

$$\phi(uvw) = (\phi(u))(\phi_u(vw)) = (\phi(u))(\phi_u(v))((\phi_u)_v(w)) = (\phi(uv))((\phi_u)_v(w)),$$

pa zbog (a) dobijamo da je

$$\phi_{uv}(w) = (\phi_u)_v(w).$$

Prema tome, važi (c). □

Zadatak 5.180. *Da li je funkcija $\phi : X^* \mapsto Y^*$, gde je $X = \{0,1\}$, data sa*

$$\phi(x_1x_2\dots x_{2k}) = x_1x_1x_2x_2\dots x_kx_k = x_1^2x_2^2\dots x_k^2,$$
$$\phi(x_1x_2\dots x_{2k}x_{2k+1}) = x_1x_1x_2x_2\dots x_kx_kx_{k+1} = x_1^2x_2^2\dots x_k^2x_{k+1},$$

automatovno preslikavanje?

Rešenje: Očigledno je da funkcija ϕ očuvava dužinu reči, t.j. $|\phi(u)| = |u|$, za svaki $u \in X^*$, pa je uslov (A1) zadovoljen. Ostaje da proverimo da li preslikavanje ϕ očuvava prefikse. Neka je $x_1 \dots x_i$, za $i \geqslant 1$, prefiks reči $u = x_1 \dots x_n$. Razlikujemo dva slučaja:

(i) Ako je $i = 2k$, $k \in \mathbb{N}$ imamo da je $\phi(x_1x_2\dots x_i) = x_1^2x_2^2\dots x_k^2$, a to je prefiks reči $\phi(u)$ dužine i, te je uslov (A2) zadovoljen.

(ii) Za $i = 2k+1$, $k \in \mathbb{N}$ je $\phi(x_1x_2\dots x_{2k}x_{2k+1}) = x_1^2x_2^2\dots x_k^2x_{k+1}$ prefiks reči $\phi(u)$ dužine i, što znači da (A2) važi.

Ovim smo dokazali da je ϕ automatovno preslikavanje. □

Zadatak 5.181. *Dati su skupovi $V = \{x,y\}$, $W = \{0,1\}$ i preslikavanje $f : V^* \mapsto W$ na sledeći način:*

$$f(u) = \begin{cases} 1, & |u|_x = |u|_y \\ 0, & |u|_x \neq |u|_y \end{cases}, \quad za\ u \in V^*.$$

Definišimo preslikavanje $\alpha : V^* \mapsto W^*$ sa $\alpha(a_1a_2\dots a_n) = f_1f_2\dots f_n$, za proizvoljne $a_i \in V$, pri čemu je $f_i = f(a_1a_2\dots a_i)$, $i \in \{1,2\dots n\}$. Da li je α automatovno preslikavanje?

Rešenje: Očito je da preslikavanja f_i uzimaju vrednosti u skupu W, za svaki $i \in \{1,2\dots n\}$, te za svaku reč $u = a_1a_2\dots a_n \in V^*$ važi $n = |f_1f_2\dots f_n| = |\alpha(u)|$. Kako je $\alpha(a_1a_2\dots a_i) = f(a_1)f(a_1a_2)\dots f(a_1a_2\dots a_i)$, jasno je da α očuvava prefikse, pa α jeste automatovno preslikavanje. □

Zadatak 5.182. *Preslikavanje ϕ iz slobodnog monoida X^* u slobodni monoid Y^* može biti indukovano inicijalnim Mealyevim automatom ako i samo ako je automatovno preslikavanje.*

Rešenje: Neka je preslikavanje ϕ indukovano inicijalnim Mealyevim automatom $\mathcal{A} = (A,a_0,X,Y,\delta,\lambda)$, tj. $\phi(u) = \lambda(a_0,u)$, za svaki $u \in X^*$. Iz definicije proširenih funkcija prelaza i izlaza se jasno vidi da ϕ očuvava dužinu reči.

Uzmimo proizvoljnu reč $u = x_1x_2\cdots x_n$, gde su $x_1,x_2,\dots,x_n \in X$. Prema definiciji proširenih funkcija prelaza i izlaza imamo

$$\delta(a_0,u) = a_1a_2\cdots a_n,$$
$$\lambda(a_0,u) = y_1y_2\cdots y_n,$$

pri čemu je

$$a_1 = \delta(a_0,x_1),\ a_2 = \delta(a_1,x_2),\ \dots,\ a_n = \delta(a_{n-1},x_n),$$
$$y_1 = \lambda(a_0,x_1),\ y_2 = \lambda(a_1,x_2)\ \dots,\ y_n = \lambda(a_{n-1},x_n),$$

što znači da je $\phi(u) = y_1y_2\cdots y_n$. Sa druge strane, proizvoljan prefiks v reči u je oblika $v = x_1\cdots x_i$, za neki $i \in \{1,2,\dots,n\}$, pa iz prethodnog sledi da je

$$\phi(v) = \lambda(a_0,v) = \lambda(a_0,x_1\cdots x_i) = y_1\cdots y_i,$$

te je $\phi(v)$ prefiks reči $\phi(u)$. Time je dokazano da je ϕ automatovno preslikavanje.

Obratno, neka je ϕ dato automatovno preslikavanje. Definišimo inicijalni Mealyev automat $\mathcal{A}^\phi$ na sledeći način: $\mathcal{A}^\phi = (X^*,e,X,Y,\delta^\phi,\lambda^\phi)$, pri čemu su preslikavanja δ^ϕ i λ^ϕ definisana sa:

$$\begin{aligned} \delta^\phi(u,x) &= ux \\ \lambda^\phi(u,x) &= x\phi_u \end{aligned} \qquad (u \in X^*, x \in X).$$

Da bi dokazali da je preslikavanje ϕ indukovano automatom $\mathcal{A}^{\phi}$, treba dokazati da za proizvoljnu reč $u \in X^*$ važi

$$\phi(u) = \lambda^{\phi}(e,u).$$

To ćemo dokazati indukcijom po dužini reči u. Jasno je da to važi za reči dužine 0 i 1. Pretpostavimo da je $\phi(u) = \lambda^{\phi}(e,u)$, za sve reči u dužine n i dokažimo da jednakost važi i za reči dužine $n+1$. Uzmimo proizvoljnu reč $u \in X^*$, takvu da je $u = x_1x_2\dots x_{n+1}$, za neke $x_1, x_2, \cdots, x_{n+1} \in A$. Sa u' označimo reč $x_1x_2\cdots x_n$. Prema prethodnom zadatku važi $\phi(u) = (\phi(u'))(\phi_{u'}(x_{n+1}))$. Dalje, prema indukcijskoj hipotezi je $\phi(u') = \lambda^{\phi}(e,u')$ i $\phi_{u'}(x_{n+1}) = \lambda^{\phi}(u', x_{n+1})$. Dakle, ostaje da dokažemo da je

$$\lambda^{\phi}(e,u')\lambda^{\phi}(u',x_{n+1}) = \lambda^{\phi}(e,u).$$

Zaista, prema definiciji proširenih funkcija prelaza i izlaza imamo da je

$$\begin{aligned}\delta^{\phi}(e,u) &= a_1a_2\cdots a_{n+1},\\ \lambda^{\phi}(e,u) &= y_1y_2\cdots y_{n+1},\end{aligned}$$

gde je

$$\begin{aligned}&a_1 = \delta^{\phi}(e,x_1),\ a_2 = \delta^{\phi}(a_1,x_2),\ \dots,\ a_n = \delta^{\phi}(a_{n-1},x_n),\ a_{n+1} = \delta^{\phi}(a_n,x_{n+1}),\\ &y_1 = \lambda^{\phi}(e,x_1),\ y_2 = \lambda^{\phi}(a_1,x_2)\ \dots,\ y_n = \lambda^{\phi}(a_{n-1},x_n),\ y_{n+1} = \lambda^{\phi}(a_n,x_{n+1}).\end{aligned}$$

Na isti način dobijamo da je

$$\begin{aligned}\delta^{\phi}(e,u') &= a_1a_2\cdots a_n,\\ \lambda^{\phi}(e,u') &= y_1y_2\cdots y_n.\end{aligned}$$

Sa druge strane, imamo da je $a_1 = \delta^{\phi}(e,x_1) = ex_1 = x_1$, $a_2 = \delta^{\phi}(a_1,x_2) = x_1x_2$, itd., čime dobijamo da je $a_n = x_1x_2\cdots x_n = u'$. Sada je

$$y_{n+1} = \lambda^{\phi}(a_n,x_{n+1}) = \lambda^{\phi}(u',x_{n+1}),$$

pa dobijamo $\phi(u) = \lambda^{\phi}(e,u)$. Ovim je dokaz kompletiran. □

Zadatak 5.183. *Neka je $\phi : X^* \to Y^*$ automatovno preslikavanje i neka je skup $A_\phi = \{\phi_u \,|\, u \in X^*\}$. Tada su sa*

$$\begin{aligned}\delta_\phi(\phi_u,x) &= \phi_{ux}\\ \lambda_\phi(\phi_u,x) &= x\phi_u\end{aligned} \qquad (u \in X^*, x \in X)$$

definisana preslikavanja $\delta_\phi : A_\phi \times X \to A_\phi$ i $\lambda_\phi : A_\phi \times X \to Y$ i $\mathcal{A}_\phi = (A_\phi, X, Y, \delta_\phi, \lambda_\phi)$ je automat.

Osim toga, za sve $u \in X^*$ i $v = x_1x_2\cdots x_n$, $x_1, x_2, \dots, x_n \in X$, važi:

$$\delta_\phi(\phi_u, v) = \phi_{ux_1}\phi_{ux_1x_2}\cdots\phi_{ux_1x_2\cdots x_n}; \tag{5.10}$$
$$(\phi_u)v = \phi_{uv}; \tag{5.11}$$
$$\lambda(\phi_u, v) = \phi_u(v). \tag{5.12}$$

Dokazati.

