UNIVERZITET METROPOLITAN BEOGRAD

CS215 - DISKRETNE STRUKTURE

RELACIJE UREĐENJA I MAŠINE KONAČNIH STANJA

Lekcija 13

PRIRUČNIK ZA STUDENTE

CS215 - DISKRETNE STRUKTURE

Lekcija 13

RELACIJE UREĐENJA I MAŠINE KONAČNIH STANJA

Copyright © 2017 – UNIVERZITET METROPOLITAN, Beograd. Sva prava zadržana. Bez prethodne pismene dozvole od strane Univerziteta METROPOLITAN zabranjena je reprodukcija, transfer, distribucija ili memorisanje nekog dela ili čitavih sadržaja ovog dokumenta., kopiranjem, snimanjem, elektronskim putem, skeniranjem ili na bilo koji drugi način.

Copyright © 2017 BELGRADE METROPOLITAN UNIVERSITY. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without the prior written permission of Belgrade Metropolitan University.

Uvod

UVOD

Fokus ove lekcije je na relaciji uređenja i njenim primerima

U ovom predavanju ćemo se prisetiti osnovnih osobina parcijalno uređenih skupova, nastaviti sa njihovim izučavanjem i uvesti Hasove dijagrame (engl. Hasse diagrams) kao korisnu grafičku reprezentaciju konačnih parcijalno uređenih skupova. U ovoj lekciji biće obrađeni sledeći pojmovi:

- Parcijalno uređeni skupovi
- Linearno uređen skup
- Parcijalno uređenje proizvoda
- Leksikografsko uređenje
- Hasovi dijagrami
- Topološko sortiranje
- Ekstremni elementi
- Mašine konačnih stanja

Ove strukture i koncepti su fundamentalni u mnogim oblastima matematike i imaju važne primene u računarskoj nauci.

Poglavlje 1

PARCIJALNO UREĐENjE

DEFINICIJA PARCIJALNO UREĐENE RELACIJE

Relacija R nad nepraznim skupom S je parcijalno uređena ili predstavlja parcijalno uređenje ako je refleksivna, antisimetrična i tranzitivna

Definicija
Relacija R nad nepraznim skupom S
je parcijalno uređena ili predstavlja
parcijalno uređenje ako je refleksivna,
antisimetrična i tranzitivna.
Skup S zajedno sa parcijalnim uređenjem
na njemu, R, se zove parcijalno uređen skup.

Primer

Relacija podskupova skupova $\subset$ je parcijalno uređena nad bilo kojom kolekcijom skupova, zato što važi

1. $A \subset A$ za bilo koji skup A;
2. ako $A \subset B$ i $B \subset A$ onda $A = B$;
3. ako $A \subset B$ i $B \subset C$ onda $A \subset C$.

PARCIJALNO UREĐENI SKUPOVI

Skup A zajedno sa parcijalnim uređenjem R se zove parcijalno uređen skup

Za relaciju R nad skupom A kažemo da je parcijalno uređena kada zadovoljava

- $(a, a) \in R$ za sve $a \in R$ refleksivnost
- $(a, b); (b, a) \in R$ sledi $a = b$ antisimetrija
- $(a, b), (b, c) \in R$ sledi $(a, c) \in R$ tranzitivnost.

Skup A zajedno sa parcijalnim uređenjem R se zove parcijalno uređen skup (engl. partially ordered set), ili skraćenoposet (engl. poset) i označavaćemo ga sa (A,R)

Primeri

Neka je S proizvoljan skup i A $\subset$ P(S), partitivni skup skupa S.
Relacija biti podskup je parcijalno uređenje na skupu A, a (A, $\subset$) je poset .

Neka $\mathbb{N}$ označava skup prirodnih brojeva. Uobičajena relacija $\leq$ je parcijalno uređenje na $\mathbb{N}$,($\mathbb{N}$, $\leq$). Isto važi i za relaciju $\geq$.
x $\leq$ x za sve x $\in$ $\mathbb{N}$
x $\leq$ y i y $\leq$ x sledi x = y
x $\leq$ y i y $\leq$ z sledi x $\leq$ z

Relacija deljivosti (aRb ako i samo ako a|b) je parcijalno uređenje na $\mathbb{N}$. Neka je R skup svih relacija ekvivalencije na skupu A. Kako se R sastoji od podskupova skupa A $\times$ A, R je parcijalno uređen skup u odnosu na parcijalno uređenje ‘sadržati se u’.Ako su R i S relacije ekvivalencije na skupu A, ista osobina se može izraziti relacijskim oznakama na sledeći način R $\subset$ S ako i samo ako xRy sledi xSy za sve x, y $\in$ A:

Tada je (R, $\subset$) poset.

