TEORIJA JEZIKA I AUTOMATA

***

– AUTOMATI –

I DEO

Teorija automata je informatička disciplina koja se bavi izučavanjem apstraktnih uređaja za računanje, takozvanih "mašina".

Još pre pojave elektronskih računara, 1930-ih godina, britanski matematičar-informatičar Alan Tjuring (Alan Turing, 1912-1954), je definisao apstraktnu mašinu koja, u pogledu onoga što može izračunati, poseduje sve mogućnosti današnjih računara.

U njegovu čast, tu mašinu danas nazivamo Tjuringova mašina.

Tjuringov cilj bio je da precizno opiše granice između onoga što računske mašine mogu i onoga što ne mogu, odnosno, da odgovori na pitanja

- Koji problemi su algoritamski rešivi?
- Koji problemi su algoritamski nerešivi?

Zaključci do kojih je tom prilikom Tjuring došao primenljivi su ne samo na apstraktne Tjuringove mašine, već i na današnje realne računare.

U 1940-im i 1950-im godinama izučavan je jednostavniji tip automata, koje danas nazivamo **konačnim automatima**.

Ti automati su najpre uvedeni da bi se njime modelirale funkcije mozga, da bi se kasnije pokazali veoma efikasnim i u raznim drugim primenama, o kojima će biti reči kasnije.

U kasnim 1950-im, američki lingvist Noam Čomski (Noam Chomsky, 1928-), je uveo koncept formalne gramatike, koji smo već pominjali.

Iako formalne gramatike nisu apstraktne mašine, one su sa njima veoma tesno povezane, i danas se koriste kao osnova mnogih važnih softverskih komponenti, uključujući i delove kompajlera.

Klasu algoritamski rešivih problema je 1969. godine američki informatičar Stefan Kuk (Stephen Cook, 1939-) podelio u dve podklase:

- Problemi koji se zaista mogu praktično realizovati računarima;
- Problemi koji su u principu algoritamski rešivi, ali njihovo praktično rešavanje zahteva toliko mnogo vremena, da su računari beskorisni u svim, osim u malom broju najjednostavnijih slučajeva tih problema.

Problemi iz ove druge klase nazivaju se neobradivi problemi (engleski intractable), ili NP-teški problemi.

Čak ni današnji eksponencijalni rast brzine računara, poznat kao Murov zakon, neće imati značajan uticaj na našu mogućnost da rešavamo velike slučajeve neobradivih problema.

Na ovoj konstataciji temelji se i današnja kriptografija.

Sva ova pomenuta teorijska dostignuća imaju veliki uticaj na ono šta informatičari (computer scientists) danas rade.

Neki koncepti, kao što su konačni automati ili neke vrste formalnih gramatika, koriste se u dizajniranju i konstrukciji važnih tipova softvera.

Drugi koncepti, poput Tjuringovih mašina, pomažu nam da razumemo šta možemo očekivati od našeg softvera.

Zato su i osnovni zadaci ovog kursa sledeći:

- Da se upoznamo sa najvažnijim tipovima apstraktnih mašina;
- Da vidimo kakvi se problemi mogu rešavati određenim tipom mašina;
- Da se upoznamo sa praktičnim primenama tih mašina;
- Da uočimo gde su granice mogućnosti današnjih apstraktnih mašina i realnih računara, kao i da nazremo kako se te granice u budućnosti eventualno mogu pomeriti.

Neformalna slika automata

Najjednostavnija vrsta apstraktnih mašina su konačni automati.

Oni predstavljaju veoma koristan model za veoma važne tipove hardvera i softvera, kao što su:

- Softver za dizajniranje i proveru ponašanja digitalnih kola;
- Leksički analizatori kod kompajlera, tj. delovi kompajlera koji vrše analizu i podelu ulaznog teksta na logičke jedinice;
- Softver za skeniranje velikih porcija teksta, kao što su kolekcije Web strana, radi pronalaženja pojavljivanja izvesnih reči, fraza, i drugih šablona;
- Softver za verifikovanje sistema svih tipova koji imaju konačan broj različitih stanja, kao što su komunikacioni protokoli ili protokoli za bezbednu razmenu informacija, itd.

Iako se u praktičnim primenama koriste isključivo konačni automati, tj. automati sa konačnim brojem unutrašnjih stanja, iz metodoloških razloga u našim razmatranjima dozvoljavamo da broj unutrašnjih stanja bude i beskonačan.

U tom smislu, u daljem tekstu govorićemo samo **automat**, dok ćemo prefiks **konačan** koristiti samo u slučajevima kada je bitno naglasiti da se radi o automatu sa konačno mnogo unutrašnjih stanja.

Naše izučavanje automata počećemo neformalnom pričom, na primeru jednog od najjednostavnijih automata, a kasnije ćemo dati formalnu matematičku definiciju automata.

Neformalna slika automata

Primer 1. Verovatno najjednostavniji primer netrivijalnog automata je obični "on/off" prekidač koji koristimo u našim domovima za paljenje i gašenje svetla.

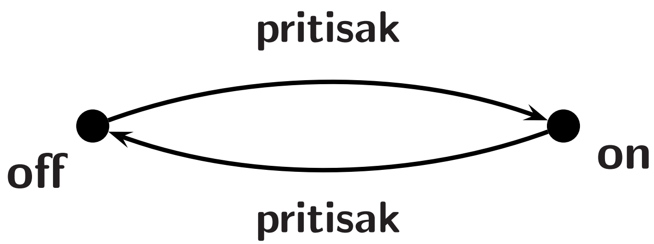
Prekidač ima svoju unutrašnju strukturu koju karakterišu dva njegova moguća unutrašnja stanja

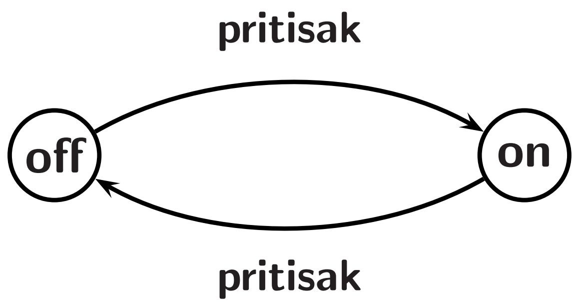
- "on" – "uključeno", i
- "off" – "isključeno".

U svakom diskretnom vremenskom trenutku prekidač se nalazi u tačno jednom od ta dva stanja.

Promena stanja vrši se kada korisnik pritisne dugme, efekat čega zavisi od trenutnog stanja prekidača – pritiskom na dugme se iz bilo kog od ta dva stanja prelazi u ono drugo.

Rad prekidača se grafički može prikazati na jedan od sledećih načina:





Prema tome, rad on/off prekidača zavisi od stanja u kome se on nalazi u datom trenutku, kao i od ulaznog signala (u ovom slučaju pritisak na dugme) koji u tom trenutku do prekidača dospeva spolja.

Neformalna slika automata

Iz prethodnog primera možemo izvući opštiju neformalnu sliku automata kao apstraktne mašine koja radi u diskretnoj vremenskoj skali, i čiji rad u svakom diskretnom vremenskom trenutku zavisi od

- unutrašnjeg stanja u kome se u datom trenutku mašina nalazi;
- ulaznog signala koji u datom trenutku dospeva do mašine.

Pod uticajem ulaznog signala mašina menja svoje unutrašnje stanje, prelaskom u neko drugo stanje.

Naravno, automat može imati proizvoljan broj različitih stanja, kao i proizvoljan broj različitih ulaznih signala.

Ovakva neformalna slika automata dovodi nas do formalne definicije automata koja se daje u nastavku.

Automat se definiše kao uređena trojka $\mathcal{A} = (A, X, \delta)$ za koju važi:

- A je neprazan skup, koji nazivamo **skup stanja**, a njegove elemente **stanjima** tog automata;
- X je neprazan skup, koji nazivamo **ulazni alfabet**, a njegove elemente **ulaznim signalima**, **simbolima** ili **slovima** tog automata;
- $\delta : A \times X \to A$ je preslikavanje koje nazivamo **funkcija prelaza** tog automata.

Primetimo da smo ovde i automat i njegov skup stanja označili istim slovom A, odnosno, poistovetili smo automat i njegov skup stanja.

To činimo u cilju pojednostavljenja oznaka, svuda gde ne postoji opasnost od zabune.

U slučajevima kada takva opasnost postoji, bićemo precizniji.

Princip rada ovako definisanog automata je sledeći:

- automat 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 svoje stanje, i u narednom trenutku prelazi u stanje $\delta(a, x) \in A$.

Ovo ćemo grafički predstavljati na jedan od sledeća dva načina:

Ovde je $b = \delta(a, x)$.

Primer 2. Razmotrimo još jednom "on/off" prekidač iz Primera 1.

Ako ulazni signal "pritisak na dugme" kraće označimo sa x, onda imamo da je

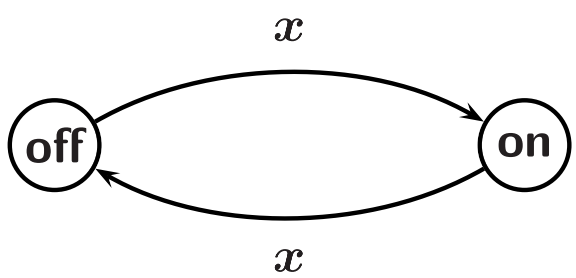
$$A = \{\text{on}, \text{off}\}, \qquad X = \{x\},$$

i funkcija prelaza $\delta : A \times X \to A$ je zadata sa

$$\delta(\text{on}, x) = \text{off}, \qquad \delta(\text{off}, x) = \text{on}.$$

To se može dati i tabelarno, odnosno grafički, na sledeće načine:

	off	on
x	on	off



Najprirodniji način predstavljanja automata je njihovo predstavljanje zadavanjem skupova i preslikavanja koji ga čine.

U tu svrhu koriste se uobičajeni metodi za predstavljanje skupova i preslikavanja.

To je posebno jednostavno kada se radi o konačnim automatima.

Konačne automate je takođe veoma zgodno zadavati i takozvanim **tablicama prelaza**, sličnim Kejlijevim tablicama koje se koriste za predstavljanje algebarskih struktura.

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

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

To je prikazano u sledećoj tablici

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

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

To je prikazano u sledećoj tablici

$\mathcal{A}$	

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

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

To je prikazano u sledećoj tablici

stanje

$\mathcal{A}$	$\dots$	a	$\dots$

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

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

To je prikazano u sledećoj tablici

stanje

$\mathcal{A}$	$\dots$	a	$\dots$
$\vdots$			
x			
$\vdots$			

ulazni simbol

Predstavljanje automata

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

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

To je prikazano u sledećoj tablici

stanje

$\mathcal{A}$	$\dots$	a	$\dots$
$\vdots$		$\vdots$	
x			
$\vdots$		$\vdots$	

ulazni simbol

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

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

To je prikazano u sledećoj tablici

stanje

$\mathcal{A}$	$\dots$	a	$\dots$
$\vdots$		$\vdots$	
x	$\dots$		$\dots$
$\vdots$		$\vdots$	

ulazni simbol

Predstavljanje automata

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

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

To je prikazano u sledećoj tablici

stanje

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

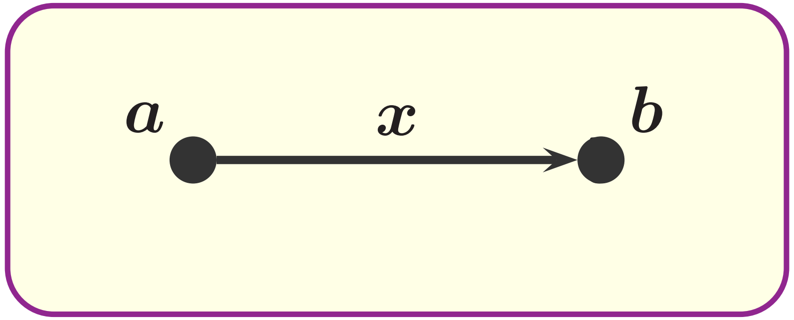
ulazni simbol

Predstavljanje automata

Drugi pogodan način za zadavanje automata, koji smo već koristili, je zadavanje pomoću grafova.

Graf prelaza automata $\mathcal{A} = (A, X, \delta)$ je označeni usmereni graf čiji čvorovi su stanja automata, oznake grana su ulazni simboli, pri čemu su grane i njihove oznake određene na sledeći način:

- ❏ ako se iz stanja $a \in A$ pod uticajem ulaznog simbola $x \in X$ prelazi u stanje $b\,(= \delta(a, x))$, tada graf prelaza ima **granu** (a, b) koja je **označena** sa x.



Napomena 1. Potsetimo se da se označeni orijentisani graf sa skupom čvorova A i skupom oznaka X formalno definiše kao trojka (A, X, E), gde je $E \subseteq A \times X \times A$ skup označenih grana.

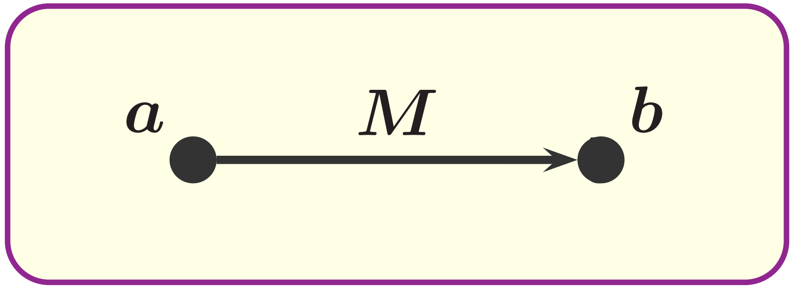
U tom smislu, za $a, b \in A$ i $x_1, x_2 \in X$, trojke

$$(a, x_1, b) \quad \text{i} \quad (a, x_2, b)$$

predstavljaju različite grane.

Ipak, sve grane sa istim početnim i krajnjim čvorom grafički predstavljamo jednom strelicom, iznad koje beležimo sve njihove oznake.