Rešenje: Najpre treba dokazati da su preslikavanja δ_ϕ i λ_ϕ dobro definisana. Pre svega, treba dokazati da za $u, v \in X^*$ i proizvoljan $x \in X$ važi:

$$\phi_u = \phi_v \;\Rightarrow\; \phi_{ux} = \phi_{vx};$$
$$\phi_u = \phi_v \;\Rightarrow\; \phi_u(x) = \phi_v(x).$$

Zaista, prva implikacija je neposredna posledica Zadatka 5.179. (c), jer jednakost $\phi_u = \phi_v$ povlači $\phi_{ux} = (\phi_u)_x = (\phi_v)_x = \phi_{vx}$, dok je druga implikacija jasna. Prema tome, δ_ϕ i λ_ϕ su dobro definisana preslikavanja. Takođe, λ_ϕ zaista slika $A_\phi \times X$ u Y. Naime, za proizvoljne $u \in X^*$, $x \in X$ imamo da je $|\phi_u(x)| = 1$, jer ϕ_u očuvava dužinu reči, pa je dakle $\phi_u(x) \in Y$. Ovim smo dokazali da su preslikavanja δ_ϕ i λ_ϕ dobro definisana, pa je, prema tome, $\mathcal{A}_\phi$ zaista automat.

Dalje, za proizvoljan $u \in X^*$ i $v = x_1x_2\cdots x_n$, za $x_1, x_2, \dots, x_n \in X$, imamo da važi
$$\delta_\phi(\phi_u, v) = a_1a_2\cdots a_n,$$
gde je
$$a_1 = \delta_\phi(\phi_u, x_1),\ a_2 = \delta_\phi(a_1, x_2), \dots,\ a_n = \delta_\phi(a_{n-1}, x_n).$$
Međutim, važi

$$\begin{aligned}
a_1 &= \delta_\phi(\phi_u, x_1) = \phi_{ux_1},\\
a_2 &= \delta_\phi(a_1, x_2) = \delta_\phi(\phi_{ux_1}, x_2) = \phi_{ux_1x_2},\\
&\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots\\
a_n &= \delta_\phi(a_{n-1}, x_n) = \delta_\phi(\phi_{ux_1x_2\cdots x_{n-1}}, x_n) = \phi_{ux_1x_2\cdots x_n},
\end{aligned}$$

pa dobijamo (5.10). Jednakost (5.11) sledi neposredno iz (5.10). Konačno, jednakost (5.12) ćemo dokazati indukcijom po dužini reči v. Jasno je da (5.12) važi za sve reči dužine 1. Prema tome, ostaje da se dokaže da iz indukcijske pretpostavke da (5.12) važi za sve reči dužine $k \leqslant n-1$ sledi da (5.12) važi i za v. Zaista, prema indukcijskoj pretpostavci, za $v' = x_1x_2\cdots x_{n-1}$ imamo da je

$$\lambda_\phi(\phi_u, v') = \phi_u(v'),$$

i dalje
$$\begin{aligned}
\lambda_\phi(\phi_u, v) &= \lambda_\phi(\phi_u, v'x_n)\\
&= \lambda_\phi(\phi_u, v')\lambda_\phi((\phi_u)v', x_n)\\
&= \lambda_\phi(\phi_u, v')\lambda_\phi(\phi_{uv'}, x_n)\\
&= (\phi_u(v'))(\phi_{uv'}(x_n))\\
&= (\phi_u(v'))((\phi_u)_{v'}(x_n))\\
&= \phi_u(v'x_n) = \phi_u(v),
\end{aligned}$$

što je i trebalo dokazati. Prema tome, (5.12) važi za sve reči $u, v \in X^*$. Ovim je dokaz kompletiran. □

Napomena 7. *Automat $\mathcal{A}_\phi = (A_\phi, X, Y, \delta_\phi, \lambda_\phi)$ definisan u prethodnom zadatku naziva se **donji automat** određen automatovnim preslikavanjem ϕ.*

Zadatak 5.184. *Dokazati da je svako automatovno preslikavanje indukovano donjim automatom $\mathcal{A}_\phi = (A_\phi, \phi_e, X, Y, \delta_\phi, \lambda_\phi)$ koji je određen ovim preslikavanjem i ima inicijalno stanje $\phi = \phi_e$.*

Rešenje: Neka je $\phi : X^* \to Y^*$ proizvoljno automatovno preslikavanje. Prema Zadatku 5.179. (b), $\phi = \phi_e$, pa iz (5.10) dobijamo da, za svaki $u \in X^*$ važi

$$\lambda_\phi(\phi_e, u) = \phi_e(u) = \phi(u).$$

Dakle, ϕ je indukovano automatom $\mathcal{A}_\phi$. □

Zadatak 5.185. *Neka je $A = (A, a_0, X, Y, \delta, \lambda)$ proizvoljan automat koji indukuje automatovno preslikavanje $\phi : X^* \to Y^*$. Dokazati da za proizvoljan $u \in X^*$ važi $\phi_u = \phi_{a_0 u}$, tj. za svaki $v \in X^*$ imamo*

$$\phi_u(v) = \lambda(a_0 u, v).$$

Rešenje: Uzmimo $u, v \in X^*$. Tada je

$$\begin{aligned}\phi_u(v) &= r_{|v|}(\phi(uv))\\ &= r_{|v|}(\lambda(a_0, uv))\\ &= r_{|v|}(\lambda(a_0, u)\lambda(a_0 u, v))\\ &= \lambda(a_0 u, v)\end{aligned}$$

čime je dokaz kompletiran. □

Zadatak 5.186. *Neka je ϕ automatovno preslikavanje, $\mathcal{A}$ je proizvoljan inicijalni automat koji indukuje ϕ i $\mathcal{A}'$ je stablo automata $\mathcal{A}$. Dokazati da je tada*

(a) $|A'| \leqslant |A^\phi|$;
(b) $|A_\phi| \leqslant |A'|$.

Rešenje: Neka $\phi : X^* \to Y^*$, $\mathcal{A} = (A, a_0, X, Y, \delta, \lambda)$ automat koji indukuje ϕ i neka je A' skup dostižnih stanja od A.

(a) Definišimo preslikavanje $\varphi : X^* \to A'$ sa

$$\varphi(u) = \delta(a_0, u) = a_0 u \qquad (u \in X^*).$$

Jasno, preslikavanje φ je dobro definisano i slika X^* na A', odnosno φ slika A^ϕ na A'.

(b) Definišimo preslikavanje $\psi : A' \to A_\phi$ sa

$$\psi(a) = \phi_u \ \Leftrightarrow\ a = a_0 u \qquad (a \in A').$$

Najpre ćemo dokazati da je ψ dobro definisano. Jasno, za svaki $a \in A'$ postoji $u \in X^*$ tako da je $a = a_0u$. Neka su $u, v \in X^*$ reči za koje je $a = a_0u = a_0v$. Tada prema Zadatku 5.185. imamo da je

$$\phi_u = \phi_{a_0u} = \phi_{a_0v} = \phi_v.$$

Na ovaj način smo dokazali da je ψ dobro definisano.

Jasno, za proizvoljan $\phi_u \in A_\phi$ je $\phi_u = (a_0u)\psi$, pa ψ slika A' na A_ϕ. Time je dokazano da važi (b). □

Zadatak 5.187. *Za dato automatovno preslikavanje ϕ, donji automat $\mathcal{A}_\phi$ jeste automat najmanje kardinalnosti koji indukuje dato preslikavanje. Dokazati.*

Rešenje: Ovo tvrđenje je direktna posledica prethodnog zadatka. □

Zadatak 5.188. *Neka je $X = Y = \{0,1\}$ i preslikavanje $\phi : X^* \to Y^*$ je definisano na sledeći način:*

(i) Ako 0101 *nije podreč od u, tada je $\phi(u) = u$. Dokazati.*
(ii) Ukoliko je 0101 *podreč od u, tada preslikavanje ϕ sva slova koja se u u javljaju posle prvog pojavljivanja podreči* 0101 *preinačuje u* 0*. Drugim rečima, ako u predstavimo u obliku $u = p0101q$, gde su $p,q \in X^*$ i* 0101 *nije podreč od $p010$, tada je $\phi(u) = \phi(p0101q) = p01010^{|q|}$. Dokazati.*

Da li je ϕ automatovno preslikavanje? Ako postoji konačan automat koji indukuje ϕ, naći minimalni takav automat.

Rešenje: Nije teško videti da je ϕ automatovno preslikavanje. Minimalni automa koji indukuje ovo preslikavanje jeste donji automat $\mathcal{A}_\phi$. Da bi smo našli ovaj automat (Zadatak 5.183.) odredićemo njegova stanja. Naime, za proizvoljnu reč $u \in X^*$ odredićemo stanje ϕ_u preslikavanja ϕ. Razlikujemo nekoliko slučajeva:
(1) Neka je 0101 podreč od u. Tada je $\phi(u) = p01010^{|q|}$, gde je $u = p0101q$ i $p,q \in X^*$ tako da 0101 nije podreč od $p010$, i za proizvoljnu reč $v \in X^*$ imamo da je

$$\phi_u(v) = r_{|v|}(\phi(uv)) = r_{|v|}(p01010^{|q|+|v|}) = 0^{|v|}.$$

(2) Neka 0101 nije podreč od u. Tada je bitan sufiks reči u dužine tri, tj. poslednja tri slova te reči. Imamo sledeće podslučejeve:
(2.1) Jedna od reči 111 i 011 je sufiks od u. Tada, za proizvoljnu reč $v \in X^*$, 0101 je podreč od uv ako i samo ako je podreč od v, pa je

$$\phi_u(v) = r_{|v|}(\phi(uv)) = r_{|v|}(u(\phi(v))) = \phi(v),$$

što znači da je $\phi_u = \phi$.