Relacija <, ‘strogo manje’ na skupu N nije parcijalno uređenje , pošto nije refleksivna. Neka je R parcijalno uređenje na skupu A, i neka je R^{-1} inverzna relacija relacije R. Tada je R^{-1} takođe parcijalno uređenje

DUALNI POSET

Poset (A, R^{-1}) se zove dualni poset poseta (A,R), a parcijalno uređenje R^{-1} se zove dualno parcijalno uređenje parcijalnog uređenja R

Poset (A, R^{-1}) se zove dualni poset poseta (A,R) (engl. dual of the poset (A,R)) a parcijalno uređenje R^{-1} se zove dualno parcijalno uređenje parcijalnog uređenja R (engl. dual of the partial order R).

Najpoznatija parcijalna uređenja su relacije $\leq$ i $\geq$ na skupovima $\mathbb{Z}$ i $\mathbb{R}$. Zbog toga, kada govorimo o proizvoljnom parcijalnom uređenju R na skupu A, često ćemo koristiti simbole $\leq$ ili $\geq$ za R.

Takođe ćemo naglasiti sledeću konvenciju.

Kadgod je (A, $\leq$) poset, koristićemo simbol $\geq$ za dualno parcijalno uređenje $\leq$ -1, pa će tako (A, $\geq$) biti dualni poset poseta (A, $\leq$).

Poglavlje 2

LINEARNO UREĐENjE

LINEARNO UREĐEN SKUP I LANAC

Ako je (A, $\leq$)poset, elementi a, b $\in$ A su uporedivi ako a $\leq$ b ili b $\leq$ a

Ako je (A, $\leq$)poset, elementi a, b $\in$ A su uporedivi (engl. comparable) ako

a $\leq$ b ili b $\leq$ a.

Ako je svaki par elemenata u posetu A uporediv,
kazemo da je A linearno uređen skup, a parcijalno uređenje se zove linearno uređenje

Linearno uređen podskup nekog poseta se zove lanac.

UPOREDIVI I NEUPOREDIVI ELEMENTI

Relacija $\leq$ je parcijalno uređenje na N

Primetimo da u parcijalno uređenom skupu mogu postojati elementi koji nisu uporedivi.

Na primer, ako posmatramo relaciju deljivosti nad skupom $\mathbb{N}$ **:**

- Relacija deljivosti (aRb ako i samo ako a|b) je parcijalno uređenje na $\mathbb{N}$
- *Onda elementi 2 i 7 nisu uporedivi, pošto 2 ne deli 7, i 7 ne deli 2.*
- Tako reč parcijalan u terminu parcijalno uređen skup znači da neki elementi iz skupa mogu biti neuporedivi.

Poset iz već navedenog primera, relacije $\leq$ manje ili je jednako, je linearno uređen

2.1 LEKSIKOGRAFSKO UREĐENJE

PARCIJALNO UREĐENJE PROIZVODA I LEKSIKOGRAFSKO UREĐENJE

Parcijalno uređenje definisano na Dekartovom proizvodu skupova kao u prethodnoj teoremi se zove parcijalno uređenje proizvoda

Parcijalno uređenje definisano na Dekartovom proizvodu
skupova kao u prethodnoj teoremi se zove
parcijalno uređenje proizvoda.
Parcijalno uređenje na A x B, u oznaci ≺, se definiše
na sledeći način:
(a, b) ≼(a′, b′) ako a < a′ ili ako a = a′ i b ≤ b′
Ovo uređenje se zove leksikografsko (lexicographic).

Uređenje se odlučuje po prvoj koordinati, a ako su prve koordinate jednake uređenje se odlučuje po drugoj koordinati.
Ako su (A, $\leq_1$) i (B, $\leq_2$) linearno uređeni skupovi, onda je i leksikografsko uređenje na A × B takođe linearno uredjenje.

Leksikografsko uređenje se lako može proširiti na Dekartov proizvod A_1 \times A_2 \times . . . \times A_n na sledeći način:

(a_1, a_2, . . . , a_n) ≼(a_1', a_2', . . . , a_n') ako i samo ako

$a_1 < a_1'$ ili

$a_1 = a_1'$ i $a_2 < a_2'$ ili

$a_1 = a_1'$ $a_2 = a_2'$, i $a_3 < a_3'$ ili . . .

$a_1 = a_1'$ $a_2 = a_2'$, . . . , $a_{n-1} = a_{n-1}'$

i

$a_n \leq a_n'$

Tako se uređenje odlučuje po prvoj koordinati. Ako su prve koordinate jednake, uređenje se odlučuje po drugoj koordinati. Ako su i one jednake, odluka se prenosi na sledeću koordinatu, i tako dalje.

PRIMER LEKSIKOGRAFSKOG UREĐENJA

Leksikografsko uređenje na A × B, u oznaci ≺, se definiše na sledeći način: (a, b) ≼(a′, b′) ako a <a′ ili ako a = a′ i b ≤ b′

Primer

Neka je S = {a, b, c, . . . , z } engleski alfabet, linearno uređen na uobičajen način a ≤ b, b ≤ c, . . . , y ≤ z

Tada $S_n = S \times S \times \ldots \times S$(n faktora) može predstavljati skup svih reči dužine n.