Naime, ako je M skup svih oznaka pridruženih grani (a, b), tada crtamo



Primer 3. Za automat $\mathcal{A} = (A, X, \delta)$, gde je

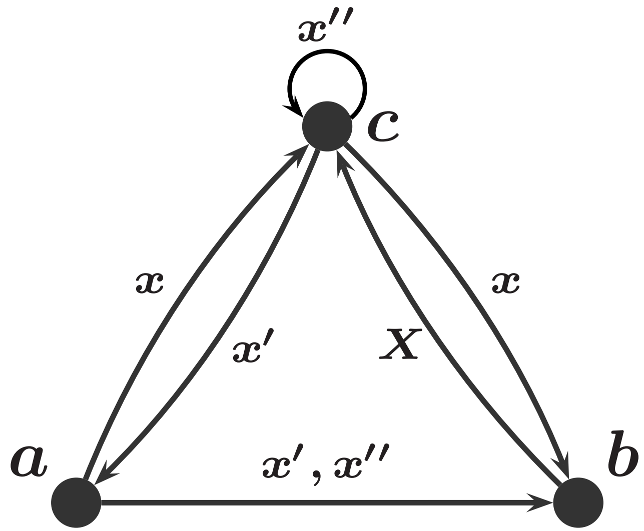
$$A = \{a, b, c\} \quad \text{i} \quad X = \{x, x', x''\},$$

a funkcija prelaza je definisana sa:

$$\begin{array}{c} \delta(a,x) = \delta(b,x) = \delta(b,x') = \delta(b,x'') = \delta(c,x'') = c, \\ \delta(a,x') = \delta(a,x'') = \delta(c,x) = b, \quad \delta(c,x') = a \end{array},$$

tablica i graf prelaza su

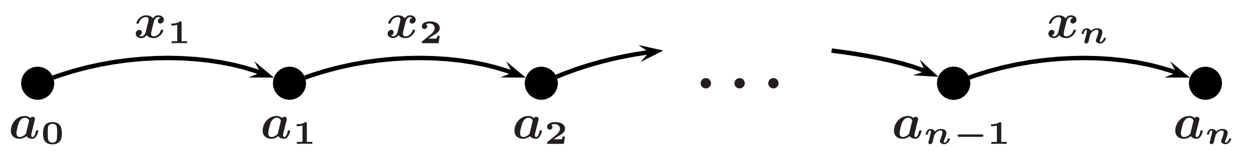
$\mathcal{A}$	a	b	c
x	c	c	b
x'	b	c	a
x''	b	c	c



Proširenje funkcije prelaza

Rad automata na 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.

To se može grafički predstaviti sa



Zato je prirodno razmatrati funkciju prelaza ne samo kao preslikavanje $\delta : A \times X \to A$, već radije kao preslikavanje $\widehat{\delta} : A \times X^* \to A$, za koje važi

- automat A iz stanja $a_0 \in A$, pod uticajem niza ulaznih simbola $u = x_1 x_2 \cdots x_n \in X^*$, prelazi u stanje $a_n = \widehat{\delta}(a_0, u) \in A$.

U nastavku ćemo videti da je takvo preslikavanje jednoznačno određeno preslikavanjem $\delta : A \times X \to A$.

Proširenje funkcije prelaza

Neka je dat automat $\mathcal{A} = (A, X, \delta)$.

Preslikavanje $\widehat{\delta} : A \times X^* \to A$ definišemo induktivno, na sledeći način:

(i) za proizvoljno stanje $a \in A$ i ulazno slovo $x \in A$ stavljamo da je

$$\widehat{\delta}(a, x) = \delta(a, x);$$

(ii) za proizvoljno stanje i $a \in A$ i praznu reč $e \in X^*$ stavljamo da je

$$\widehat{\delta}(a, e) = a;$$

(iii) za proizvoljno stanje $a \in A$, ulaznu reč $u \in X^*$ i slovo $x \in A$, ukoliko je definisano $\widehat{\delta}(a, u)$, onda se $\widehat{\delta}(a, ux)$ definiše sa

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

Drugim rečima, ova definicija kaže sledeće:

- Uslov (i) kaže da se preslikavanje $\widehat{\delta}$ poklapa sa δ na skupu $A \times X$, odnosno, da je $\widehat{\delta}$ **proširenje** funkcije prelaza δ sa $A \times X$ na $A \times X^*$.
- Značenje uslova (ii) je da se, kada na ulaz automata dođe prazna ulazna reč (prazan signal), u automatu ništa ne dešava.

  Dakle, prazna ulazna reč nema nikakav efekat na rad automata.
- Konačno, uslov (iii) kaže da ukoliko
  - ulazna reč u prevodi automat iz stanja a u stanje $b = \widehat{\delta}(a, u)$, i
  - ulazno slovo x prevodi automat iz stanja b u stanje $c = \delta(b, x)$,

  onda ulazna reč ux prevodi automat iz stanja a u stanje c, odnosno, $c = \widehat{\delta}(a, ux)$.

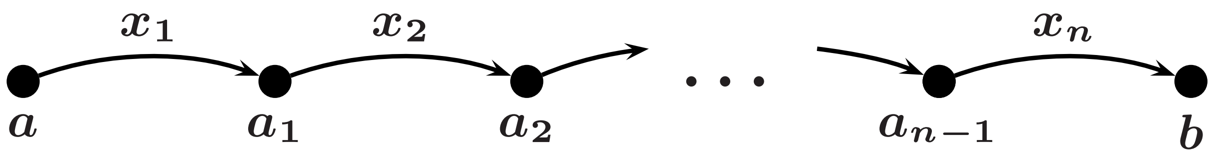
Neka je ulazna reč $u \in X^*$ predstavljena u obliku $u = x_1 x_2 \cdots x_n$, gde su $x_1, x_2, \ldots, x_n \in X$ ulazna slova, neka su $a, b \in A$, i neka je

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

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

Grafički je to prikazano na sledeći način:



Drugim rečima, ovom prelazu u grafu prelaza automata odgovara put iz čvora a u čvor b označen nizom simbola $u = x_1 x_2 \cdots x_n$.

Proširenje funkcije prelaza

Preslikavanje $\widehat{\delta}$ nazivamo **proširena funkcija prelaza** automata $\mathcal{A}$. Kako je ona potpuno određena funkcijom prelaza $\delta : A \times X \longrightarrow A$, neće biti zabune ako nadalje umesto $\widehat{\delta}$ pišemo samo δ.

Na osnovu definicije proširene funkcije prelaza, lako se dokazuje da važi:

Lema 1. Neka je dat automat $\mathcal{A} = (A, X, \delta)$.

Tada za proizvoljno stanje $a \in A$ i ulazne reči $u, v \in X^*$ važi

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

Ovo znači da se pod uticajem ulazne reči uv prelaz iz stanja a u stanje $b = \delta(a, uv)$ realizuje tako što

- iz stanja a, pod dejstvom reči u, se pređe u stanje $\delta(a, u)$, a potom
- iz stanja $\delta(a, u)$, pod dejstvom reči v, se pređe u stanje b.

Napomenimo i to da umesto $\delta(a, u)$ često pišemo kraće samo au.

Inicijalno stanje

Automat svoj rad uvek mora da započne iz nekog stanja.

U nekim slučajevima mi unapred fiksiramo stanje iz koga će automat krenuti sa radom, i takvo stanje nazivamo inicijalno stanje.

Ukoliko je to urađeno, odnosno unapred je određeno stanje a_0 koje će biti inicijalno, onda četvorku $\mathcal{A} = (A, a_0, X, \delta)$, gde

- (A, X, δ) **je automat, i**
- $a_0 \in A$ **je inicijalno stanje,**

nazivamo inicijalni automat.

Pri predstavljanju inicijalnog automata grafom prelaza, njegovo inicijalno stanje označavamo ulazećom strelicom, na primer

Tako, ako u primeru "on/off" prekidača uzmemo da stanje "off" bude inicijalno, onda taj automat predstavljamo na jedan od sledećih načina:

Kada inicijalni automat zadajemo njegovim grafom prelaza, u nekim slučajevima inicijalno stanje nećemo posebno isticati u tekstu, već ćemo ga označavati na samom grafu prelaza.

Završna stanja

Kao što smo videli, rad automata sastoji se iz niza uzastopnih prelaza iz stanja u stanje, pod dejstvom niza uzastopnih ulaznih signala.

Međutim, prirodno se nameće pitanje: Šta je rezultat tog rada?

Automati su mašine koje služe za procesiranje informacija.

Informacije se obično predstavljaju nizovima simbola (rečima) nad nekim alfabetom, i tipična mašina za procesiranje informacija to radi tako što niz ulaznih simbola (ulaznu reč) transformiše u niz izlaznih signala (izlaznu reč).

Takav vid procesiranja informacija vrše, na primer, takozvani automati sa izlazom ili transduktori, koji se definišu slično automatima kakve smo napred definisali, s tom razlikom što pored skupa stanja, ulaznog alfabeta i funkcije prelaza imaju i izlazni alfabet i funkciju izlaza.

Završna stanja

Automati koje ovde razmatramo nemaju izlaz, ali se ipak može zamisliti da i oni imaju izvesnu vrstu izlaza, koji se realizuje pomoću samo dva izlazna signala: "da" i "ne".

Naime, može se uzeti da svaka ulazna reč dovodi do toga da automat emituje jedan od ta dva izlazna signala.

Ako ona dovodi do emitovanja signala "da", onda kažemo da automat prepoznaje ili da prihvata tu reč.

U suprotnom, ukoliko dovodi do emitovanja signala "ne", onda kažemo da automat ne prepoznaje ili da odbacuje tu reč.

Međutim, i ovde se postavlja pitanje: Kako se ovi izlazi predstavljaju u strukturi automata kakvu smo zadali njihovom definicijom?

Videćemo da se to može učiniti pomoću posebne vrste stanja, koje nazivamo završnim stanjima automata.

Neka je dat automat $\mathcal{A} = (A, X, \delta)$.

Pored fiksiranja inicijalnog stanja $a_0 \in A$, u nekim slučajevima unapred fiksiramo i skup stanja $T \subseteq A$, koji nazivamo skup **završnih stanja** ili skup **terminalnih stanja**.

U tom slučaju petorku (A, a_0, X, δ, T) nazivamo **prepoznavač** (engl. **recognizer**) ili **prihvatač** (engl. **acceptor**).

Kao što smo to i ranije radili, i ovu petorku označavaćemo kraće sa $\mathcal{A}$.

Neka je $u \in X^*$ proizvoljna ulazna reč.

- Ako je $\delta(a_0, u) \in T$, onda kažemo da $\mathcal{A}$ **prepoznaje** ili **prihvata** reč u.
- U suprotnom, ako $\delta(a_0, u) \notin T$, onda kažemo da $\mathcal{A}$ **ne prepoznaje**, **ne prihvata** ili **odbacuje** reč u.

Završna stanja

Odavde se može videti kako automat može dati izlaz "da" ili "ne":

- ako je $\delta(a_0, u)$ završno stanje, onda je izlaz "da", i
- ako $\delta(a_0, u)$ nije završno stanje, onda je izlaz "ne".

Jednostavnosti radi, umesto raspoznavač ili prihvatač, u daljem radu govorićemo kraće automat, pri čemu ćemo imati u vidu da je fiksirano inicijalno stanje a_0 i skup završnih stanja T.

Kada automat predstavljamo grafom prelaza, onda finalna stanja označavamo duplim kružićima, na jedan od sledećih načina:

Proizvoljnom automatu $\mathcal{A} = (A, a_0, X, \delta, T)$ sa inicijalnim stanjem a_0 i skupom završnih stanja T možemo pridružiti jezik

$$L(\mathcal{A}) = \{w \in X^* \mid \delta(a_0, w) \in T\},$$

koji nazivamo **jezikom automata** $\mathcal{A}$.

Neka je dalje $L \subseteq X^*$ proizvoljan jezik.

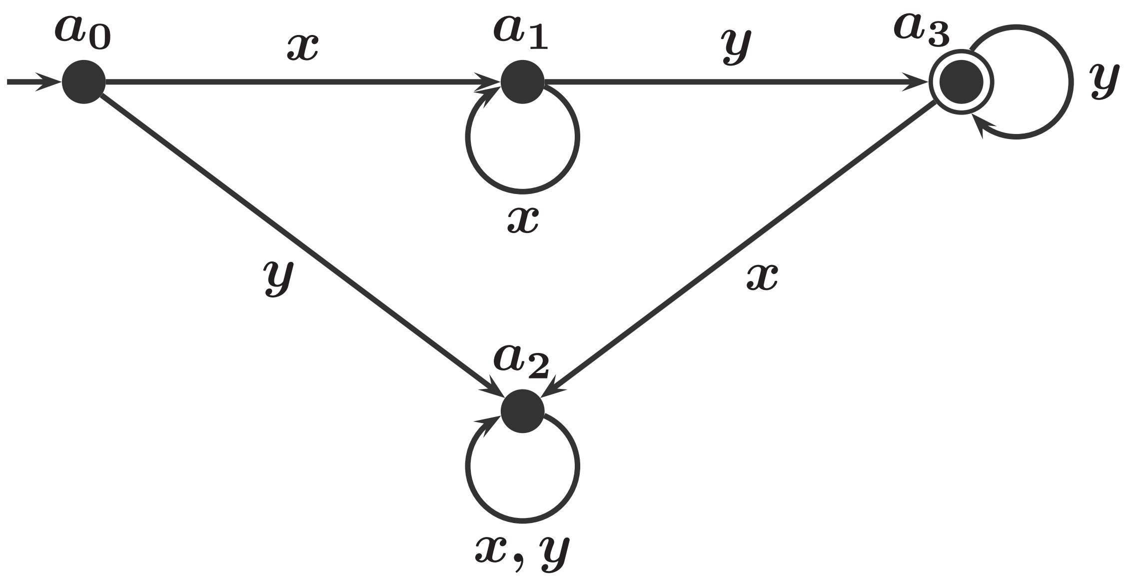
Ako je $L = L(\mathcal{A})$, onda kažemo da automat $\mathcal{A}$ **raspoznaje jezik** L, ili preciznije, da A **raspoznaje jezik** L **skupom** stanja T.