(2.2) Jedna od reči 000, 100 i 110 je sufiks od u. Neka je $v \in X^*$ proizvoljna reč. Ako je 101 prefiks od v, tada je

$$\phi_u(v) = r_{|v|}(\phi(uv)) = r_{|v|}(u1010^{|v|-3}) = 1010^{|v|-3}.$$

Sa druge strane, ako 101 nije prefiks od v, tada je 0101 podreč od uv ako i samo ako je podreč od v, pa kao u slučaju (2.1) dobijamo da je $v\phi_u = v\phi$.

Prema tome, za proizvoljan $v \in X^*$ je

$$\phi_u(v) = \begin{cases} 1010^{|v|-3}, & \text{ako je 101 prefiks od } v, \\ \phi(v), & \text{ako 101 nije prefiks od } v. \end{cases}$$

(2.3) Jedna od reči 001 i 101 je sufiks od u. Tada za proizvoljnu reč $v \in X^*$, kao u prethodnom slučaju dobijamo

$$\phi_u(v) = \begin{cases} 010^{|v|-2}, & \text{ako je 01 prefiks od } v, \\ \phi(v), & \text{ako 01 nije prefiks od } v. \end{cases}$$

(2.4) Reč 010 je sufiks od u. Tada za proizvoljnu reč $v \in X^*$, kao u prethodnim slučajevima dobijamo

$$\phi_u(v) = \begin{cases} 10^{|v|-1}, & \text{ako je 1 prvo slovo u } v, \\ \phi(v), & \text{ako je 0 prvo slovo u } v. \end{cases}$$

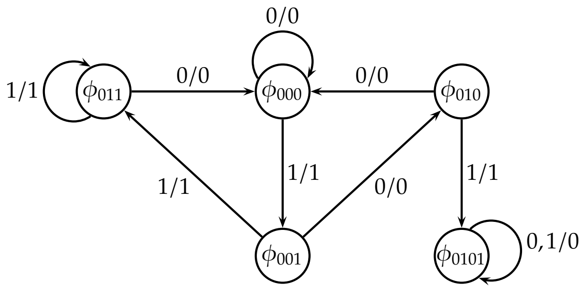
Prema tome, svako stanje automatovnog preslikavanja ϕ jednako je jednom od preslikavanja iz (1), (2.1), (2.2), (2.3) i (2.4). Ne umanjujući opštost, možemo uzeti da sva stanja preslikavanja ϕ jesu sledeća preslikavanja

$$\phi_{0101},\ \phi_{011},\ \phi_{000},\ \phi_{001} \text{ i } \phi_{010}.$$

Pri tome je $\phi = \phi_{011}$. Dakle, prema definiciji automata A_ϕ, iz Zadatka 5.183. za dato automatovno preslikavanje ϕ imamo da je to automat predstavljen tablicom

A_ϕ	ϕ_{0101}	ϕ_{000}	ϕ_{001}	ϕ_{010}	ϕ_{011}
0	$(\phi_{0101}, 0)$	$(\phi_{000}, 0)$	$(\phi_{010}, 0)$	$(\phi_{000}, 0)$	$(\phi_{000}, 0)$
1	$(\phi_{0101}, 0)$	$(\phi_{001}, 1)$	$(\phi_{011}, 1)$	$(\phi_{0101}, 1)$	$(\phi_{011}, 1)$

ili grafom



Tako smo dobili minimalni automat koji realizuje dato automatovno preslikavanje ϕ. $\square$

Zadatak 5.189. *Odrediti da li postoji konačan Mealyev automat koji realizuje preslikavanje* $\alpha : \{x,y\}^* \mapsto \{x,y\}^*$ *definisano sa*

$$\alpha(x_1x_2\dots x_n) = \begin{cases} x_1x_2\dots x_n, & \textit{ako je } x_1 = x; \\ x_1^c x_2^c \dots x_n^c, & \textit{ako je } x_1 = y, \textit{ gde je } x_i^c = \begin{cases} x, & x_i = y; \\ y, & x_i = x \end{cases} \end{cases} ?$$

Rešenje: Nije teško videti da je α automatovno preslikavanje, pa postoji konačan Mealyev automat koji ga realizuje. Jedan od načina za konstrukciju ovog automata je konstrukcija donjeg automata $\mathcal{A}_\alpha$. Odredićemo njegova stanja. Naime, za proizvoljnu reč $u \in X^*$ odredićemo stanje ϕ_u preslikavanja ϕ. Razlikujemo sledeće slučajeve:

(1) Neka je x prvo slovo reči u. Tada je $u = xu'$ i $u' \in X^*$. Za proizvoljnu reč $v = x_1x_2\dots x_n \in X^*$ imamo da je

$$\alpha_u(v) = r_{|v|}(\alpha(uv)) = r_{|v|}(xu'x_1x_2\dots x_n) = x_1x_2\dots x_n = v.$$

(2) Neka reč u počinje slovom y. U tom slučaju je $u = yu'$ i $u' \in X^*$ i za proizvoljnu reč $v = x_1x_2\dots x_n \in X^*$ važi

$$\alpha_u(v) = r_{|v|}(\alpha(uv)) = r_{|v|}(yu'x_1x_2\dots x_n) = x_1^c x_2^c \dots x_n^c.$$

Dakle, svako stanje automatovnog preslikavanja α jednako je jednom od preslikavanja iz (1) ili (2). Bez umanjenja opštosti možemo pretpostaviti da sva stanja preslikavanja α jesu preslikavanja α_x, α_y, ako reč ulaznog alfabeta počinje sa x, odnosno sa y, tim redom.

Pri tome je $\lambda_\alpha(\alpha, y) \neq \lambda_\alpha(\alpha_x, y)$. Dakle, prema definiciji automata A_α, na osnovu Zadatka 5.183. za dato automatovno preslikavanje α imamo da je to automat predstavljen tablicom

A_α	α_e	α_x	α_y
x	(α_x, x)	(α_x, x)	(α_y, y)
y	(α_y, x)	(α_y, y)	(α_x, x)

Tako smo dobili automat koji realizuje dato preslikavanje α. □

Zadatak 5.190. *Dokazati da je kompozicija automatovnih preslikavanja automatovno preslikavanje.*

Rešenje: Neka su $\alpha : X^* \mapsto Y^*$ i $\beta : Y^* \mapsto Z^*$ data automatovna preslikavanja realizovana automatima $\mathcal{A}_1 = (A_1, a_0^1, X, Y, \delta_1, \lambda_1)$ i $\mathcal{A}_2 = (A_2, a_0^2, Y, Z, \delta_2, \lambda_2)$, tim redom, što znači da je

$$\alpha(u) = \lambda_1(a_0^1, x_1x_2\dots x_n) \quad \text{i} \quad \beta(v) = \lambda_2(a_0^2, y_1y_2\dots y_k), \tag{5.13}$$

za proizvoljne $u = x_1x_2\dots x_n \in X^*$ i $v = y_1y_2\dots y_k \in Y^*$. Definišimo novi automat $\mathcal{A} = (A_1 \times A_2, (a_0^1, a_0^2), X, Z, \delta, \lambda)$ na sledeći način:

$$\begin{aligned} \delta((a_1,a_2),x) &= \big(\delta_1(a_1,x),\delta_2(a_2,\lambda_1(a_1,x))\big), \\ \lambda((a_1,a_2),x) &= \lambda_2(a_2,\lambda_1(a_1,x)). \end{aligned} \qquad ((a_1,a_2)\in A_1\times A_2,\ x\in X)$$

Dokazaćemo da ovako definisan automat $\mathcal{A}$ realizuje automatovno preslikavanje $\phi = \alpha\circ\beta$, tj. da je $\phi_{(a_0^1,a_0^2)} = \phi$.

Uzmimo proizvoljnu reč $u = x_1x_2\cdots x_n$, gde su $x_1,x_2,\dots,x_n \in X$. Prema definiciji proširenih funkcija prelaza i izlaza je

$$\begin{aligned} \delta_1(a_0^1,x_1x_2\dots x_n) &= a_1a_2\cdots a_n, \\ \lambda_1(a_0^1,x_1x_2\dots x_n) &= y_1y_2\cdots y_n, \\ \delta_2(a_0^2,y_1y_2\dots y_n) &= b_1b_2\cdots b_n. \end{aligned}$$

Primećujemo da važi:

$$\delta((a_0^1,a_0^2),x_1) = \big(\delta_1(a_0^1,x_1),\delta_2(a_0^2,\lambda_1(a_0^1,x_1))\big) = (\delta_1(a_0^1,x_1),\delta_2(a_0^2,y_1)) = (a_1,b_1).$$

Indukcijom po dužini reči $u = x_1x_2\dots x_n \in X^*$ dokazaćemo da je $\delta((a_i,b_i),x_{i+1}) = (a_{i+1},b_{i+1})$, za $i\in\mathbb{N}$. Naime, za $n=2$, imamo

$$\delta((a_1,b_1),x_2) = \big(\delta_1(a_1,x_2),\delta_2(b_1,\lambda_1(a_1,x_2))\big) = (\delta_1(a_1,x_2),\delta_2(b_1,y_2)) = (a_2,b_2).$$

Pretpostavimo da je $\delta((a_{k-1},b_{k-1}),x_k) = (a_k,b_k)$, za $n=k$ i dokažimo tvrđenje za $n=k+1$.

$$\begin{aligned} \delta((a_k,b_k),x_{k+1}) &= \big(\delta_1(a_k,x_{k+1}),\delta_2(b_k,\lambda_1(a_k,x_{k+1}))\big) \\ &= (\delta_1(a_k,x_{k+1}),\delta_2(b_k,y_{k+1})) = (a_{k+1},b_{k+1}). \end{aligned}$$