Drugim rečima, jezik koji automat $\mathcal{A}$ raspoznaje sastoji se od svih ulaznih reči koje taj automat prihvata (skupom završnih stanja T).

Za jezik L tada kažemo da **može biti raspoznat automatom** $\mathcal{A}$.

Često se kaže i da automat $\mathcal{A}$ **prihvata** jezik L.

Primer 5.1.1. Neka je automat A zadat grafom



Tada je $L(A) = \{x^m y^n \mid m, n \in \mathbb{N}\}$.

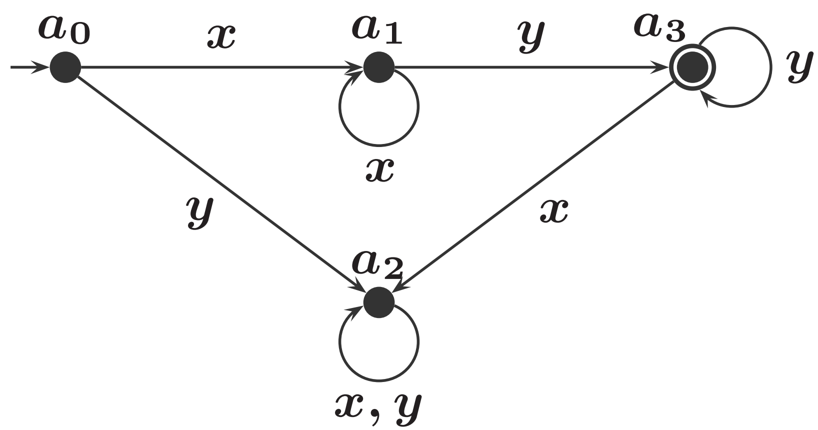
Dokaz: Uočimo najpre da je inicijalno stanje a_0, a skup završnih stanja je $T = \{a_3\}$.

Takođe, primetimo da je stanje a_2 takvo da, kada jednom uđemo u njega, onda je iz njega nemoguće izaći. Takvo stanje se naziva trap ili mrtvo stanje.

Sa slike se jasno vidi da proizvoljna reč oblika $x^m y^n$ vodi iz a_0 u a_3, što znači da je $x^m y^n \in L(A)$, za proizvoljne $m, n \in \mathbb{N}$.

Obratno, neka je $w \in L(A)$, tj. $\delta(a_0, w) = a_3$.

Ako je prvo slovo u reči w jednako y, onda iz a_0 ulazimo u stanje a_2, iz koga je nemoguće izaći, pa nikako ne možemo stići u a_3.



Prema tome, zaključujemo da reč w počinje sa x, pa postoji broj $m \in \mathbb{N}$ takav da je x^m najduži prefiks reči w koji ne sadrži slovo y.

To znači da se w može zapisati u obliku $w = x^m u$, za neku reč $u \in X^*$ koja ne počinje sa x. Kako w mora sadržati y, jer se bez tog slova ne može stići iz a_1 u a_3, to zaključujemo da u počinje sa y, pa postoji broj $n \in \mathbb{N}$ takav da je y^n najduži prefiks reči u koji ne sadrži slovo x.

Tada se u može zapisati u obliku $u = y^n v$, za neku reč $v \in X^*$ koja ne počinje sa y, pa je $w = x^m y^n v$. Kako reč $x^m y^n$ vodi iz a_0 u a_3, to reč v ne može počinjati ni sa x, jer bi nas onda slovo x odvelo iz a_3 u a_2, odakle nema povratka u a_3.

Dakle, v je prazna reč, odakle sledi da je $w = x^m y^n$, što je i trebalo dokazati. $\square$

Razlomci jezika

Naš sledeći cilj je da dokažemo da za proizvoljan jezik $L \subseteq X^*$ postoji automat koji ga raspoznaje, pri čemu taj automat ne mora biti konačan.

Kasnije ćemo videti da postoje i takvi jezici koji ne mogu biti raspoznati konačnim automatom.

Da bi smo dokazali postojanje automata koji raspoznaje dati jezik L, uvodimo pojam razlomka jezika.

Neka je dat jezik $L \subseteq X^*$ i reč $u \in X^*$.

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

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

U nekim izvorima umesto ovog naziva koristi se i naziv **izvod jezika**.

Lema 2. 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) $e \in L.u \;\Leftrightarrow\; u \in L$.

Dokaz: (i) Imamo da važi niz ekvivalencija

$$\begin{aligned} w \in (L.u).v &\Leftrightarrow vw \in L.u \\ &\Leftrightarrow u(vw) \in L \\ &\Leftrightarrow (uv)w \in L \\ &\Leftrightarrow w \in L.uv, \end{aligned}$$

odakle sledi da je $(L.u).v = L.uv$.

(ii) Prema definiciji razlomka imamo da je

$$\begin{aligned} w \in L.e &\Leftrightarrow ew \in L \\ &\Leftrightarrow w \in L, \end{aligned}$$

odakle zaključujemo da je $L.e = L$**.**

(iii) Imamo da je

$$\begin{aligned} e \in L.u &\Leftrightarrow eu \in L \\ &\Leftrightarrow u \in L, \end{aligned}$$

pa dakle, važi (ii). $\square$

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

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

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

$$T_L = \{L.u \mid u \in L\}.$$

Na osnovu tvrđenja (iii) u Lemi 2, T_L se može izraziti i sa:

$$T_L = \{H \in A_L \mid e \in H\}.$$

Pre no što pređemo na konstrukciju automata koji raspoznaje dati jezik $L \subseteq X^*$, koju dajemo korišćenjem razlomaka tog jezika, potrebno je da dokažemo i sledeće pomoćno tvrđenje

Lema 3. Neka je $L \subseteq X^*$ proizvoljan jezik, neka je $H \in A_L$ i $v \in X^*$. Tada je $H.v \in A_L$.

Dokaz: Uzmimo da je $H = L.u$, za neku reč $u \in X^*$.

Tada na osnovu tvrđenja (ii) u Lemi 2. imamo da je

$$H.v = (L.u).v = L.uv \in A_L,$$

što je i trebalo dokazati. $\square$

Neka je $L \subseteq X^*$ proizvoljan jezik.

Prema prethodnoj lemi, za $H \in A_L$ i $x \in X$, sa

$$\delta_L(H, x) = H.x. \tag{1.1}$$

je definisano preslikavanje $\delta_L : A_L \times X \to A_L$.

Dakle, $\mathcal{A}_L = (A_L, L, X, \delta_L, T_L)$ je automat sa inicijalnim stanjem L i skupom završnih stanja T_L.

Koristeći (1.1), indukcijom po dužini reči jednostavno dokazujemo da za proizvoljnu reč $v \in X^*$ važi

$$\delta_L(H, v) = H.v. \tag{1.2}$$

Teorema 1. Proizvoljan jezik $L \subseteq X^*$ može se raspoznati automatom

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

Dokaz: Neka je $w \in X^*$ proizvoljna reč.

Na osnovu (1.2) imamo da je $\delta_L(L, w) = L.w$, i takođe,

$$\begin{aligned} L.w \in T_L &\Leftrightarrow e \in L.w \\ &\Leftrightarrow we \in L \\ &\Leftrightarrow w \in L. \end{aligned}$$

Prema tome,

$$L(\mathcal{A}_L) = \{w \in X^* \mid \delta_L(L, w) \in T_L\} = \{w \in X^* \mid L.w \in T_L\} = L,$$

čime smo dokazali da $\mathcal{A}_L$ raspoznaje L. $\square$

Prirodno se postavlja pitanje: Kako odrediti sve razlomke datog jezika?

Jedan postupak za to baziran je na sledećoj teoremi:

Teorema 5.1.4. **Neka je $L \subseteq X^*$ proizvoljan jezik.**

Definišimo induktivno niz $\{A_k\}_{k\in\mathbb{N}^0}$ podskupova od A_L sa:

$$
\begin{aligned}
A_0 &= \{L\}, \\
A_{k+1} &= A_k \cup \{H.x \mid H \in A_k,\ x \in X\}, \qquad k \in \mathbb{N}^0.
\end{aligned}
\tag{1.3}
$$

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 $A_k = A_L$.

Dokaz: Tvrđenje (a) sledi neposredno iz (1.3).

(b) Podskup $A' \subseteq A_L$ koji sadrži L nazvaćemo zatvorenim za slovo $u \in X^*$ ako je $H.u \in A'$, za svaki element $H \in A'$.

Kako je svaki element iz A_L oblika $L.u$, za neku reč $u \in X^*$, i A' sadrži L, to je A' zatvoren za sve reči iz X^* ako i samo ako je $A' = A_L$.

Drugim rečima, A_L je jedini podskup od A_L zatvoren za sve reči iz X^*.

Sa druge strane, nije teško dokazati da je A' zatvoren za sve reči iz X^* ako i samo ako je zatvoren za sva slova iz X.

Kako jednakost $A_k = A_{k+1}$, za neki $k \in \mathbb{N}^0$ u stvari znači da je

$$\{H.x \mid H \in A_k,\ x \in X\} \subseteq A_k,$$

odnosno da je skup A_k zatvoren za sva slova iz X, to prema napred ustanovljenom imamo da je $A_k = A_L$, što je i trebalo dokazati.

(c) Kako je niz $\{A_k\}_{k\in\mathbb{N}^0}$ rastući, to je

$$|A_0| \leqslant |A_1| \leqslant \cdots \leqslant |A_k| \leqslant |A_{k+1}| \leqslant \cdots \leqslant |A_L|,$$

pa u slučaju da je A_L konačan skup dobijamo da je $|A_k| = |A_{k+1}|$, za neki $k \in \mathbb{N}^0$, i u tom slučaju je $A_k = A_{k+1}$, ponovo zbog toga što je niz $\{A_k\}_{k\in\mathbb{N}^0}$ rastući.

Prema tome, tada je $A_k = A_L$. $\square$

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

Demonstracija tog algoritma data je u sledećem primeru.

Primer 5.1.2. Razmotrimo još jednom jezik $L = \{x^m y^n \mid m, n \in \mathbb{N}\}$ nad alfabetom $X = \{x, y\}$ iz Primera 5.1.1.

Najpre određujemo skup A_1:

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

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:

$$L_1.x = L.x^2 = \{u \in X^* \mid x^2 u \in L\} = L \cup \{y\}^+ = L_1,$$
$$L_1.y = L.xy = \{u \in X^* \mid xyu \in L\} = \{y\}^*,$$
$$L_2.x = \emptyset.x = \emptyset = L_2,$$
$$L_2.y = \emptyset.y = \emptyset = L_2,$$

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

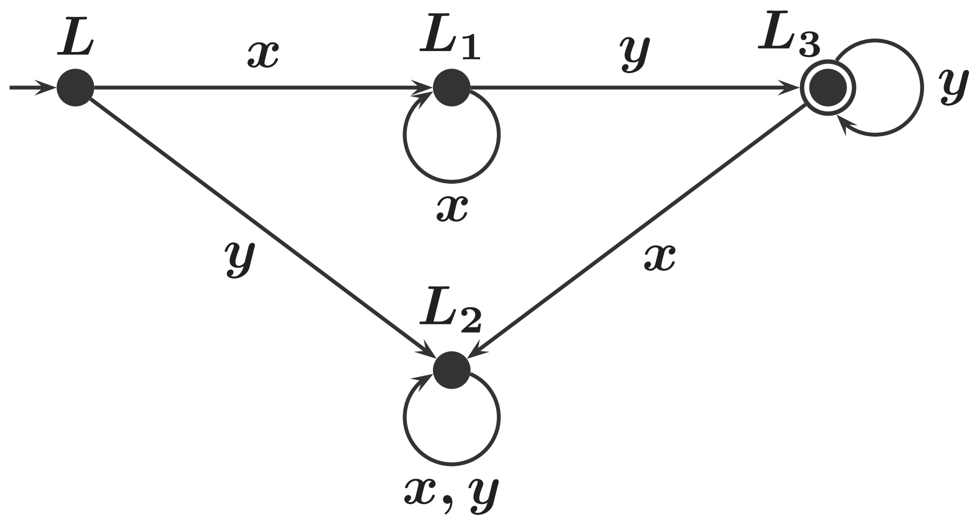
$$L_3.x = L.xyx = \{u \in X^* \mid xyxu \in L\} = \emptyset = L_2,$$

$$L_3.y = L.xy^2 = \{u \in X^* \mid xy^2u \in L\} = \{y\}^* = L_3,$$

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 A_L je zadat grafom:



Skup A_L svih razlomaka jezika L ne mora biti konačan, što pokazuje sledeći primer:

Primer 5.4.1. Neka je $X = \{x, y\}$ i $L = \{x^n y^n \mid n \in \mathbb{N}\}$.

Prvo, imamo da je

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

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

Zatim imamo da je

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

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

Dalje je

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

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

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

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

$$A_m = A_{m-1} \cup \{L_m, K_m\}, \tag{1.4}$$

gde su L_m i K_m zadati sa

$$\begin{aligned} L_m &= \{x^{n-m}y^n \mid n \geqslant m\}, \\ K_m &= \{y^{m-2}\}, \text{ za } m \geqslant 2, \qquad K_1 = \emptyset. \end{aligned} \tag{1.5}$$

To ćemo dokazati indukcijom.

Pretpostavimo da je (1.4) i (1.5) tačno. Tada je

$$L_m.x = \{u \in X^* \mid xu \in L_m\} = \{x^{n-m-1}y^n \mid n \geqslant m+1\} = L_{m+1},$$

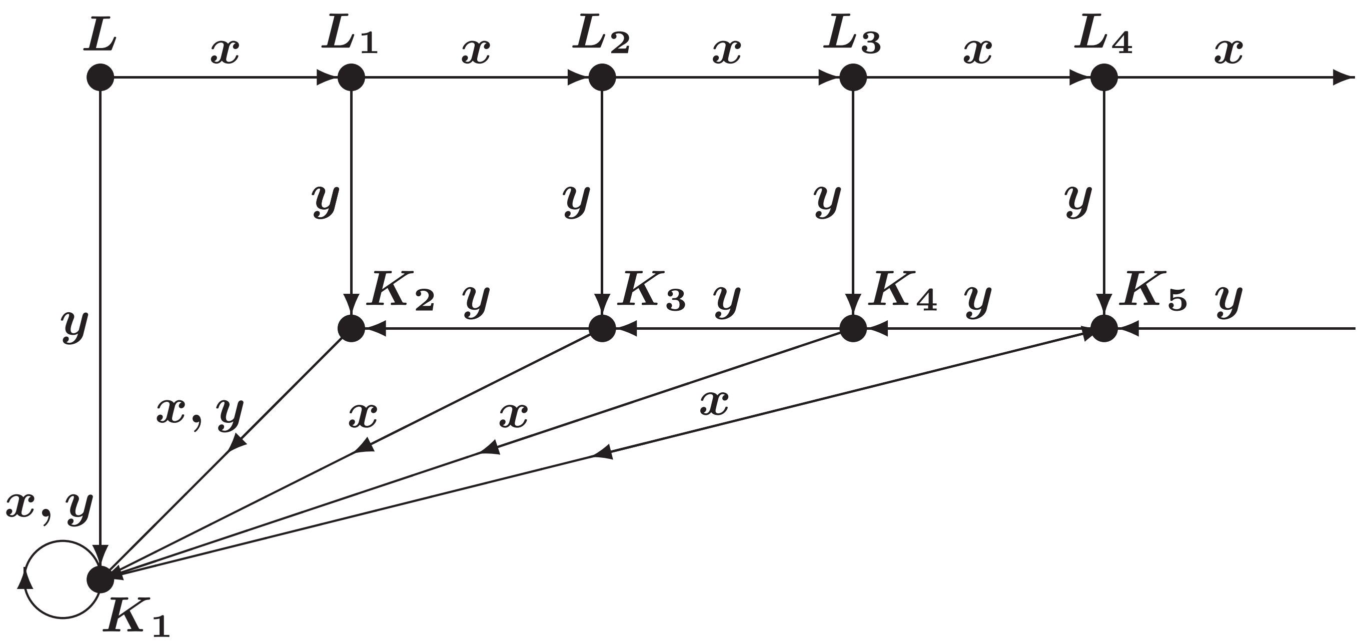
$$L_m.y = \{u \in X^* \mid yu \in L_m\} = \{y^{m-1}\} = K_{m+1},$$

$$K_m.x = \{u \in X^* \mid xu \in K_m\} = \{u \in X^* \mid xu = y^{m-2}\} = \emptyset = K_1,$$

$$K_m.y = \{u \in X^* \mid yu \in K_m\} = \{u \in X^* \mid yu = y^{m-2}\} = \{y^{m-1}\} = K_{m-1},$$

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

Iz svega ovog zaključujemo da razlomaka jezika $L = \{x^n y^n \mid n \in \mathbb{N}\}$ ima beskonačno mnogo, i da je A_L automat sa beskonačno mnogo stanja, koji se može grafički predstaviti na sledeći način:



Kako je K_2 jedini razlomak koji sadrži praznu reč, to je $T_L = \{K_2\}$, pa A_L raspoznaje jezik L tim stanjem K_2.

Teorema 2. Za jezik $L = \{x^n y^n \,|\, n \in \mathbb{N}\}$ nad alfabetom $X = \{x, y\}$ ne postoji konačan automat koji ga raspoznaje.

Dokaz: Pretpostavićemo suprotno, da postoji konačan automat $A = (A, a_0, X, \delta, T)$ koji raspoznaje L.

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 $m, n \in \mathbb{N}$, $m \neq n$. Međutim, onda dobijamo da je

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

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. $\square$

Kongruencije i homomorfizmi automata

Kada se bavimo dizajniranjem automata radi njihovih praktičnih primena, srećemo se sa dva osnovna problema:

- Da li za dati jezik postoji konačan automat koji ga raspoznaje?
- Kako konstruisati automat sa što je moguće manje stanja koji raspoznaje dati jezik?

Prvi problem je bitan jer u praktičnim primenama automata učestvuju samo konačni automati.

Beskonačni automati imaju samo teoretski značaj, i izučavaju se iz metodoloških razloga, jer je lakše raditi u teoriji u kojoj nismo sputani uslovom konačnosti skupa stanja automata.

Drugi problem je bitan iz razloga što veći broj stanja automata znači

- veći broj hardverskih komponenti, kada se radi o primeni automata u dizajniranju hardvera,
- glomaznije i sporije programe, kada se radi o primeni automata u dizajniranju softvera.

Mi ćemo se ovde pozabaviti najpre drugim problemom.

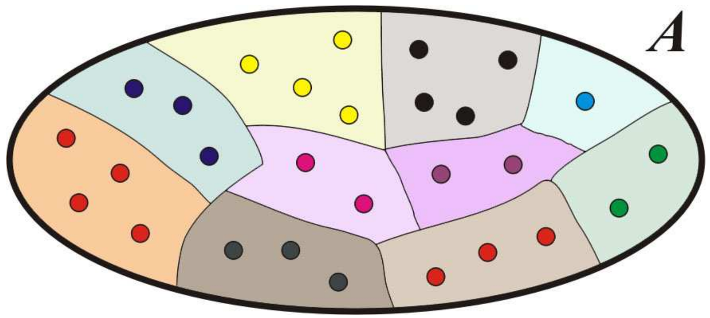
Naime, prvo ćemo dokazati da među svim automatima koji raspoznaju dati jezik postoji automat najmanje kardinalnosti, odnosno, automat sa najmanjim brojem stanja, ako se radi o konačnom automatu.

Štaviše, dokazaćemo da je takav upravo automat $\mathcal{A}_L$, kojim smo se bavili u prethodnom odeljku.

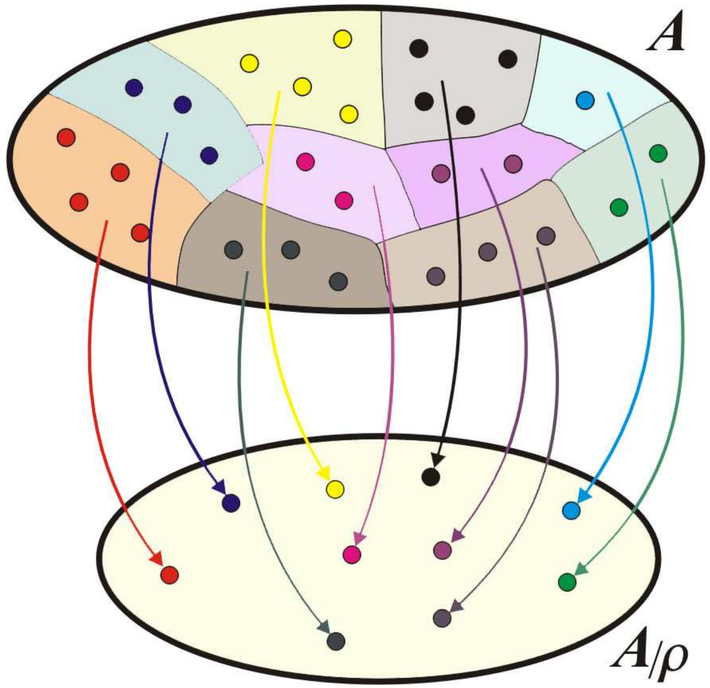
Pri redukciji broja stanja automata koji raspoznaje dati jezik koristimo ideje i metode koji dolaze iz algebre, odnosno, dva centralna algebarska koncepta:

- kongruencije, i
- homomorfizme.

Setimo se iz prethodnih kurseva 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.



Od klasa ekvivalencije relacije ekvivalencije ϱ, koje sada tretiramo kao individualne elemente, formiramo skup A/ϱ, koji nazivamo **faktor skup** skupa A u odnosu na ϱ.

Kao što se vidi sa slike desno, faktor skup se zapravo dobija tako što se svaka ϱ-klasa sažme u jedan element.

Osnovni smisao upotrebe relacija ekvivalencija je upravo u tom prevođenju skupa A na skup A/ϱ sa manjim brojem elemenata, odnosno, u redukovanju broja elemenata iz A.

Pri tome A/ϱ treba da sačuva osnovna svojstva elemenata iz A, jer se klase iz A/ϱ sastoje od elemenata iz A sa istim svojstvima.

Kongruencije i homomorfizmi automata

Neka sada A ima izvesnu algebarsku strukturu, odnosno, neka je na A definisan izvestan sistem operacija.

Da bi operacije sa A mogli da na prirodan način prenesemo na faktor skup A/ϱ, onda nije dovoljno da ϱ bude samo relacija ekvivalencije.

Da bi se to učinilo, potrebno je da relacija ekvivalencije ϱ bude **saglasna** sa operacijama na A, i takve relacije ekvivalencije zovemo **kongruencije**.

Prema tome, 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.

Kongruencije i homomorfizmi automata

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 $\mathcal{A} = (A, X, \delta)$ i relacija ekvivalencije ϱ na skupu stanja A tog automata.

Za ϱ kažemo da je kongruencija na automatu $\mathcal{A}$ ako je saglasna sa funkcijom prelaza δ, odnosno ako važi sledeće:

- za proizvoljne $a, b \in A$, iz $(a, b) \in \varrho$ sledi

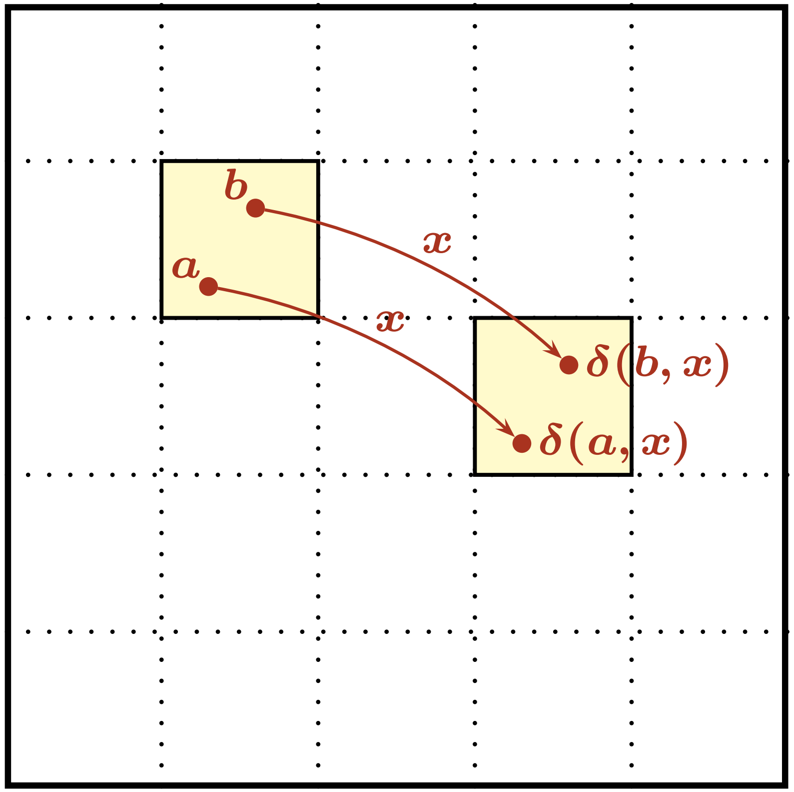
$$(\delta(a, x), \delta(b, x)) \in \varrho, \quad \text{za svaki } x \in X.$$

Indukcijom se lako dokazuje da prethodni uslov povlači uslov

- za proizvoljne $a, b \in A$, iz $(a, b) \in \varrho$ sledi

$$(\delta(a, u), \delta(b, u)) \in \varrho, \quad \text{za svaki } u \in X^*.$$

Značenje saglasnosti je grafički prikazano na sledećoj slici.



Naime, relacija ekvivalencije ϱ je saglasna sa funkcijom prelaza ako za proizvoljna stanja a i b iz iste ϱ-klase imamo da su i stanja $\delta(a, x)$ i $\delta(b, x)$ u istoj ϱ-klasi.

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

Na faktor-skupu A/ϱ definišemo preslikavanje

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

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.

Koristeći činjenicu da je ϱ kongruencija na A, lako se proverava da je preslikavanje δ_ϱ dobro definisano, tj. da njihove vrednosti ne zavise od izbora predstavnika ϱ-klasa.

Dakle, $A/\varrho = (A/\varrho, X, \delta_\varrho)$ je automat koji zovemo **faktor-automat** automata A u odnosu na kongruenciju ϱ.

Za dva automata kažemo da su istog tipa ako imaju iste ulazne alfabete.

Neka su $\mathcal{A} = (A, X, \delta)$ i $\mathcal{A}' = (A', X, \delta')$ automati istog tipa.