Time smo dokazali da tvrđenje važi za svaki $n\in\mathbb{N}$. Na osnovu definicije proširene funkcije prelaza sada imamo

$$\delta((a_0^1,a_0^2),x_1x_2\dots x_n) = (a_1,b_1)(a_2,b_2)\dots(a_n,b_n).$$

Za preslikavanje indukovano automatom $\mathcal{A}$ i datu reč $u = x_1x_2\dots x_n \in X^*$ važi

$$\begin{aligned} \phi_{(a_0^1,a_0^2)}(u) &= \lambda((a_0^1,a_0^2),x_1x_2\dots x_n) = \lambda((a_0^1,a_0^2),x_1)\lambda((a_1,b_1),x_2)\dots\lambda((a_{n-1},b_{n-1}),x_n) \\ &= \lambda_2(a_0^2,\lambda_1(a_0^1,x_1))\lambda_2(b_1,\lambda_1(a_1,x_2))\dots\lambda_2(b_{n-1},\lambda_1(a_{n-1},x_n)) \\ &= \lambda_2(a_0^2,y_1)\lambda_2(b_1,y_2)\dots\lambda_2(b_{n-1},y_n)) = \lambda_2(a_0^2,y_1y_2\dots y_n). \end{aligned}$$

Sa druge strane, prema (5.13) imamo

$$\begin{aligned} \phi(u) &= (\alpha\circ\beta)(u) = \beta(\alpha(u)) = \beta(\lambda_1(a_0^1,x_1x_2\dots x_n)) \\ &= \beta(y_1y_2\dots y_n) = \lambda_2(a_0^2,y_1y_2\dots y_n) = \phi_{(a_0^1,a_0^2)}(u), \end{aligned}$$

čime je dokaz kompletiran. □

Zadatak 5.191. *Pod težinom automatovnog preslikavanja ϕ, u oznaci $w(\phi)$, podrazumevamo kardinalni broj skupa stanja automata A_ϕ. Ukoliko je $w(\phi)$ konačan, za preslikavanje ϕ kažemo da je konačne težine. Ako su ϕ' i ϕ'' automatovna preslikavanja konačnih težina, tada je i $\phi'\phi''$ automatovno preslikavanje konačne težine i važi $w(\phi'\phi'') \leqslant w(\phi')w(\phi'')$. Dokazati.*

Rešenje: Prema prethodnom zadatku kompozicija automatovnih preslikavanja $\phi'\phi''$ je automatovno preslikavanje i prema Zadatku 5.187. automat $\mathcal{A}_{\phi'\phi''}$ je automat najmanje kardinalnost koji indukuje kompoziciju datih preslikavanja, što znači da je

$$|A_{\phi'\phi''}| \leqslant |A_{\phi'} \times A_{\phi''}|, \ \text{tj.} \ w(\phi'\phi'') \leqslant w(\phi')w(\phi'').$$

Ovim je dokaz kompletiran. □

Zadatak 5.192. *Neka je data proizvoljna kolekcija $\mathcal{F}$ automatovnih preslikavanja iz slobodnog monoida X^* u slobodan monoid Y^*. Dokazati da postoji automat $\mathcal{A}$ koji realizuje sva preslikavanja iz ove kolekcije.*

Rešenje: Za svako automatovno preslikavanje $\phi \in \mathcal{F}$, takvo da $\phi : X^* \mapsto Y^*$, možemo konstruisati automat $\mathcal{A}_\phi = (A_\phi, \phi_e, X, Y, \delta_\phi, \lambda_\phi)$ koji realizuje to preslikavanje, na način kako je to učinjeno u Zadatku 5.183.

Definišimo, sada, na skupu

$$A = \bigcup_{\phi \in \mathcal{F}} (A_\phi \times \phi)$$

funkciju prelaza $\delta : A \times X \mapsto A$ i izlaza $\lambda : A \times X \mapsto Y$ na sledeći način:

$$\begin{aligned} \delta((\phi_u, \phi), x) &= (\delta_\phi(\phi_u, x), \phi) = (\phi_{ux}, \phi) \\ \lambda((\phi_u, \phi), x) &= \lambda_\phi(\phi_u, x) = \phi_u(x). \end{aligned} \qquad (u \in X^*, \phi_u \in A_\phi)$$

Jednostavno se pokazuje da su funkcije δ i λ dobro definisane, te smo, na ovaj način, konstruisali automat $\mathcal{A} = (A, (\phi_e, \phi), X, Y, \delta, \lambda)$. Ako sa $\mathcal{F}_A$ označimo skup svih preslikavanja indukovanih automatom $\mathcal{A}$ treba da pokazemo da je $\mathcal{F} \subseteq \mathcal{F}_A$. U tom cilju ćemo pokazati da je proizvoljno preslikavanje $\phi \in \mathcal{F}$ indukovano inicijalnim stanjem automata $\mathcal{A}$.

Neka je $u = x_1 x_2 \dots x_n \in X^*$ proizvoljna reč. Prema definiciji proširene funkcije prelaza imamo da je $\delta((\phi_e, \phi), x_1 x_2 \dots x_n) = \alpha_1 \alpha_2 \dots \alpha_n$ i indukcijom po n ćemo dokazati da je $\alpha_k = (\phi_{x_1 \dots x_k}, \phi)$.

Za $n = 1$ važi $\alpha_1 = \delta((\phi_e, \phi), x_1) = (\delta_\phi(\phi_e, x_1), \phi) = (\phi_{ex_1}, \phi) = (\phi_{x_1}, \phi)$.

Pretpostavimo da tvrđenje važi za $n = k$, i dokažimo da važi za $n = k + 1$. Imamo da je

$$a_{k+1} = \delta(\alpha_k, x_{k+1}) = \delta((\phi_{x_1 \dots x_k}, \phi), x_{k+1}) = (\phi_{x_1 \dots x_k x_{k+1}}, \phi),$$

pa tvrđenje važi za svaki $n \in \mathbb{N}$. Dakle,

$$\delta((\phi_e,\phi),x_1x_2\dots x_n) = (\phi_{x_1},\phi)(\phi_{x_1x_2},\phi)\dots(\phi_{x_1\dots x_n},\phi),$$

dok iz definicije proširene funkcije izlaza i prema jednakosti (5.9) dobijamo sledeće:

$$\begin{aligned}\lambda((\phi_e,\phi),x_1x_2\dots x_n) &= \lambda((\phi_e,\phi),x_1)\lambda((\phi_{x_1},\phi),x_2)\dots\lambda((\phi_{x_1\dots x_{n-1}},\phi),x_n)\\ &= \lambda_\phi(\phi_e,x_1)\lambda_\phi(\phi_{x_1},x_2)\dots\lambda_\phi(\phi_{x_1x_2\dots x_{n-1}},x_n)\\ &= \phi_e(x_1)\phi_{x_1}(x_2)\dots\phi_{x_1x_2\dots x_{n-1}}(x_n) = \phi(x_1x_2\dots x_n)\\ &= \phi(u), \quad \text{za } u = x_1x_2\dots x_n \in X^*.\end{aligned}$$

Ovim smo pokazali da, za proizvoljno automatovno prelikavanje $\phi \in \mathcal{F}$ jeste $\phi(u) = \phi_{(\phi_e,\phi)}(u)$, za svaki $u \in X^*$, tj. da je $\mathcal{F} \subseteq \mathcal{F}_A$, čime smo kompletirali dokaz. □

5.3. Ekvivalentni automati

Neka je $\mathcal{A} = (A,X,Y,\delta,\lambda)$ Mealyev automat. Sa Φ_A ćemo označavati skup svih automatovnih preslikavanja indukovanih stanjima automata $\mathcal{A}$.

Ako su dati automati $\mathcal{A} = (A,X,Y,\delta,\lambda)$ i $\mathcal{A}' = (A',X,Y,\delta',\lambda')$, onda za $a \in A$ i $a' \in A'$ kažemo da su *ekvivalentna stanja* ako a i a' indukuju isto automatovno preslikavanje, tj. ako je $\phi_a = \phi_{a'}$. Slično, za $\mathcal{A}$ i $\mathcal{A}'$ kažemo da su *ekvivalentni automati* ako je $\Phi_A = \Phi_{A'}$, tj. ako je svako stanje automata $\mathcal{A}$ ekvivalentno nekom stanju automata $\mathcal{A}'$ i obratno. Jasno, ovako uvedena relacija među automatima jeste relacija ekvivalencije na skupu svih automata, što opravdava njen naziv. Dalje, ako su $\mathcal{A}$ i $\mathcal{A}'$ inicijalni automati, onda kažemo da su $\mathcal{A}$ i $\mathcal{A}'$ *ekvivalentni inicijalni automati* ako su ekvivalentna njihova inicijalna stanja, tj. ako $\mathcal{A}$ i $\mathcal{A}'$ indukuju isto automatovno preslikavanje.

Zadatak 5.193. *Dokazati da je svaki automat $\mathcal{A}$ Mealyevog tipa ekvivalentan je nekom automatu $\mathcal{B}$ Mooreovog tipa.*

Pri tome, ako je $|A| = n$ i $|X| = m$, gde je X ulazni alfabet, tada se Mooreov automat $\mathcal{B}$ može izabrati tako da bude $|B| = n(m+1)$. Dokazati.

Rešenje: Neka je $\mathcal{A} = (A,X,Y,\delta,\lambda)$ automat Mealyevog tipa. Stavimo da bude $B = A \cup A \times X$ i definišimo preslikavanja $\delta' : B \times X \to B$ i $\lambda' : B \times X \to Y$ sa:

$$\delta'(b,x) = \begin{cases} (a,x) & \text{ako je } b = a \in A,\\ (\delta(a,x'),x) & \text{ako je } b = (a,x') \in A \times X,\end{cases}$$

$$\lambda'(b,x) = \begin{cases} \lambda(a,x) & \text{ako je } b = a \in A,\\ \lambda(\delta(a,x'),x) & \text{ako je } b = (a,x') \in A \times X,\end{cases}$$

gde su $b \in B$ i $x \in X$. Tada je $\mathcal{B} = (B,X,Y,\delta',\lambda')$ automat za koji ćemo dokazati da je ekvivalentan sa $\mathcal{A}$ i da je Mooreovog tipa.