Preslikavanje $\varphi : A \to A'$ zovemo homomorfizam automata $\mathcal{A}$ u automat $\mathcal{A}'$ ako za proizvoljne $a \in A$ i $x \in X$ važi:

$$(\delta(a, x))\varphi = \delta'(a\varphi, x). \tag{1.6}$$

Indukcijom se lako dokazuje da ako je φ homomorfizam, onda za proizvoljno stanje $a \in A$ i ulaznu reč $u \in X^*$ važi

$$(\delta(a, u))\varphi = \delta'(a\varphi, u).$$

Ako su $\mathcal{A} = (A, a_0, X, \delta)$ i $\mathcal{A}' = (A', a_0', X, \delta')$ inicijalni automati, onda je $\varphi : A \to A'$ homomorfizam ako pored uslova (1.6) važi i

$$a_0\varphi = a_0'.$$

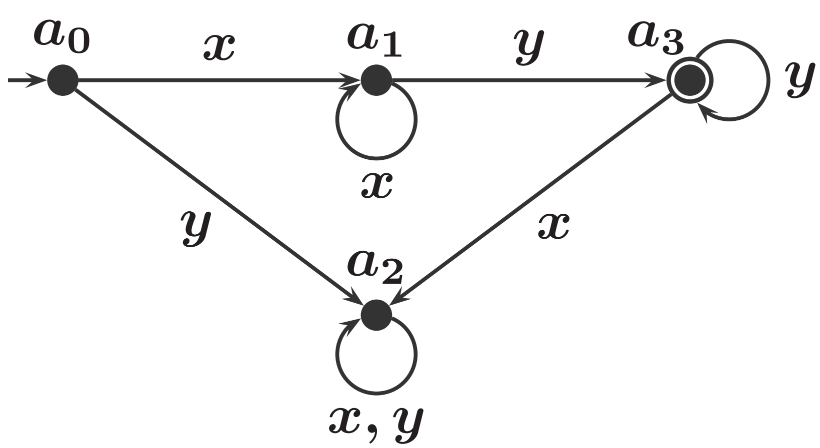
Ako je $\varphi : A \longrightarrow A'$ surjektivni homomorfizam, onda ga zovemo **epimorfizam** automata A na automat A'.

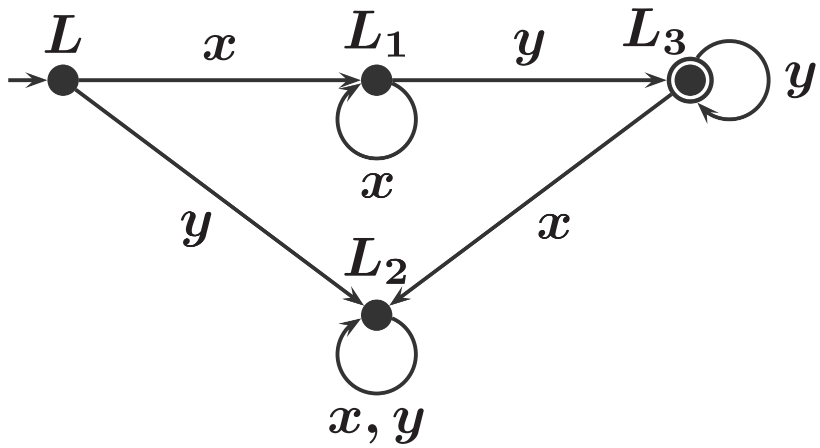
U tom slučaju automat A' zovemo **homomorfna slika** automata A.

Ako je $\varphi : A \longrightarrow A'$ bijektivni homomorfizam, onda ga zovemo **izomorfizam** automata A i A'.

U tom slučaju kažemo da su A i A' **izomorfni automati**.

Primer 4. Razmotrimo automate iz Primera 5.1.1 i 5.1.2.





Ovi automati su izomorfni, pri čemu se izomorfizam se može zadati sa

$$\varphi = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ L & L_1 & L_2 & L_3 \end{pmatrix}$$

Inače, izomorfne automate možemo shvatiti kao automate koji su dobijeni jedan iz drugog jednostavnom promenom oznaka stanja, dok je sve ostalo isto.

Kao i kod algebri, i kod automata postoji veza između kongruencija i homomorfizama koja je data sledećom Teoremom o homomorfizmu:

Teorema 2. Ako je ϱ kongruencija na automatu A, tada prirodno preslikavanje $\varrho^\natural : A \longrightarrow A/\varrho$, definisano sa

$$\varrho^\natural : a \mapsto a\varrho, \quad \text{za svaki } a \in A,$$

jeste epimorfizam automata A na faktor-automat A/ϱ.

Obratno, ako je φ epimorfizam automata A na automat A', tada njegovo jezgro $\ker\varphi$, tj. relacija na A definisana sa

$$(a,b) \in \ker\varphi \quad \Leftrightarrow \quad a\varphi = b\varphi, \quad \text{za sve } a, b \in A,$$

jeste kongruencija na A, i faktor automat $A/\ker\varphi$ je izomorfan automatu A'.

Sada prelazimo na dokaz tvrđenja da za proizvoljan jezik $L \subseteq X^*$, A_L je automat najmanje kardinalnosti, među automatima koji raspoznaju L (odnosno, sa najmanjim brojem stanja, ako je konačan).

Prvo uvodimo novi pojam koji nam je potreban.

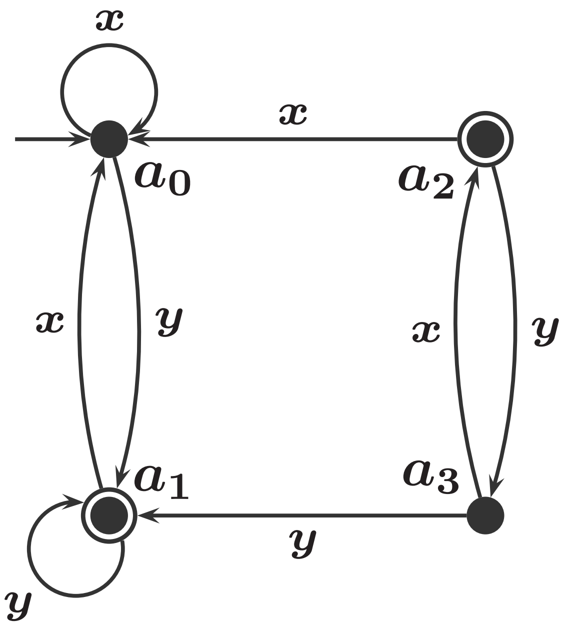
Neka je dat automat $A = (A, a_0, X, \delta, T)$ sa inicijalnmim stanjem a_0 i skupom završnih stanja T.

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 u njega može stići iz inicijalnog stanja, odnosno, ako u grafu prelaza tog automata postoji put iz inicijalnog stanja a_0 u stanje a.

Automat zovemo dostižan automat ako su sva njegova stanja dostižna.

Primer 5. Neka je automat A dat sledećim grafom:



Dostižna stanja ovog automata su a_0 i a_1, jer je, na primer,

$$a_0 = \delta(a_0, e) \quad \text{i} \quad a_1 = \delta(a_0, y).$$

Stanja a_2 i a_3 su nedostižna, jer je očigledno da se ni do jednog od njih ne može stići iz inicijalnog stanja a_0.

Dostižni deo automata

Razmotrimo ponovo automat $\mathcal{A} = (A, a_0, X, \delta, T)$.

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 $\mathcal{A}$;
- $\delta^d : A^d \times X \to A^d$ 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 $\mathcal{A}$.

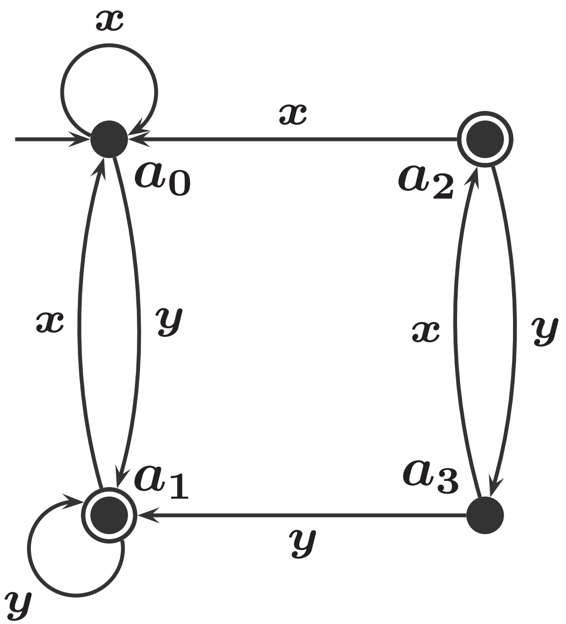
Uočimo da je $a_0 \in A^d$ jer je inicijalno stanje automata uvek dostižno.

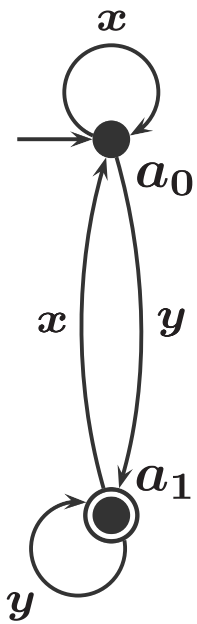
Za proizvoljne $a \in A^d$ i $x \in X$ imamo da je $a = \delta(a_0, u)$, za neku reč $u \in X^*$, pa je $\delta(a, x) = \delta(\delta(a_0, u), x) = \delta(a_0, ux)$, što znači da je $\delta(a, x) \in A^d$.

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

Jasno, dostižni deo automata je dostižan automat.

Primer 6. Za automat A iz prethodnog primera, na slici levo, njegov dostižni deo A^d je automat dat na slici desno:





Kao što se vidi sa ovih slika, dostižni deo A^d se dobija iz automata A na veoma jednostavan način: brisanjem svih njegovih nedostižnih stanja i svih prelaza koji polaze iz ili se završavaju u nedostižnom stanju.

Dostižni deo automata

Koja je svrha razmatranja dostižnog dela automata?

Kako svaki automat počinje svoj rad iz inicijalnog stanja, u daljem radu će se uvek nalaziti u dostižnom stanju.

Prema tome, nedostižna stanja ni na koji način ne utiču na rad automata.

Zbog toga ih možemo slobodno odbaciti, zajedno sa prelazima koji polaze iz njih ili se završavaju u njima.

Odbacivanjem nedostižnih stanja ništa ne menjamo u radu automata, a pri tome automat pojednostavljujemo, smanjujući mu broj stanja.

Sve ove konstatacije formalno dokazujemo u nerednoj teoremi.

Teorema 3. Za proizvoljan automat A i njegov dostižni deo A^d je

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

Dokaz: Uzmimo da je $A = (A, a_0, X, \delta, T)$.

Neka je $u \in L(A^d)$, odnosno, $\delta^d(a_0, u) = a \in T^d$. To znači da je $\delta(a_0, u) = a \in T$, i prema tome, $u \in L(A)$. Dakle, $L(A^d) \subseteq L(A)$.

Obratno, neka je $u \in L(A)$. To znači da je $\delta(a_0, u) = a \in T$, i jasno je da stanje a i sva međustanja u prelazu iz a_0 u a pod dejstvom ulazne reči u jesu dostižna stanja.

Prema tome, $\delta^d(a_0, u) = \delta(a_0, u) = a \in T \cap A^d = T^d$, odakle sledi da je $u \in L(A^d)$.

Ovim smo dokazali da je $L(A) \subseteq L(A^d)$, i dakle, $L(A) = L(A^d)$. $\square$

Sada smo spremni da dokažemo postojanje automata sa minimalnim brojem stanja koji raspoznaje dati jezik. Najpre dokazujemo sledeće:

Teorema 4. Neka je $\mathcal{A} = (A, a_0, X, \delta, T)$ automat koji raspoznaje jezik $L \subseteq X^*$, i $\mathcal{A}^d$ je njegov dostižni deo.

Tada je automat $\mathcal{A}_L$ homomorfna slika automata $\mathcal{A}^d$, odnosno, postoji epimorfizam automata $\mathcal{A}^d$ na automat $\mathcal{A}_L$.

Dokaz: Neka je $a \in A^d$ proizvoljno stanje.

Prema definiciji dostižnog stanja imamo da postoji reč $u \in X^*$ tako da je $a = \delta(a_0, u)$, i u tom slučaju ćemo staviti da je

$$a\varphi = L.u.$$

Dokazaćemo da je φ traženi surjektivni homomorfizam iz $\mathcal{A}^d$ na $\mathcal{A}_L$.

Prvo treba dokazati da je φ dobro definisano preslikavanje iz A^d u A_L, odnosno, da ne zavisi od izbora reči u koja zadovoljava $a = \delta(a_0, u)$.

Naime, neka su $u, v \in X^*$ bilo koje reči za koje važi $a = \delta(a_0, u)$ i $a = \delta(a_0, v)$. Dokazaćemo da je tada $L.u = L.v$.

Zaista, za proizvoljnu reč $w \in X^*$ važi

$$\delta(a_0, uw) = \delta(\delta(a_0, u), w) = \delta(a, w) = \delta(\delta(a_0, v), w) = \delta(a_0, vw),$$

odakle dobijamo sledeći niz ekvivalencija

$$\begin{aligned} w \in L.u \Leftrightarrow uw \in L \Leftrightarrow \delta(a_0, uw) \in F \\ \Leftrightarrow \delta(a_0, vw) \in F \Leftrightarrow vw \in L \Leftrightarrow w \in L.v. \end{aligned}$$

Prema tome, $L.u = L.v$, čime smo dokazali da je φ dobro definisano preslikavanje iz A^d u A_L.

Dalje, proizvoljni element iz A_L je razlomak oblika $L.u$, za neku reč $u \in X^*$, i ako stavimo da je $a = \delta(a_0, u)$, dobijamo da je $a\varphi = L.u$.

Prema tome, φ je surjektivno preslikavanje.

Konačno, da bi dokazali da je φ homomorfizam, treba dokazati da je

$$\big(\delta^d(a, x)\big)\varphi = \delta_L(a\varphi, x),$$

za proizvoljne $a \in A^d$ i $x \in X$.

Zaista, kako je $a = \delta^d(a_0, u)$, za neku reč $u \in X^*$, to je

$$\begin{aligned}\big(\delta^d(a, x)\big)\varphi &= \big(\delta^d(\delta^d(a_0, u), x)\big)\varphi = \big(\delta^d(a_0, ux)\big)\varphi \\ &= L.ux = \delta_L(L.u, x) = \delta_L(a\varphi, x).\end{aligned}$$

Dakle, φ je epimorfizam automata A^d na automat A_L, čime je dokaz teoreme završen. $\square$

Neposredno iz prethodne teoreme, dobijamo sledeću teoremu:

Teorema 5. Za proizvoljan jezik $L \subseteq X^*$, automat A_L je automat najmanje kardinalnosti među automatima koji raspoznaju L.