Prvo ćemo dokazati da svako stanje $a \in A$ indukuje isto automatovno preslikavanje u automatu A i u automatu B, odnosno da važi

$$\lambda(a,u) = \lambda'(a,u),$$

za svaku reč $u \in X^*$. Uzmimo da je $u = x_1x_2\dots x_k$, za neke $x_1,x_2,\dots,x_k \in X$. Tada je

$$\delta(a,u) = a_1a_2\dots a_k \quad \text{i} \quad \lambda(a,u) = y_1y_2\dots y_k,$$

za $a_1,a_2,\dots,a_k \in A$ i $y_1,y_2,\dots,y_k \in Y$ određene sa

$$\begin{aligned} &a_1 = \delta(a,x_1),\ a_2 = \delta(a_1,x_2),\ \dots,\ a_k = \delta(a_{k-1},x_k),\\ &y_1 = \lambda(a,x_1),\ y_2 = \lambda(a_1,x_2),\ \dots,\ y_k = \lambda(a_{k-1},x_k). \end{aligned}$$

Imamo da je

$$\begin{aligned} &\delta'(a,x_1) = (a,x_1),\\ &\delta'((a,x_1),x_2) = (\delta(a,x_1),x_2) = (a_1,x_2),\\ &\dots\dots\dots\dots\dots\dots\\ &\delta'((a_{k-2},x_{k-1}),x_k) = (\delta(a_{k-2},x_{k-1}),x_k) = (a_{k-1},x_k), \end{aligned}$$

što znači da je

$$\delta'(a,u) = b_1b_2\dots b_k,$$

gde je

$$b_1 = (a,x_1),\ b_2 = (a_1,x_2),\ \dots,\ b_k = (a_{k-1},x_k).$$

Odavde dalje dobijamo da je

$$\begin{aligned} &\lambda'(a,x_1) = \lambda(a,x_1) = y_1,\\ &\lambda'(b_1,x_2) = \lambda(\delta(a,x_1),x_2) = \lambda(a_1,x_2) = y_2,\\ &\dots\dots\dots\dots\dots\dots\\ &\lambda'(b_{k-1},x_k) = \lambda(\delta(a_{k-2},x_{k-1}),x_k) = \lambda(a_{k-1},x_k) = y_k, \end{aligned}$$

pa je

$$\lambda'(a,u) = y_1y_2\dots y_k = \lambda(a,u),$$

što je i trebalo dokazati.

Na potpuno isti način se dokazuje da je svako stanje $(a,x) \in A\times X$ automata B ekvivalentno stanju $\delta(a,x)$ automata A. Prema tome, automati A i B su zaista ekvivalentni.

Da bi smo dokazali da je B automat Mooreovog tipa, razmotrimo $b,b' \in B$ i $x,x' \in X$ takve da je $\delta'(b,x) = \delta'(b',x')$. Moguća su četiri slučaja:

(1) $b = a \in A$, $b' = a' \in A$,

(2) $b = a \in A$, $b' = (a',x_1') \in A\times X$,

(3) $b = (a,x_1) \in A\times X$, $b' = a' \in A$,

(4) $b = (a,x_1) \in A\times X$, $b' = (a',x_1') \in A\times X$.

U svakom od tih slučajeva, koristeći činjenicu da je $\delta'(b,x) = \delta'(b',x')$, dobijamo da je $\lambda'(b,x) = \lambda'(b',x')$. Time je dokazano da je $\mathcal{B}$ automat Mooreovog tipa.

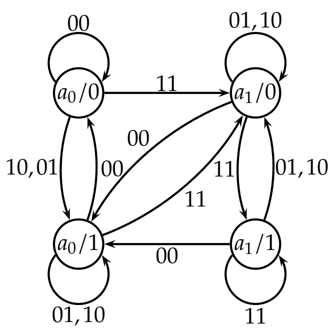
Jasno je da iz $|A| = n$ i $|X| = m$ sledi da je $|B| = n + mn = n(m+1)$. □

Zadatak 5.194. *Dokazati da automat Mooreovog tipa koji realizuje dato automatovno preslikavanje ima veći broj stanja od automata Mealyevog tipa koji indukuje to isto preslikavanje.*

Rešenje: Tvrđenje je direktna posledica prethodnog zadatka. □

Zadatak 5.195. *Konstruisati Mooreov automat ekvivalentan Mealyevom automatu koji je konstruisan u Zadatku 5.175. (Automat sabira dva broja u binarnom zapisu.)*

Rešenje: Prema načinu konstrukcije Mooreovog automata ekvivalentnog datom Mealyevom automatu koji je predstavljen u Zadatku 5.193. graf prelaza traženog automata Mooreovog tipa je



Stanja automata su označena sa a_i/j, za $i,j \in \{0,1\}$ pri čemu i označava prenos, a j označava izlaz. □

5.4. Homomorfizmi i kongruencije

Kongruencija na automatu sa izlazom $\mathcal{A} = (A,X,Y,\delta,\lambda)$ definiše kao relacija ekvivalencije ϱ na A koja zadovoljava uslov da za proizvoljne $a,b \in A$ i $x \in X$, iz $(a,b) \in \varrho$ sledi da je $(ax,bx) \in \varrho$ i $\lambda(a,x) = \lambda(b,x)$.

Jednostavno se pokazuje da, za kongruenciju ϱ na $\mathcal{A}$ i proizvoljnu reč $u \in X^*$ i stanja $a,b \in A$, iz $(a,b) \in \varrho$ sledi da je $(au,bu) \in \varrho$ i $\lambda(a,u) = \lambda(b,u)$.

Ako su $\mathcal{A} = (A,X,Y,\delta,\lambda)$ i $\mathcal{A}' = (A',X,Y,\delta',\lambda')$ dati automati i $\varphi : A \to A'$ preslikavanje takvo da za svaki $a \in A$ i $x \in X$ važi

$$\varphi(\delta(a,x)) = \delta'(\varphi(a),x) \qquad \text{i} \qquad \lambda(a,x) = \lambda'(\varphi(a),x),$$

tada preslikavanje φ nazivamo *homomorfizmom* automata $\mathcal{A}$ u automat $\mathcal{A}'$. Za $\varphi(A)$ kažemo da je *homomorfna slika* automata $\mathcal{A}$. Osim toga, ako je φ i bijekcija, onda ga nazivamo *izomorfizmom* automata $\mathcal{A}$ na automat $\mathcal{A}'$, a za automate $\mathcal{A}$ i $\mathcal{A}'$ kažemo da su *izomorfni*.

U slučaju kada su $\mathcal{A}$ i $\mathcal{A}'$ inicijalni automati, tada *homomorfizmom inicijalnih automata* nazivamo homomorfizam automata $\mathcal{A}$ u automat $\mathcal{A}'$ koji inicijalno stanje automata $\mathcal{A}$ slika u inicijalno stanje automata $\mathcal{A}'$.

Ako je ϱ kongruencija na automatu $\mathcal{A}$, tada slično kao kod algebarskih struktura uvodimo pojam faktor-automata na sledeći način:

Na faktor-skupu A/ϱ definišemo preslikavanja

$$\delta_\varrho : (A/\varrho)\times X \to A/\varrho \quad \text{i} \quad \lambda_\varrho : (A/\varrho)\times X \to Y,$$

sa

$$\delta_\varrho(a\varrho, x) = (\delta(a,x))\varrho \quad \text{i} \quad \lambda_\varrho(a\varrho, x) = \lambda(a,x),$$

za sve $a \in A$ i $x \in X$. Koristeći činjenicu da je ϱ kongruencija na A, lako se proverava da su preslikavanja δ_ϱ i λ_ϱ dobro definisana, tj. da njihove vrednosti ne zavise od izbora predstavnika ϱ-klasa, pa $(A/\varrho, X, Y, \delta_\varrho, \lambda_\varrho)$ jeste automat koji obeležavamo sa $\mathcal{A}/\varrho$ i nazivamo *faktor-automatom* automata $\mathcal{A}$ u odnosu na kongruenciju ϱ.

Vezu između kongruencija na automatu i homomorfizama daje nam sledeća teorema.

Teorema 5.19. (Teorema o homomorfizmu). *Ako je ϱ kongruencija na automatu $\mathcal{A}$, tada je $\varrho^\natural$ homomorfizam iz A na A/ϱ.*

Obratno, ako je φ homomorfizam iz automata $\mathcal{A} = (A, X, Y, \delta, \lambda)$ na automat $\mathcal{A}' = (A', X, Y, \delta', \lambda')$, tada je $\ker\varphi$ kongruencija na $\mathcal{A}$ i preslikavanje $\Phi : A/\ker\varphi \mapsto A'$ definisano sa $\Phi : (a\ker\varphi) \mapsto \varphi(a)$ je izomorfizam iz $A/\ker\varphi$ na A'.

Zadatak 5.196. *Dokazati da je svaka homomorfna slika automata $\mathcal{A}$ ekvivalentna sa tim automatom.*

Rešenje: Neka je $\mathcal{A} = (A, X, Y, \delta, \lambda)$, $\mathcal{A}' = (A', X, Y, \delta', \lambda')$ i φ je homomorfizam iz $\mathcal{A}$ na $\mathcal{A}'$. Tada, za proizvoljno stanje $a \in A$ važi

$$\phi_a(u) = \lambda(a,u) = \lambda'(\varphi(a), u) = \phi_{\varphi(a)}(u),$$

za svaki $u \in X^*$. Prema tome, $\phi_a = \phi_{\varphi(a)}$, za svaki $a \in A$, pa kako φ slika $\mathcal{A}$ na $\mathcal{A}'$, to je $\Phi_A = \Phi_{A'}$, čime smo dokazali da su $\mathcal{A}$ i $\mathcal{A}'$ ekvivalentni automati. □

Zadatak 5.197. *Za proizvoljan automat $\mathcal{A} = (A, X, Y, \delta, \lambda)$, relacija ϱ_A na $\mathcal{A}$ definisana sa*

$$(a,b) \in \varrho_A \;\Leftrightarrow\; (\forall u \in X^*)\;\; \lambda(a,u) = \lambda(b,u), \qquad (a,b \in A)$$

je najveća kongruencija na A. Dokazati.