Dokaz: Setimo se da se kardinalnost skupa S označava sa $|S|$.

Neka je A proizvoljni automat koji raspoznaje L.

Prema prethodnoj teoremi, postoji surjektivno preslikavanje iz A^d na A_L, što znači da je $|A_L| \leqslant |A^d|$.

Sa druge strane, A^d je podskup od A, odakle je $|A^d| \leqslant |A|$.

Dakle, iz $|A_L| \leqslant |A^d|$ i $|A^d| \leqslant |A|$ zaključujemo da je $|A_L| \leqslant |A|$.

Time je teorema dokazana. $\square$

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

Prema prethodnoj teoremi, za svaki jezik L postoji bar jedan takav automat – automat A_L.

Drugim rečima, svaki jezik ima minimalni automat, koji može biti konačan, ali i beskonačan.

Ako je A minimalni automat jezika L, onda ne postoji automat manje kardinalnosti od njega koji raspoznaje isti jezik L.

Međutim, to ne znači da ne postoji automat iste kardinalnosti kao automat A koji raspoznaje L, odnosno, da ne postoji više minimalnih automata jezika L iste kardinalnosti.

Ipak, ukoliko jezik L ima minimalni automat koji je konačan, onda je on jedinstven do na izomorfizam, odnosno, svaka dva minimalna automata jezika L su izomorfna.

Naime, važi sledeće:

Teorema 6. Neka je $L \subseteq X^*$ jezik takav da automat A_L ima n stanja, gde je n prirodan broj.

Tada je svaki automat sa n stanja koji raspoznaje L izomorfan sa A_L.

Dokaz: Neka je A proizvoljan automat sa n stanja koji raspoznaje L.

Tada je A^d takođe automat koji raspoznaje L, i broj stanja mu je manji ili jednak n, jer je $A^d \subseteq A$.

Kako automat koji raspoznaje L ne može imati manje od n stanja, zaključujemo da i A^d ima n stanja.

Prema Teoremi 4. imamo da postoji surjektivni homomorfizam φ iz A^d na A_L.

Međutim, znamo da svaka surjekcija između konačnih skupova sa istim brojem stanja mora biti bijekcija, odakle zaključujemo da je i φ bijekcija.

Prema tome, φ je izomorfizam automata A^d na automat A_L. $\square$

Dakle, na osnovu prethodne teoreme, svaki automat sa n stanja koji raspoznaje L je izomorfan sa A_L, i prema tome, svaka dva automata sa n stanja koji raspoznaju L su izomorfna.

Recimo, za jezik $L = \{x^m y^n \mid m, n \in \mathbb{N}\}$, oba automata iz Primera 4. su njegovi minimalni automati, i izomorfni su.

Minimizacija automata

U prethodnim razmatranjima videli smo kako se nalazi minimalni automat A_L datog jezika L.

U ovom odeljku razmatraćemo drugačiju situaciju, kada minimalni automat jezika L treba konstruisati polazeći ne od jezika L, već od nekog automata 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 A_L jezika L, polazeći od automata A koji raspoznaje L, naziva se minimizacija automata A.

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

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

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

$$T.a = \{u \in X^* \mid \delta(a, u) \in T\}. \tag{1.7}$$

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

Primetimo da, iako je T skup stanja a a je stanje automata $\mathcal{A}$, razlomak $T.a$ nije skup stanja, već jezik u X^*.

Skup svih razlomaka skupa T označavaćemo sa $\mathcal{A}_T$.

Glavna svojstva ovakvih razlomaka, i veza sa razlomcima jezika, data su narednom teoremom.

Teorema 5.2.1. **Neka je dat automat $\boldsymbol{A = (A, a_0, X, \delta, T)}$. Tada za proizvoljne $\boldsymbol{a \in A}$ i $\boldsymbol{u \in X^*}$ važi**

$$\boldsymbol{T.\delta(a,u) = (T.a).u\,.} \tag{1.8}$$

Osim toga, ako je $\boldsymbol{A}$ dostižan automat i $\boldsymbol{L}$ je jezik koji taj automat raspoznaje, tada je

$$\boldsymbol{A_L = A_T,}$$

tj., skup $\boldsymbol{A_L}$ svih razlomaka jezika $\boldsymbol{L}$ jednak je skupu $\boldsymbol{A_T}$ svih razlomaka skupa $\boldsymbol{T}$.

Dokaz: Uzmimo proizvoljne $a \in A$, $u \in X^*$.

Tada za proizvoljnu reč $v \in X^*$ imamo da je

$$\begin{aligned} v \in T.\delta(a,u) &\Leftrightarrow \delta(\delta(a,u),v) \in T && \text{(def. razlomka skupa stanja)} \\ &\Leftrightarrow \delta(a,uv) \in T && \text{(jer } \delta(\delta(a,u),v) = \delta(a,uv)\text{)} \\ &\Leftrightarrow uv \in T.a && \text{(def. razlomka skupa stanja)} \\ &\Leftrightarrow v \in (T.a).u && \text{(def. razlomka jezika),} \end{aligned}$$

što dokazuje (1.8).

Uzmimo dalje da je $\mathcal{A} = (A, a_0, X, \delta, T)$ dostižan automat koji raspoznaje jezik L.

Jasno da je $L = T.a_0$.

Prema (1.8), za proizvoljnu reč $u \in X^*$ je

$$L.u = (T.a_0).u = T.\delta(a_0, u),$$

pa je $A_L \subseteq A_T$ podskup skupa razlomaka skupa T.

Obratno, kako je, prema pretpostavci, svako stanje $a \in A$ dostižno, to je $a = \delta(a_0, u)$, za neku reč $u \in X^*$, pa opet prema (1.8) imamo da je

$$T.a = T.\delta(a_0, u) = (T.a_0).u = L.u,$$

pa je dakle $A_T \subseteq A_L$.

Dakle, $A_L = A_T$, čime je dokaz teoreme završen. $\square$

Setimo se da se algoritam za konstrukciju minimalnog automata datog jezika $L \subseteq X^*$ zasnivao na nalaženju svih razlomaka jezika L.

On bi se u istom obliku mogao 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 A sastoji se u sledećem:

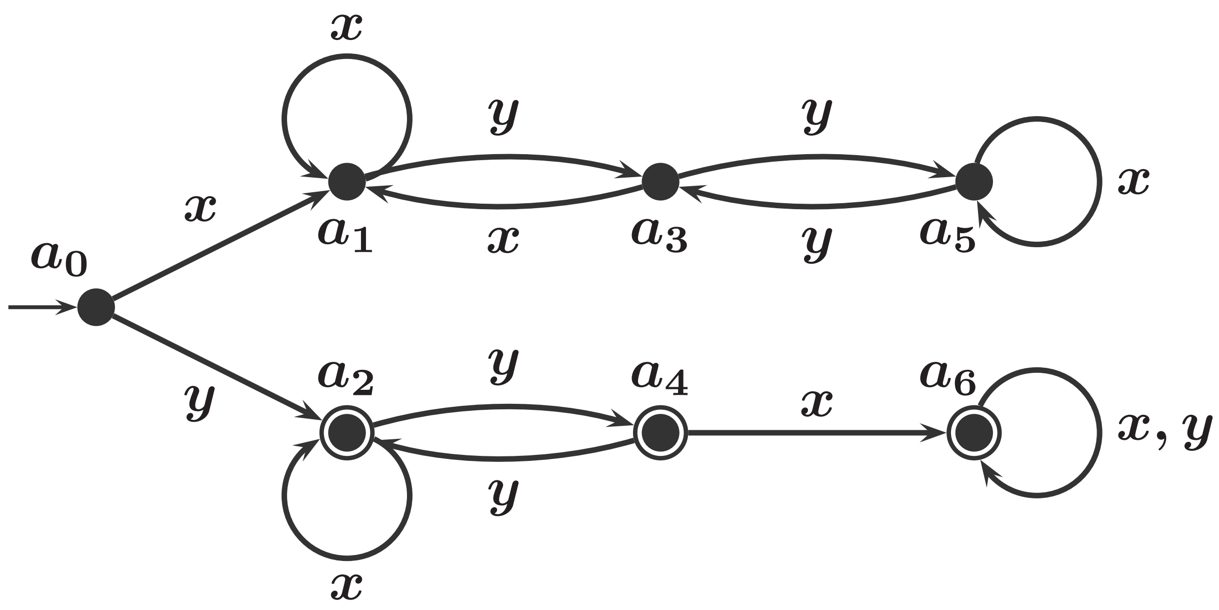
1. Najpre nalazimo dostižni deo A^d automata A.

   Kao što znamo, automat A^d raspoznaje L skupom $T^d = T \cap A^d$.

2. U automatu A^d nalazimo skup razlomaka skupa T^d.

   Kako je A^d dostižan automat, to prema prethodnoj teoremi imamo da je skup razlomaka skupa stanja T^d jednak skupu A_L razlomaka jezika L.

Primer 5.2.1. Neka je $X = \{x, y\}$, $A = \{a_0, a_1, a_2, a_3, a_4, a_5, a_6\}$, neka je automat $\mathcal{A} = (A, a_0, X, \delta, T)$ zadat grafom

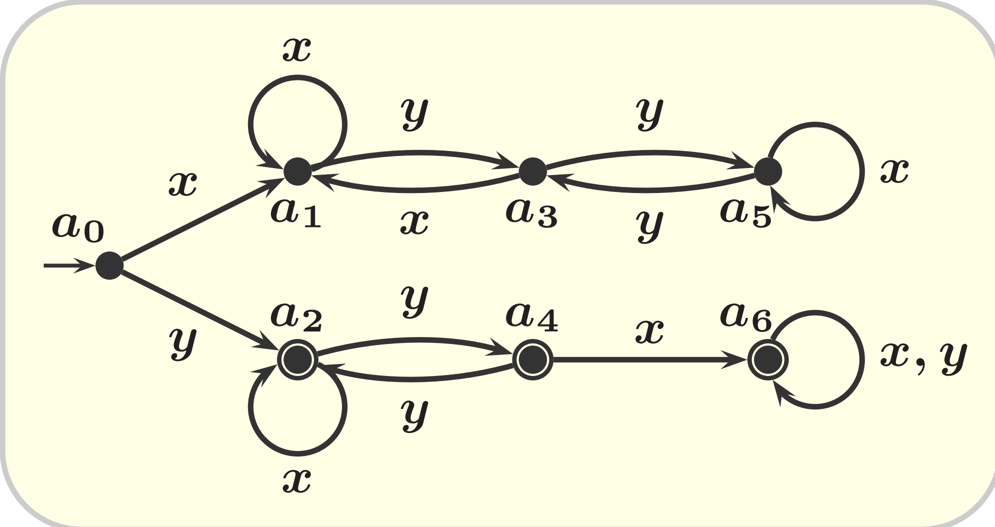


gde je $T = \{a_2, a_4, a_6\}$. Jasno da je $\mathcal{A}$ dostižan automat.

Jezik L koji automat $\mathcal{A}$ raspoznaje je $L = yX^*$.

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. kada se uđe u T, onda se iz njega ne može izaći.

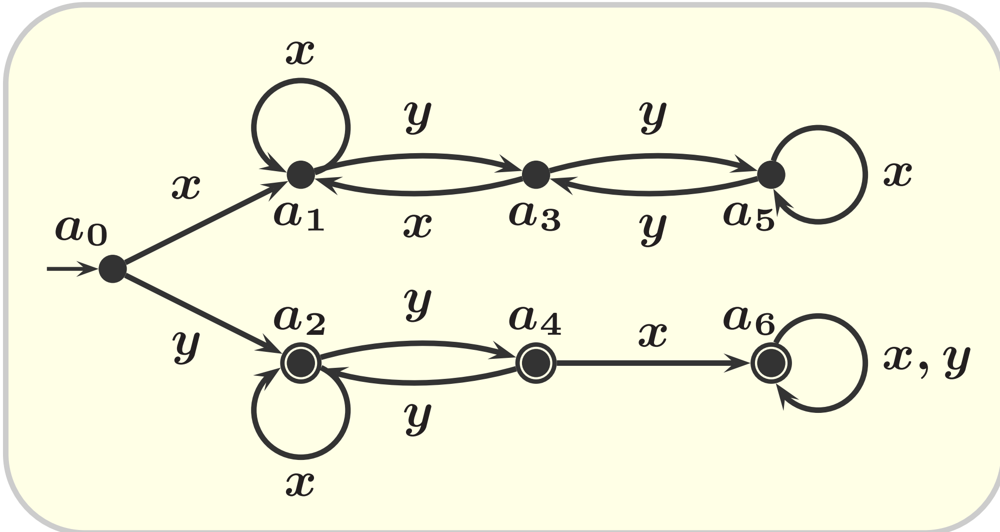
Dakle, $yv \in L$, što znači da je $yX^* \subseteq L$.

Time smo dokazali da je zaista $L = yX^*$.

Dalje treba uočiti da je

$$T.a_2 = T.a_4 = T.a_6 = X^*$$

$$T.a_1 = T.a_3 = T.a_5 = \emptyset.$$



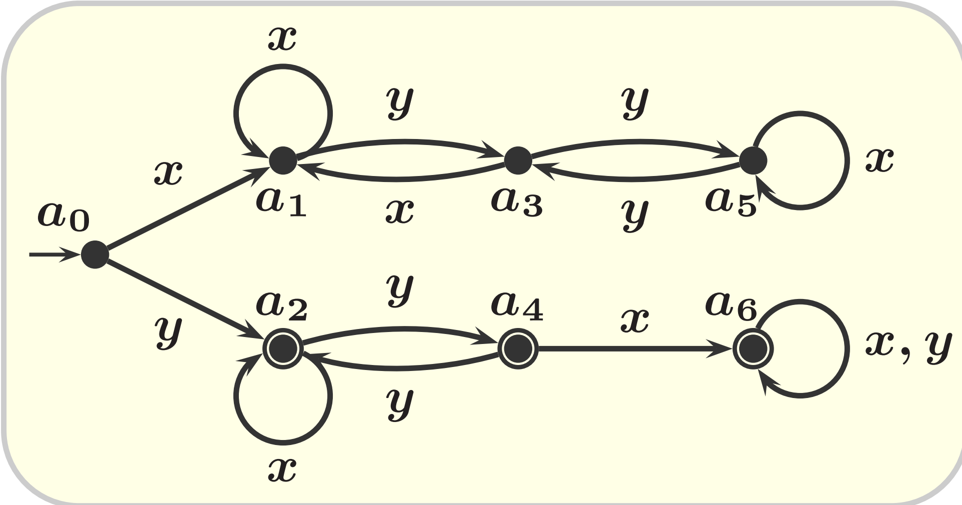
Dakle,

$$A_L = A_T = \{L, L_1, L_2\},$$

gde je

$$L_1 = \emptyset = T.a_1 = T.a_3 = T.a_5$$

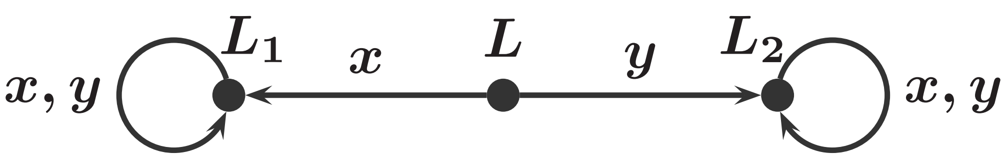
$$L_2 = X^* = T.a_2 = T.a_4 = T.a_6.$$



Takođe imamo da je

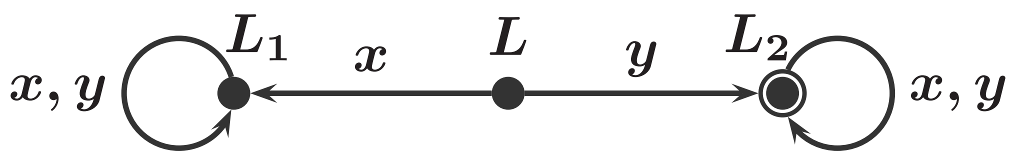
$$
\begin{aligned}
L.x &= (T.a_0).x = T.\delta(a_0, x) = T.a_1 = L_1, \\
L.y &= (T.a_0).y = T.\delta(a_0, y) = T.a_2 = L_2, \\
L_1.x &= (T.a_1).x = T.\delta(a_1, x) = T.a_1 = L_1, \\
L_1.y &= (T.a_1).y = T.\delta(a_1, y) = T.a_3 = L_1, \\
L_2.x &= (T.a_2).x = T.\delta(a_2, x) = T.a_2 = L_2, \\
L_2.y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

Prema tome, minimalni automat A_L jezika L je automat zadat grafom



Uočimo da je $L_2 = X^*$ jedini element iz A_L koji sadrži praznu reč e.

Dakle, automat A_L raspoznaje L skupom $T_L = \{L_2\}$, tj. stanjem L_2.



Kod drugog algoritma koji ćemo ovde prikazati se određuju stanja koja su međusobno ekvivalentna, i klasa međusobno ekvivalentnih stanja se zamenjuje samo jednim stanjem.

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

$$(a, b) \in \pi_T \;\Leftrightarrow\; T.a = T.b \;, \tag{1.9}$$

za $a, b \in A$.

Najpre dokazujemo sledeće:

Teorema 5.2.2 Neka je dat automat $A = (A, a_0, X, \delta, T)$.

Tada je π_T kongruencija na A.

Dokaz: Jasno da je π_T refleksivna, simetrična i tranzitivna relacija, tj. relacija ekvivalencije.

Neka su $a, b \in A$ stanja takva da je $(a, b) \in \pi_T$, tj. da je $T.a = T.b$, i neka je $x \in X$ proizvoljno slovo.

Tada za proizvoljnu reč $u \in X^*$ važi sledeće

$$
\begin{aligned}
u \in T.\delta(a, x) &\Leftrightarrow \delta(\delta(a, x), u) \in T \Leftrightarrow \delta(a, xu) \in T \\
&\Leftrightarrow xu \in T.a \Leftrightarrow xu \in T.b \\
&\Leftrightarrow \delta(b, xu) \in T \Leftrightarrow \delta(\delta(a, x), u) \in T \\
&\Leftrightarrow u \in T.\delta(b, x) \ .
\end{aligned}
$$

Prema tome, $(\delta(a, x), \delta(b, x)) \in \pi_T$, čime je dokazano da je π_T kongruencija na automatu A. $\square$

Teorema 5.2.3. Neka je $\mathcal{A} = (A, a_0, X, \delta, T)$ dostižan automat koji raspoznaje jezik $L \subseteq X^*$.

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

Dokaz: Neka je $\varphi : A \to A_L$ preslikavanje definisano sa

$$a\varphi = T.a \;, \qquad \text{za svaki } a \in A.$$

Prema Teoremi 5.2.1, φ slika $\mathcal{A}$ na $\mathcal{A}_L$.

Takođe, jasno je da je $\ker \varphi = \pi_T$, pa prema Teoremi o homomorfizmu, automat $\mathcal{A}/\pi_T$ je izomorfan sa $\mathcal{A}_L$, što je i trebalo dokazati. $\square$

Iz prethodne teoreme se vidi da se jedan od načina minimizacije automata A koji raspoznaje jezik L skupom stanja T svodi na konstrukciju kongruencije π_T.

To se može učiniti 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

$$(a, b) \in \varepsilon_T \quad \Leftrightarrow \quad (\, a \in T \;\Leftrightarrow\; b \in T\,). \tag{1.10}$$

Zanimljiva veza između relacija π_T i ε_T utvrđena je narednom lemom.

Lema 4. Za proizvoljan automat $A = (A, a_0, X, \delta, T)$ i proizvoljna stanja $a, b \in A$ važi

$$(a,b) \in \pi_T \quad \Leftrightarrow \quad (\forall u \in X^*)\ (\delta(a,u), \delta(b,u)) \in \varepsilon_T. \tag{1.11}$$

Dokaz: Na osnovu (1.10) imamo da je

$$\begin{aligned} (a,b) \in \pi_T \ &\Leftrightarrow\ T.a = T.b \\ &\Leftrightarrow\ (\forall u \in X^*)\ (u \in T.a \ \Leftrightarrow\ u \in T.b) \\ &\Leftrightarrow\ (\forall u \in X^*)\ (\delta(a,u) \in T \ \Leftrightarrow\ \delta(b,u) \in T) \\ &\Leftrightarrow\ (\forall u \in X^*)\ (\delta(a,u), \delta(b,u)) \in \varepsilon_T, \end{aligned}$$

pa dakle, važi (1.11). $\square$

Teorema 5.2.4. 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 $\mathcal{A}$ konačan automat, tada postoji $k \in \mathbb{N}^0$ tako da je $\pi_T^{(k)} = \pi_T$.

Dokaz: (a) Jasno da su svi članovi niza $\{\pi_T^{(k)}\}_{k\in\mathbb{N}^0}$ relacije ekvivalencije, a neposredno iz definicije tog niza sledi da je on opadajući lanac.

Dakle, preostaje da se dokaže da je $\pi_T \subseteq \pi_T^{(k)}$, za svaki $k \in \mathbb{N}^0$.

To će biti dokazano indukcijom po k.

Prvo, ako je $(a, b) \in \pi_T$, onda prema (1.11) imamo da je

$$(a, b) = (\delta(a, e), \delta(b, e)) \in \varepsilon_T,$$

što znači da je $\pi_T \subseteq \varepsilon_T = \pi_T^{(0)}$.

Dalje, pretpostavimo da je $\pi_T \subseteq \pi_T^{(k)}$ za neke $k \in \mathbb{N}^0$ i uzmimo da je $(a,b) \in \pi_T$.

Ako $(a,b) \notin \pi_T^{(k+1)}$, to znači da postoji $x \in X$ tako da

$$(\delta(a,x), \delta(b,x)) \notin \pi_T^{(k)} \subseteq \varepsilon_T.$$

Međutim, ovo prema (1.11) znači da $(a,b) \notin \pi_T$, što je u suprotnosti sa polaznom pretpostavkom da je $(a,b) \in \pi_T$.

Prema tome, zaključujemo da je $(a,b) \in \pi_T^{(k+1)}$, što znači da je $\pi_T \subseteq \pi_T^{(k+1)}$.

Time je dokaz tvrđenja (a) završen.

(b) I ovo tvrđenje će biti dokazano indukcijom po m.

Na osnovu polazne pretpostavke imamo da ono važi za $m = 1$.

Pretpostavimo da je $\pi_T^{(k)} = \pi_T^{(k+m)}$, za neki $m \in \mathbb{N}$. Tada je

$$
\begin{aligned}
\pi_T^{(k+m+1)} &= \left\{ (a,b) \in \pi_T^{(k+m)} \;\middle|\; (\forall x \in X)\,(\delta(a,x),\delta(b,x)) \in \pi_T^{(k+m)} \right\} \\
&= \left\{ (a,b) \in \pi_T^{(k)} \;\middle|\; (\forall x \in X)\,(\delta(a,x),\delta(b,x)) \in \pi_T^{(k)} \right\} \\
&= \pi_T^{(k+1)} = \pi_T^{(k)},
\end{aligned}
$$

čime je dokaz završen.

Prema (a), da bi smo dokazali da je $\pi_T^{(k)} = \pi_T$, dovoljno je dokazati da je $\pi_T^{(k)} \subseteq \pi_T$.

U tom cilju, uzmimo proizvoljan par $(a, b) \in \pi_T^{(k)}$ i indukcijom po dužini reči u dokažimo da je $(\delta(a, u), \delta(b, u)) \in \varepsilon_T$, za svaku reč $u \in X^*$.

Ako je $|u| = 0$, tj. u je prazna reč, tada je

$$(\delta(a, u), \delta(b, u)) = (a, b) \in \pi_T^{(k)} \subseteq \varepsilon_T,$$

što smo i hteli dobiti.

Za bilo koji $n \in \mathbb{N}^0$, uzmimo da je $(\delta(a, u), \delta(b, u)) \in \varepsilon_T$, za svaku reč u dužine $|u| \leqslant n$, i uzmimo proizvoljnu reč $v \in X^*$ dužine $|v| = n+1$.

Tada je $v = ux$, za neke $u \in X^*$, $|u| = n$, i $x \in X$.

Prema indukcijskoj pretpostavci imamo da je

$$(\delta(a,u),\delta(b,u)) \in \pi_T^{(k)} = \pi_T^{(k+1)}.$$

Međutim, odatle neposredno sledi da je

$$\begin{aligned}(\delta(a,v),\delta(b,v)) &= (\delta(a,ux),\delta(b,ux)) \\ &= (\delta(\delta(a,u),x),\delta(\delta(b,u),x)) \in \pi_T^{(k)} \subseteq \varepsilon_T,\end{aligned}$$

što je i trebalo dokazati.

Prema tome, zaključujemo da je $(\delta(a,u),\delta(b,u)) \in \varepsilon_T$, za svaku reč $u \in X^*$, što prema (1.11) znači da je $(a,b) \in \pi_T$.

Time smo dokazali da je $\pi_T^{(k)} = \pi_T$, čime je dokaz tvrđenja (b) završen.

(c) Ako je automat A konačan, tada postoji konačno mnogo relacija na A.

To znači da bar dve relacije u opadajućem lancu iz (a) moraju biti jednake.

Prema tome, postoje $k \in \mathbb{N}^0$ i $m \in \mathbb{N}$ tako da je $\pi_T^{(k)} = \pi_T^{(k+m)}$.

Sada je jasno da je $\pi_T^{(k)} = \pi_T^{(k+1)}$, pa prema (b) sledi da je $\pi_T^{(k)} = \pi_T$.

Ovim je dokaz teoreme završen. $\square$

Sada možemo dati još jedan algoritam za konstrukciju minimalnog automata jezika.

Algoritam: Minimizacija automata

Ulaz: $\mathcal{A} = (A, a_0, X, \delta, T)$ – automat koji raspoznaje jezik L

Izlaz: $\mathcal{A}_L$ – minimalni automat jezika L

1. Formiramo listu P svih parova stanja automata $\mathcal{A}$.

   Listu P možemo grafički predstaviti tablicom, pri čemu je, zbog refleksivnosti i simetričnosti relacija koje konstruišemo, dovoljno razmatrati samo parove koji leže ispod glavne dijagonale.

2. Određujemo dostižni deo A^d automata $\mathcal{A}$, brisanjem sa liste P svih kolona i vrsta koje odgovaraju nedostižnim stanjima.

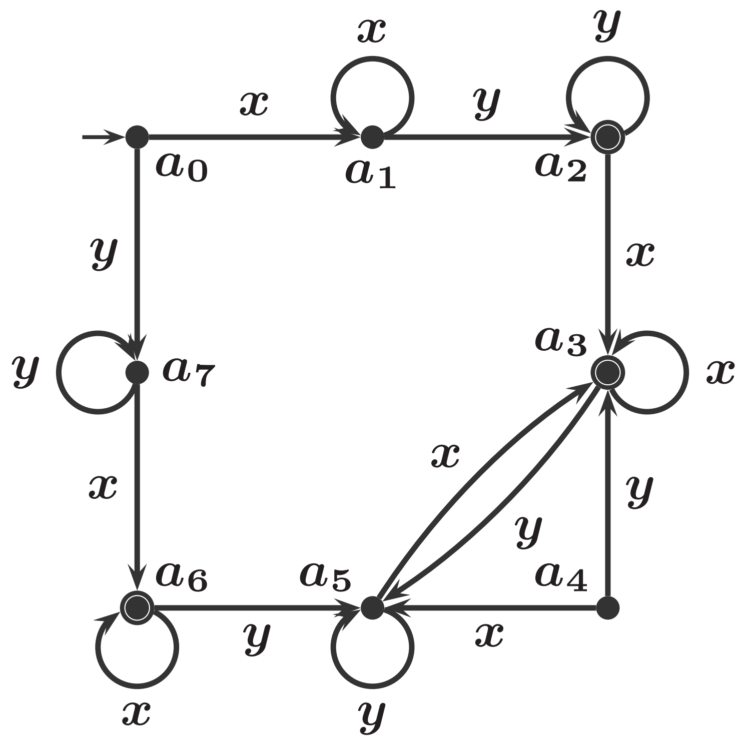
3. Konstruišemo relaciju ε_T, brisanjem sa liste P svih parova iz skupa $T \times (A^d \setminus T) \cup (A^d \setminus T) \times T$.