Rešenje: Lako se proverava da je ϱ_A relacija ekvivalencije na $\mathcal{A}$. Da bi smo dokazali njenu saglasnost, uzmimo $a, b \in A$ takve da je $(a,b) \in \varrho_A$ i uzmimo proizvoljan $x \in X$. Tada za svaku reč $u \in X^*$ važi

$$\begin{aligned}\lambda(\delta(a,x),u) &= \lambda(ax,u) = r_{|u|}(\lambda(a,xu)) \\ &= r_{|u|}(\lambda(b,xu)) \\ &= \lambda(bx,u) = \lambda(\delta(b,x),u),\end{aligned}$$

što znači da je $\big(\delta(a,x),\delta(b,x)\big) \in \varrho_A$. Pored toga, iz $(a,b) \in \varrho_A$ sledi $\lambda(a,x) = \lambda(b,x)$. Dakle, ϱ_A je kongruencija na A.

Da bi smo dokazali da je ϱ_A najveća kongruencija na A, uzmimo proizvoljnu kongruenciju ϱ na A i $a,b \in A$ takve da je $(a,b) \in \varrho$. Za proizvoljnu reč $u \in X^*$ je $\lambda(a,u) = \lambda(b,u)$, odakle sledi da je $(a,b) \in \varrho_A$. Prema tome, $\varrho \subseteq \varrho_A$. Ovim je dokaz kompletiran. □

Zadatak 5.198. *Faktor automat $\mathcal{A}/\varrho_A = (A/\varrho_A, a_0\varrho_A, X, Y, \delta_{\varrho_A}, \lambda_{\varrho_A})$ dostižnog, inicijalnog automata $\mathcal{A} = (A, a_0, X, Y, \delta, \lambda)$, u odnosu na relaciju ϱ_A koja je definisana u prethodnom zadatku, jeste automat najmanje kardinalnosti koji indukuje isto automatovno preslikavanje kao automat $\mathcal{A}$. Dokazati.*

Rešenje: Neka je ϕ automatovno preslikavanje realizovano automatom $\mathcal{A}$. Definisaćemo preslikavanje $\varphi : A \mapsto A/\varrho_A$ koje svakom stanju $a \in A$ dodeljuje njegovu klasu u odnosu na kongruenciju ϱ_A, tj. $\varphi(a) = a\varrho_A$. Jednostavno se pokazuje da je φ homomorfizam automata $\mathcal{A}$ na $\mathcal{A}/\varrho_A$. Kao homomorfna slika automata $\mathcal{A}$, prema Zadatku 5.196. faktor automat $\mathcal{A}/\varrho_A$ je ekvivalentan automatu $\mathcal{A}$, pa indukuje preslikavanje ϕ.

Sa druge strane, primetimo da za stanja $a,b \in A$ važi

$$(a,b) \in \varrho_A \;\Leftrightarrow\; \phi_a = \phi_b. \tag{5.14}$$

Kako je automat $\mathcal{A}$ dostižan može se definisati preslikavanje $\psi : A/\varrho_A \mapsto A_\phi$ sa $\psi(a\varrho_A) = \phi_u$, ako je $a = \delta(a_0,u)$, za $u \in X^*$. Prema (5.14) preslikavanje ψ jeste dobro definisano. Pokazuje se da je ψ bijekcija i homorfizam između automata $\mathcal{A}/\varrho_A$ i A_ϕ, pa je $|A/\varrho_A| = |A_\phi|$. U Zadatku 5.187. pokazali smo da je automat $\mathcal{A}_\phi$ automat najmanje kardinalnosti koji indukuje preslikavanje ϕ, pa tvrđenje važi i za $\mathcal{A}/\varrho_A$. □

Zadatak 5.199. *Faktor automat $\mathcal{A}/\varrho_A = (A/\varrho_A, X, Y, \delta_{\varrho_A}, \lambda_{\varrho_A})$ jeste automat najmanje kardinalnosti u klasi svih automata ekvivalentnih sa $\mathcal{A} = (A, X, Y, \delta_A, \lambda_A)$. Dokazati.*

Rešenje: Neka je $\mathcal{B} = (B, X, Y, \delta_B, \lambda_B)$ proizvoljan automat ekvivalentan automatu $\mathcal{A}$. Kako za proizvoljno stanje $b \in B$ postoji $a \in A$ tako da je $\phi_b = \phi_a$, možemo definisati preslikavanje $\varphi : B \mapsto A/\varrho_A$ tako da je $\varphi(b) = a\varrho_A$.

Dokažimo dobru definisanost preslikavanja φ. Pretpostavimo da za neko stanje $b \in B$ važi $\varphi(b) = a\varrho_A$ i $\varphi(b) = a'\varrho_A$. To znači da je $\phi_b = \phi_a$ i $\phi_b = \phi_{a'}$ odakle je $\phi_a = \phi_{a'}$, odakle prema (5.14) sledi da $(a,a') \in \varrho_A$, tj. $a\varrho_A = a'\varrho_A$. Dakle, φ je dobro definisano.

Takođe, za proizvoljno stanje $a\varrho_A \in A/\varrho_A$ je $a \in A$, pa postoji $b \in B$ takvo da je $\phi_a = \phi_b$, što znači da je $\varphi(b) = a\varrho_A$. Zaključujemo da je φ "na" preslikavanje i odavde sledi da je $|A/\varrho_A| \leqslant |B|$, čime je tvrđenje zadatka dokazano. □

5.5. Minimizacija automata sa izlazom

Videli smo da redukovani automat $\mathcal{A}/\varrho_A$ automata $\mathcal{A}$ jeste automat sa minimalnim brojem stanja među automatima ekvivalentnim sa $\mathcal{A}$. Sada ćemo govoriti o procesu *minimizacije automata* $\mathcal{A}$, pod čime podrazumevamo svaki efektivni postupak kojim nalazimo njegov faktor automat $\mathcal{A}/\varrho_A$, odnosno kojim određujemo kongruenciju ϱ_A. Takav postupak nazivamo *algoritmom minimizacije*.

Važi sledeća teorema.

Teorema 5.20. *Neka je na automatu* $\mathcal{A} = (A, X, Y, \delta, \lambda)$ *definisan niz relacija* $\{\varrho_k\}_{k\in\mathbb{N}}$ *sa:*

$$\varrho_1 = \{(a,b) \in A \times A \,|\, (\forall x \in X)\ \lambda(a,x) = \lambda(b,x)\}$$
$$\varrho_{k+1} = \{(a,b) \in \varrho_k \,|\, (\forall x \in X)\,(ax,bx) \in \varrho_k\}.$$

Tada važi sledeće:

(a) *Svaki član niza* $\{\varrho_k\}_{k\in\mathbb{N}}$ *je relacija ekvivalencije na* A *i važi*

$$\varrho_1 \supseteq \varrho_2 \supseteq \cdots \supseteq \varrho_k \supseteq \varrho_{k+1} \supseteq \cdots \supseteq \varrho_A.$$

(b) *Ako je* $\varrho_k = \varrho_{k+1}$, *za neki* $k \in \mathbb{N}$, *tada je* $\varrho_k = \varrho_{k+m}$, *za svaki* $m \in \mathbb{N}$.

(c) *Ako je* $\mathcal{A}$ *konačan automat, tada postoji* $k \in \mathbb{N}$ *tako da je* $\varrho_k = \varrho_A$.

Koristeći prethodnu teoremu, možemo dati jedan *algoritam za minimizaciju automata sa izlazom*. Algoritam započinjemo formiranjem liste P svih parova stanja automata $\mathcal{A}$. Ovu listu grafički predstavljamo tablicom, a zbog refleksivnosti i simetričnosti relacija koje konstruišemo, dovoljno je razmatrati samo parove koji leže ispod glavne dijagonale te tablice.

U prvom koraku algoritma određujemo relaciju ϱ_1 i sa liste P brišemo sve parove stanja koji nisu u toj relaciji. Neka su, posle k-tog koraka, na listi P ostali parovi koji čine relaciju ϱ_k. Tada u $k+1$-vom koraku gradimo relaciju ϱ_{k+1} na taj način što razmatramo sve parove (a,b) koji su na početku tog koraka bili na listi P, i ukoliko proverom ustanovimo da postoji $x \in X$ tako da par (ax,bx) na početku tog koraka nije bio na listi, onda sa liste brišemo parove (a,b) i (b,a). Algoritam se završava prvim korakom u kome nije bilo brisanja sa liste.

Minimizacijom Mooreovog automata, korišćenjem navedenog algoritma za automate Mealyevog tipa, ne dobijamo uvek Mooreov automat. To nas dalje navodi na zaključak da, kada radimo sa Mooreovim automatima, treba drugačije definisati pojmove homomorfizma i kongruencije i naći drugačiji algoritam za minimizaciju, koji bi kao rezultat dao Mooreov automat.

Neka su $\mathcal{A} = (A, X, Y, \delta, \mu)$ i $\mathcal{A}' = (A', X, Y, \delta' \mu')$ dva Mooreova automata istog tipa. Preslikavanje $\varphi : A \to A'$ takvo da za svaki $a \in A$ i $x \in X$ važi

$$\varphi(\delta(a,x)) = \delta'(\varphi(a), x) \qquad \text{i} \qquad \mu(a) = \mu'(\varphi(a)),$$

nazivamo *homomorfizmom Mooreovog automata* $\mathcal{A}$ u Mooreov automat $\mathcal{A}'$, ili *homomorfizmom Mooreovog tipa*.

Relaciju ekvivalencije τ na Mooreovom automatu $\mathcal{A} = (A, X, Y, \delta, \mu)$ nazivamo *kongruencijom na Mooreovom automatu*, ili *kongruencijom Mooreovog tipa* na $\mathcal{A}$, ako za sve $a, b \in A$ iz $(a, b) \in \tau$ sledi

(1) $\mu(a) = \mu(b)$;
(2) $(ax, bx) \in \tau$, za svaki $x \in X$.

Nije teško proveriti da homomorfna slika Mooreovog automata, u odnosu na homomorfizam Mooreovog tipa, takođe jeste Mooreov automat. Obratno, ako je τ kongruencija Mooreovog tipa na Mooreovom automatu $\mathcal{A}$, tada je faktor-skup A/τ i sam Mooreov automat sa funkcijama prelaza i znaka

$$\delta_\tau : (A/\tau) \times X \to A/\tau \qquad \text{i} \qquad \mu_\tau : A/\tau \to Y$$

definisanim sa:

$$\delta_\tau(a\tau, x) = (\delta(a,x))\tau \qquad \text{i} \qquad \mu_\tau(a\tau) = \mu(a),$$

za $a \in A$, $x \in X$.

Slično kao kod automata Mealyevog tipa važi sledeće tvrđenje:

Teorema 5.21. *Za proizvoljan Mooreov automat $\mathcal{A} = (A, X, Y, \delta, \mu)$, relacija τ_A na A definisana sa*

$$(a,b) \in \tau_A \iff (a,b) \in \varrho_A \ \ i \ \ \mu(a) = \mu(b),$$

je najveća kongruencija Mooreovog tipa na A.

Definišimo niz $\{\tau_k\}_{k\in\mathbb{N}}$ relacija na A sa

$$\tau_1 = \{(a,b) \in A \times A \mid \mu(a) = \mu(b)\}$$
$$\tau_{k+1} = \{(a,b) \in \tau_k \mid (\forall x \in X)(ax, bx) \in \tau_k\}.$$

Tada važi sledeće:

(a) *Svaki član niza $\{\tau_k\}_{k\in\mathbb{N}}$ je relacija ekvivalencije na A i važi*

$$\tau_1 \supseteq \tau_2 \supseteq \cdots \supseteq \tau_k \supseteq \tau_{k+1} \supseteq \cdots \supseteq \tau_A.$$

(b) *Ako je $\tau_k = \tau_{k+1}$, za neki $k \in \mathbb{N}$, tada je $\tau_k = \tau_{k+m}$, za svaki $m \in \mathbb{N}$.*
(c) *Ako je $\mathcal{A}$ konačan automat, tada postoji $k \in \mathbb{N}$ tako da je $\tau_k = \tau_A$.*

Algoritam za minimizaciju automata Mooreovog tipa, tj. za određivanje kongruencije τ_A, zasnovan na prethodnoj teoremi, analogan je datom algoritmu za minimizaciju automata Mealyevog tipa.

Zadatak 5.200. *Neka je automat* $\mathcal{A} = (A, X, Y, \delta, \lambda)$ *zadat sledećom tablicom:*

A	a_1	a_2	a_3	a_4	a_5
x_1	(a_1, y_1)	(a_1, y_1)	(a_3, y_1)	(a_4, y_1)	(a_2, y_1)
x_2	(a_1, y_2)	(a_3, y_1)	(a_4, y_2)	(a_4, y_2)	(a_2, y_1)

Minimizirati dati automat.

Rešenje: Primenićemo algoritam za minimizaciju Mealyevog automata $\mathcal{A}$. Dakle, kreiraćmo listu svih parova P i krenućemo dalje sa algoritmom.

1. korak: Formirajmo relaciju ϱ_1. Ona ima dve klase:

$$\{a_1, a_3, a_4\},\ \{a_2, a_5\}.$$

Prema tome, sa liste brišemo sve parove iz skupa

$$\{a_1, a_3, a_4\} \times \{a_2, a_5\} \cup \{a_2, a_5\} \times \{a_1, a_3, a_4\}.$$

2. korak: Proveravamo parove koji leže ispod glavne dijagonale u P posle 1. koraka:

$(a_3x_1, a_1x_1) = (a_3, a_2)$ – nije na listi, pa se parovi (a_3, a_1) i (a_1, a_3) brišu;
$(a_4x_1, a_1x_1) = (a_4, a_2)$ – nije na listi, pa se parovi (a_4, a_1) i (a_1, a_4) brišu;
$(a_4x_1, a_3x_1) = (a_4, a_3)$ – na listi je;
$(a_4x_2, a_3x_2) = (a_4, a_4)$ – na listi je;
$(a_5x_1, a_2x_1) = (a_2, a_1)$ – nije na listi, pa se parovi (a_5, a_2) i (a_2, a_5) brišu.

	a_1	a_2	a_3	a_4	a_5
a_1		×			×
a_2	×		×	×	
a_3		×			×
a_4		×			×
a_5	×		×	×	

posle 1. koraka
relacija ϱ_1

	a_1	a_2	a_3	a_4	a_5
a_1		×	×	×	×
a_2	×		×	×	×
a_3	×	×			×
a_4	×	×			×
a_5	×	×	×	×	

posle 2. i 3. koraka
relacije $\varrho_2 = \varrho_3 = \varrho_A$

3. korak: U ovom koraku ostaje da se proveri samo par (a_3, a_4). Kako je $(a_3x_1, a_4x_1) = (a_3, a_4)$ i $(a_3x_2, a_4x_2) = (a_4, a_4)$, i oba ova para su na listi, to se u ovom koraku ništa ne brise, što znači da se algoritm zaustavlja.

Dakle, dobili smo da je $\varrho_2 = \varrho_3 = \varrho_A$ i ϱ_A-klase su

$$\{a_1\},\ \{a_2\},\ \{a_3, a_4\},\ \{a_5\},$$

pa je automat $\mathcal{A}/\varrho_A$ zadat tablicom

A/ϱ_A	$\overline{a_1}$	$\overline{a_2}$	$\overline{a_3}$	$\overline{a_5}$
x_1	$(\overline{a_1}, y_1)$	$(\overline{a_1}, y_1)$	$(\overline{a_3}, y_1)$	$(\overline{a_2}, y_1)$
x_2	$(\overline{a_1}, y_2)$	$(\overline{a_3}, y_1)$	$(\overline{a_3}, y_2)$	$(\overline{a_2}, y_1)$

gde je sa $\overline{q}$ označena ϱ_A-klasa stanja $q \in A$. □

Zadatak 5.201. *Neka je dat automat $\mathcal{A} = (A, X, Y, \delta, \lambda)$ sa sledećom prelazno-izlaznom tablicom:*

A	a	b	c	d
x_1	(a, y_1)	(c, y_2)	(a, y_1)	(b, y_1)
x_2	(b, y_1)	(c, y_2)	(b, y_1)	(b, y_1)

Pokazati da je $\mathcal{A}$ Mooreov automat, a zatim naći minimalni Mealyev automat ekvivalentan automatu $\mathcal{A}$. Da li je dobijeni automat Mooreov?

Rešenje: Iz tablice se jasno vidi da se radi o Mooreovom automatu koji se može predstaviti tablicom:

A	y_1	y_1	y_2	y_2
	a	b	c	d
x_1	a	c	a	b
x_2	b	c	b	b

Primenimo algoritam za minimizaciju automata $\mathcal{A}$ kao Mealyevog automata. Dakle, kreirajmo listu svih parova P i krenimo dalje sa algoritmom.

1. korak: Formirajmo relaciju ϱ_1. Ona ima dve klase: $\{a, c, d\}$ i $\{b\}$. Prema tome, sa liste brišemo sve parove iz skupa

$$\{a, c, d\} \times \{b\} \cup \{b\} \times \{a, c, d\}.$$

2. korak: Proveravamo parove koji leže ispod glavne dijagonale u P posle 1. koraka:

$(cx_1, ax_1) = (a, a)$ – na listi je;
$(cx_2, ax_2) = (b, b)$ – na listi je;
$(dx_1, ax_1) = (b, a)$ – nije na listi, pa se parovi (d, a) i (a, d) brišu;
$(dx_1, cx_1) = (b, a)$ – nije na listi, pa se parovi (d, c) i (c, d) brišu.

	a	b	c	d
a		X		
b	X		X	X
c		X		
d		X		

posle 1. koraka
relacija ϱ_1

	a	b	c	d
a		X		X
b	X		X	X
c		X		X
d	X	X	X	

posle 2. i 3. koraka
relacije $\varrho_2 = \varrho_3 = \varrho_A$

3. korak: U ovom koraku ostaje da se proveri samo par (c,a). Kako je $(cx_1,ax_1) = (a,a)$ i $(cx_2,ax_2) = (b,b)$, i oba ova para su na listi, to se u ovom koraku ništa ne brise, što znači da se algoritm zaustavlja.

Dakle, dobili smo da je $\varrho_2 = \varrho_3 = \varrho_A$ i ϱ_A-klase su $\{a,c\}$, $\{b\}$ i $\{d\}$, pa je automat $\mathcal{A}/\varrho_A$ zadat tablicom

A/ϱ_A	$\overline{a}$	$\overline{b}$	$\overline{d}$
x_1	$(\overline{a},y_1)$	$(\overline{a},y_2)$	$(\overline{b},y_1)$
x_2	$(\overline{b},y_1)$	$(\overline{a},y_2)$	$(\overline{b},y_1)$

gde je sa $\overline{q}$ označena ϱ_A-klasa stanja $q \in A$.

Za dobijeni automat $\mathcal{A}/\varrho_A$ važi

$$\overline{\delta}(\overline{a},x_1) = \overline{a} = \overline{\delta}(\overline{b},x_1), \quad \overline{\lambda}(\overline{a},x_1) = y_1 \neq y_2 = \overline{\lambda}(\overline{b},x_1),$$

te je jasno da $\mathcal{A}/\varrho_A$ nije automat Mooreovog tipa. □

Zadatak 5.202. *Naći minimalni Mooreov automat automata* $\mathcal{A} = (A,X,Y,\delta,\lambda)$ *zadat tablicom (videti prethodni zadatak).*

A	y_1	y_1	y_2	y_2
	a	b	c	d
x_1	a	c	a	b
x_2	b	c	b	b

Rešenje: Formiraćemo najpre listu P svih parova stanja automata $\mathcal{A}$, a potom ćemo krenuti sa formiranjem relacija τ_k.