   Parovi koji ostaju čine relaciju $\varepsilon_T = \pi_T^{(0)}$.

4. Ako posle $k+1$-vog koraka imamo relaciju $\pi_T^{(k)}$, gde je $k \geqslant 0$, onda se u $k+2$-gom koraku konstruiše relacija $\pi_T^{(k+1)}$.
   4.1 Razmatramo parove (a, b) koji su na početku ovog koraka bili na listi P.
   4.2 Ukoliko proverom ustanovimo da postoji slovo $x \in X$ takvo da na početku ovog koraka par $(\delta(a, x), \delta(b, x))$ nije bio na listi P, onda se sa liste brišu parovi (a, b) i (b, a).
5. Ovaj postupak se završava prvim korakom u kome nije bilo nijednog brisanja sa liste.

   Parovi koji su ostali na listi čine relaciju π_T.
6. Od klasa relacije π_T formiramo automat A_L.

Primer 5.2.2. Neka je automat $\mathcal{A} = (A, a_0, X, \delta, T)$ dat grafom



Kao što smo rekli, formiramo listu P svih parova stanja automata $\mathcal{A}$, koju ovde predstavljamo tablicom parova.

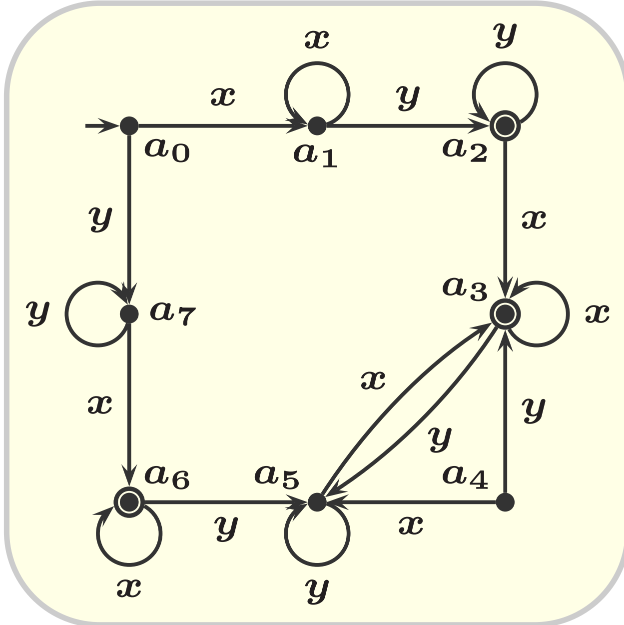
Kako su relacije koje generišemo refleksivne i simetrične, to posmatramo samo deo tablice ispod glavne dijagonale.

Minimizacija automata – Algoritam II

1. korak: Izbacujemo iz liste P sve parove iz vrsti i kolona koje odgovaraju nedostižnim stanjima, u ovom slučaju stanja a_4.

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

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



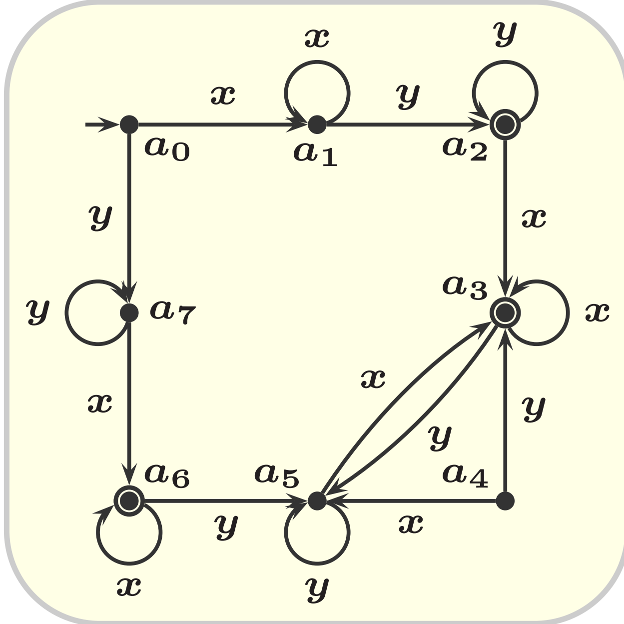
2. korak: Izbacujemo iz liste P sve parove iz skupa $T \times (A^d \setminus T) \cup (A^d \setminus T) \times T$.

Setimo se 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	
a_1			X	X	X		X	
a_2	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	
a_6	X	X			X	X		X
a_7			X	X	X		X	

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;

$(\delta(a_1, y), \delta(a_0, y)) = (a_2, a_7)$ – 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,

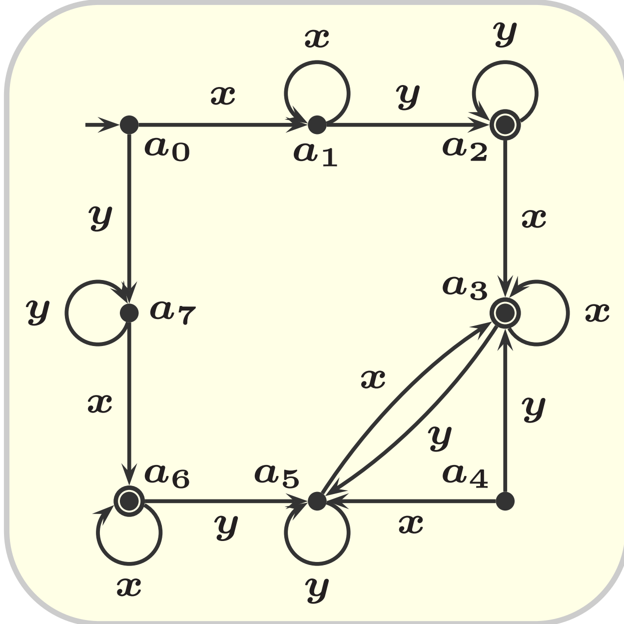
brišemo parove (a_3, a_2) i (a_2, a_3);

$(\delta(a_5, x), \delta(a_0, x)) = (a_3, a_1)$ – 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);



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

lista P posle 2. koraka

$(\delta(a_6,x),\delta(a_2,x)) = (a_6,a_3)$ – 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,

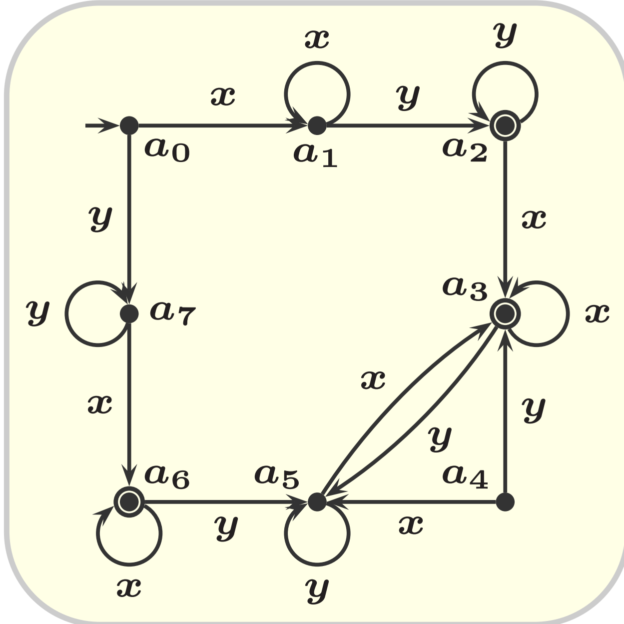
brišemo parove (a_7,a_0) i (a_0,a_7);

$(\delta(a_7,x),\delta(a_1,x)) = (a_6,a_1)$ – 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.



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

lista P posle 2. koraka

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 sledeći izgled:

	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	X
a_2	X	X		X	X	X	X	X
a_3	X	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	

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

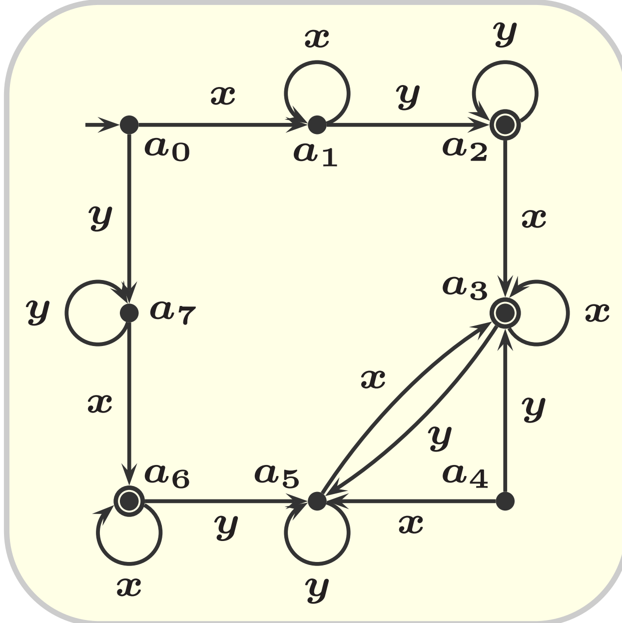
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_0			X	X	X		X	
a_1			X	X	X		X	
a_2	X	X			X	X		X
a_3	X	X			X	X		X
a_4	X	X	X	X	X	X	X	X
a_5			X	X	X		X	
a_6	X	X			X	X		X
a_7			X	X	X		X	

lista P posle 2. koraka

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.



	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	╳	╳	╳	╳	╳		╳	

posle 4. koraka – relacija $\pi_T^{(2)} = \pi_T^{(1)}$

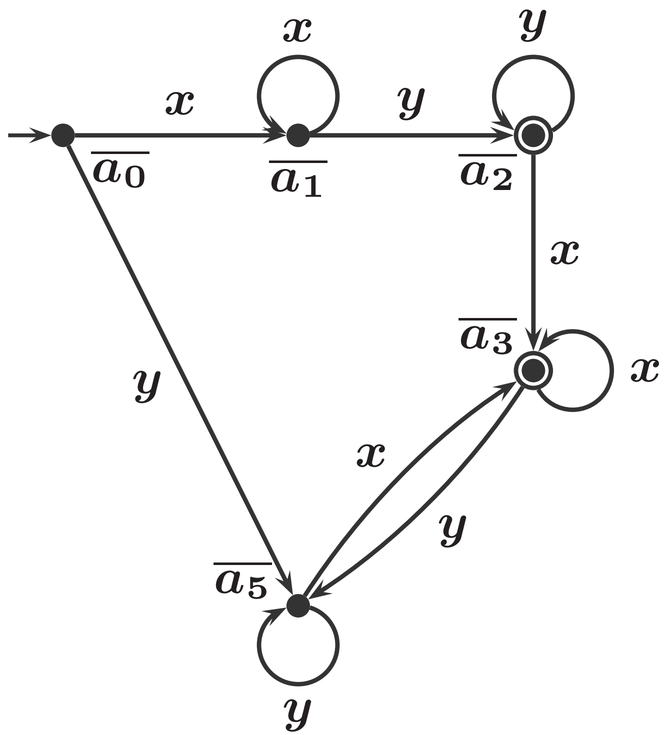
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_0		╳	╳	╳	╳	╳	╳	╳
a_1	╳		╳	╳	╳	╳	╳	╳
a_2	╳	╳		╳	╳	╳	╳	╳
a_3	╳	╳	╳		╳	╳		╳
a_4	╳	╳	╳	╳	╳	╳	╳	╳
a_5	╳	╳	╳	╳	╳		╳	
a_6	╳	╳	╳		╳	╳		╳
a_7	╳	╳	╳	╳	╳		╳	

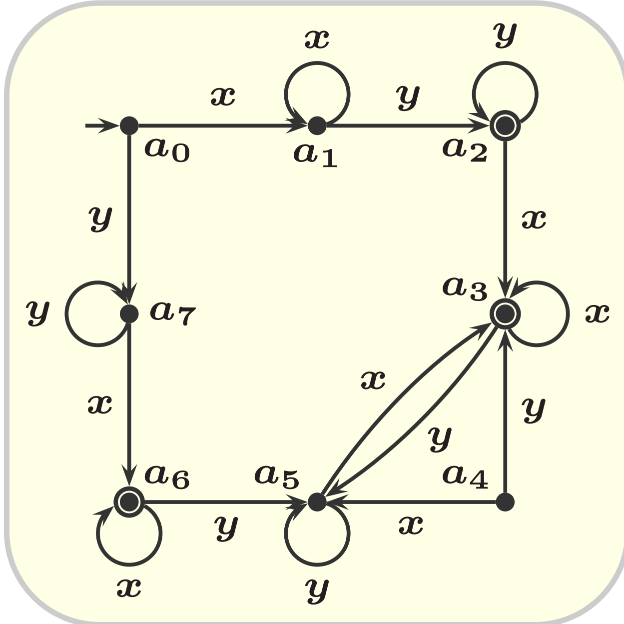
lista P posle 3. koraka

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

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

Traženi minimalni automat predstavljen je grafom





	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	X
a_2	X	X		X	X	X	X	X
a_3	X	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	

lista P na kraju algoritma