1. korak: Formirajmo relaciju τ_1. Iz tablice se vidi da ona ima dve klase: $\{a,b\}$ i $\{c,d\}$. Prema tome, sa liste brišemo sve parove iz skupa

$$\{a,b\} \times \{c,d\} \cup \{c,d\} \times \{a,b\}.$$

2. korak: Proveravamo parove koji su posle 1. koraka ostali na listi P ispod njene glavne dijagonale:

$(bx_1,ax_1) = (c,a)$ – nije na listi, pa se brišu parovi (b,a) i (a,b);
$(dx_1,cx_1) = (b,a)$ – nije na listi, pa se brišu parovi (d,c) i (c,d).

	a	b	c	d
a			×	×
b			×	×
c	×	×		
d	×	×		

posle 1. koraka
relacija τ_1

	a	b	c	d
a		×	×	×
b	×		×	×
c	×	×		×
d	×	×	×	

posle 2. koraka
relacija $\tau_A = \tau_2 = \Delta_A$

Ovim su obrisani svi parovi, osim onih na glavnoj dijagonali, što znači da se algoritam zaustavio i da je $\tau_A = \tau_2 = \Delta_A$, odnosno da je automat $\mathcal{A}$ redukovan kao Mooreov automat.

Na osnovu prethodnog zadatka primećujemo da ovaj automat nije redukovan kao automat Mealyevog tipa, jer se minimizacijom Mooreovog automata algoritmom za automate Mealyevog tipa dobija ekvivalentan automat sa manjim brojem stanja od ovog automata koji je dobijen korišćenjem algoritma za minimizaciju automata Mooreovog tipa. □

Zadatak 5.203. *Neka je $\mathcal{A} = \mathcal{A}/\tau_A = (A, X, Y, \delta, \lambda)$ (tj. $\tau_A = \Delta_A$) konačan Mooreov automat koji realizuje datu klasu automatovnih preslikavanja. Tada postoji njemu ekvivalentan automat Mealyevog tipa sa manjim brojem stanja ako i samo ako postoje dva različita stanja $a, b \in A$, takva da je $\delta(a, x) = \delta(b, x)$, za svaki $x \in X$. Dokazati.*

Rešenje: Pretpostavimo da postoji Mealyev automat $\mathcal{B} = (B, X, Y, \delta', \lambda')$ ekvivalentan automatu $\mathcal{A}$, takav da je $|B| < |A|$. Automat $\mathcal{A}/\varrho_A$ je minimalni automat u klasi automata ekvivalentnih sa $\mathcal{A}$, t.j $|A/\varrho_A| \leqslant |B| < |A|$. To znači da postoje bar dva stanja $a, b \in A$ takva da je $(a, b) \in \varrho_A$. Tada, za svaki $x \in X$ važi $\lambda(a, x) = \lambda(b, x)$, tj. $\mu(\delta(a, x)) = \mu(\delta(b, x))$. Prema definiciji kongruencije τ_A, zaključujemo da $\big(\delta(a, x), \delta(b, x)\big) \in \tau_A$ i kako je $\tau_A = \Delta_A$ imamo $\delta(a, x) = \delta(b, x)$, za svaki $x \in X$.

Obratno, neka su $a, b \in A$ različita stanja takva da je $\delta(a, x) = \delta(b, x)$, za svaki $x \in X$. Tada je $\mu(\delta(a, x)) = \mu(\delta(b, x))$, tj. $\lambda(a, x) = \lambda(b, x)$, što znači da su $(a, b) \in \varrho_A$. Kako je $a \neq b$ zaključujemo da $(a, b) \notin \tau_A = \Delta_A$, pa je jasno da važi $|A/\varrho_A| < |A/\tau_A| = |A|$. Ovim je tvrđenje dokazano. □

Literatura

1. J. A. Anderson, Diskretna matematika sa kombinatorikom, Računarski fakultet, Beograd, i CET, Beograd, 2005.
2. J. A. Anderson, Automata Theory with Modern Applications, Cambridge University Press, New York, 2006.
3. M. A. Arbib, Algebraic Theories of Abstract Automata, Prentice Hall, 1969.
4. M. A. Arbib (ed.), Algebraic Theory of Machines, Languages, and Semigroups, Academic Press, New York - London, 1968.
5. S. Bogdanović, M. Ćirić, Polugrupe, Prosveta, Niš, 1993.
6. S. Burris, H. P. Sankappanavar, A course in universal algebra, Springer-Verlag, New York, 1981.
7. J. Carroll, D. Long, Theory of Finite Automata with an Introduction to Formal Languages, Prentice Hall, Englewood Cliffs, New Jersey, 1989.
8. I. Chiswell, A Course in Formal Languages, Automata and Groups, Springer-Verlag, London, 2009.
9. M. Ćirić, T. Petković, S. Bogdanović, Jezici i automati, Prosveta, Niš, 2000.
10. J. H. Conway, Regular Algebra and Finite Machines, Chapman & Hall, London, 1971.
11. D. Cvetković, Teorija grafova i njene primene, Naučna knjiga, Beograd, 1990.
12. D. Cvetković, S. Simić, Diskretna matematika - matematika za kompjuterske nauke, Drugo izdanje, Prosveta, Niš, 1996.
13. N. Cutland, Computability - An Introduction to Recursive Function Theory, Cambridge University Press, Cambridge-New York, 1980.
14. I. Dolinka, Kratak uvod u analizu algoritama, Univerzitet u Novom Sadu, Prirodno-matemati?ki fakultet, Novi Sad, 2008.
15. S. Eilenberg, Automata, Languages and Machines, Volume A, Academic Press, New York - London, 1974.
16. S. Eilenberg, Automata, Languages and Machines, Volume B, Academic Press, New York - London, 1976.
17. S. S. Epp, Discrete Mathematics with Applications, Thomson - Brooks/Cole, 2004.
18. F. Gécseg, Products of automata, EATCS Monographs in Theor. Comput. Sci. Vol. 7, Springer-Verlag, Berlin-Haidelberg, 1986.
19. F. Gécseg, I. Peák, Algebraic Theory of Automata, Akadémiai Kiadó, Budapest, 1972.
20. A. Ginzburg, Algebraic theory of automata, Academic Press, New York, 1968.
21. S. Ginsburg, An introduction to mathematical machine theory, Addison-Wesley, Reading, Mass., 1962.

22. S. Ginsburg, The mathematical theory of context-tree languages, McGraw-Hill Book Company, New York, 1966.
23. S. Ginsburg, Algebraic and Automata - Theoretic Properties of Formal Languages, North-Holland, Amsterdam, 1975.
24. V. M. Glushkov, The Abstract Theory of Automata, Russian Mathematical Surveys, 16:1-53, 1961.
25. W. L. M. Holcombe, Algebraic Automata Theory, Cambridge University Press, Cambridge, 1982.
26. J. E. Hopcroft, R. Motwani, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, Reading, 2001.
27. J. E. Hopcroft, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading, 1979.
28. M. Ito, Algebraic Theory of Automata and Languages, World Scientific Publishing Co., Singapore, 2004.
29. B. Khoussainov, A. Nerode, Automata Theory and Its Applications, Birkhauser, Boston, 2001.
30. G. Lallement, Semigroups and Combinatorial Applications, Willey Interscience, New York, 1979.
31. M. V. Lawson, Finite Automata, Chapman & Hall / CRC Press, Boca Raton, 2004.
32. W. J. M. Levelt, An Introduction to the Theory of Formal Languages and Automata, John Benjamins Publishing Company, Amsterdam / Philadelphia, 2008.
33. P. Linz, An Introduction to Formal Languages and Automata, Jones and Bartlett Publisher, Sudbury, MA, 2001.
34. R. Sz. Madarász, S. Crvenković, Uvod u teoriju automata i formalnih jezika, Univerzitet u Novom Sadu, Novi Sad, 1995.
35. D. Perrin, J.-E. Pin, Infinite Words, Elsevier, 2004.
36. K. H. Rosen, Discrete Mathematics and Its Applications, Mc Graw Hill, 2003.
37. G. Rozenberg, A. Salomaa (eds), Handbook of Formal Languages, Vol 1,2,3, Springer-Verlag, Berlin, 1997.
38. J. Sakarovitch, Elements of Automata Theory, Cambridge University Press, Cambridge, 2009.
39. V. N. Saliĭ, Universal algebra and automata, Izd. Saratovskogo Univ., Saratov, 1988 (in Russian).
40. A. Salomaa, Theory of automata, Pergamon Press, Oxford, 1969.
41. A. Salomaa, Formal Languages, Academic Press, New York-London, 1973.
42. A. Salomaa, Computation and Automata, Cambridge Univ. Press, Cambridge, 1985.
43. B. Šešelja, Matematika informatike, Institut za Matematiku, Novi Sad, 1990.
44. B. Šešelja, A. Tepavčević, Algebra I, Univerzitet u Novom Sadu, Prirodno-matematički fakultet, Novi Sad, 2000.
45. D. Stevanović, S. Simić, M. Ćirić, V. Baltić, Diskretna matematika - Osnove kombinatorike i teorije grafova, Društvo matematičara Srbije, Beograd, 2007.
46. J. Shallit, A Second Course in Formal Languages and Automata Theory, Cambridge University Press, 2009.

Indeks pojmova

TEORIJA ALGORITAMA, AUTOMATA I JEZIKA

-zbirka zadataka-

Miroslav Ćirić • Jelena Ignjatović

ISBN 978-86-83481-87-3

9 788683 481873