Leksikografsko uređenje na Sn ima osobinu da, ako $w_1 \preccurlyeq w_2$ ($w_1, w_2 \in S_n$), onda w_1 predhodi w_2 u rečniku. Po tome je leksikografsko uređenje dobilo ime.

Tako,
park $\preccurlyeq$ part, pošto p = p, a = a, r = r, k < t,
jump $\preccurlyeq$ lump, pošto j < l,
help $\preccurlyeq$ hint, pošto h = h, e < i.

Poglavlje 3

GRAFOVI PARCIJALNO UREĐENIH RELACIJA

GRAFOVI PARCIJALNO UREĐENE RELACIJE

Digraf parcijalnog uredjenja nema cikluse dužine veće od 1

Kako je parcijalno uređenje relacija, možemo digrafom (usmeren graf) predstaviti bilo koje parcijalno uređenje na konačnom skupu. Videćemo da se digrafovi parcijalnih uređenja mogu predstaviti na sličan način kao oni kod opštih relacija. Sledeća teorema daje prvi rezultat u tom smislu.

Teorema
Digraf parcijalnog uredjenja nema cikluse dužine veće od 1

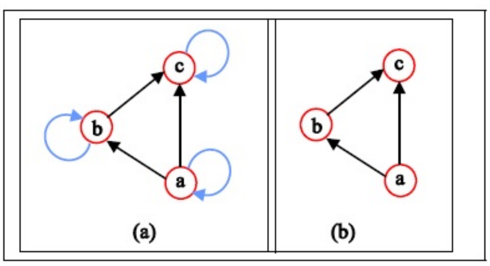
3.1 HASOV DIJAGRAM

PREDSTAVLJANJE PARCIJALNOG UREĐENJA HASOVIM DIJAGRAMOM

Hasov dijagram potpuno opisuje odgovarajuće parcijalno uređenje

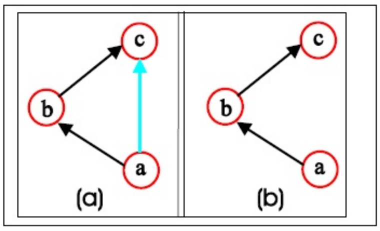
Neka je A konačan skup. Tada digraf parcijalnog uređenja ≤ na skupu A ima samo cikluse dužine 1 po teoremi. *Pošto je parcijalno uređenje refleksivno, svaki čvor digrafa parcijalnog uređenja se nalazi u nekom ciklusu dužine 1.*

Da bi uprostili razmatranje, obrisaćemo sve takve cikluse iz digrafa. Tako bi digraf prikazan na slici (a) bio nacrtan kao sto je prikazano na slici (b).

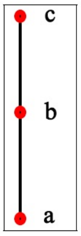


Slika 3.1.1 (a) Kompletan digraf relacije, (b) Digraf iste relacije kod koga su uklonjeni ciklusi dužine 1 [Izvor: Autor]

Takođe ćemo eliminisati sve grane koje slede iz osobine tranzitivnosti. Tako, ako a ≤ b i b ≤ c, sledi da a ≤ c. U tom slučaju ćemo izostaviti granu od a do c, a zadržaćemo grane od a do b i od b do c.



Slika 3.1.2 Eliminacija grana iz digrafa koje slede iz osobine tranzitivnosti [Izvor: Autor]



Slika 3.1.3 Konačan Hasov dijagram [Izvor: Autor]

Takođe ćemo uvesti konvenciju da digraf parcijalnog uređenja crtamo tako da su sve *grane usmerene nagore, tako da se i strelice na granama mogu izostaviti.* Konačno, *krugove koji predstavljaju čvorove zamenjujemo tačkama.* Tako dobijamo konačnu formu digrafa parcijalnog uređenja, koji je mnogo jednostavniji od svog digrafa. Ovaj digraf se zove Hasov dijagram (engl. Hasse diagram) parcijalnog uređenja poseta.

Pošto Hasov dijagram potpuno opisuje odgovarajuće parcijalno uređenje, on predstavlja vrlo korisnu alatku.

PRIMER 1

Nacrtati Hasov dijagram parcijalnog uređenja relacije deljivosti na skupu A

Primer

Neka je A = {1, 2, 3, 4, 12}.

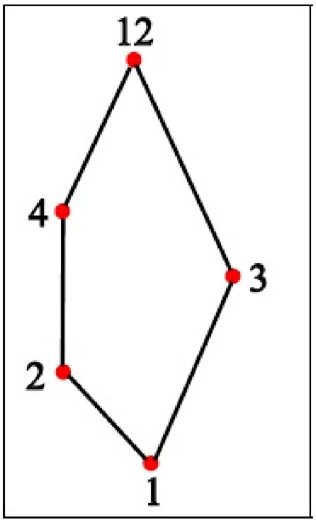
Posmatramo parcijalno uređenje relacije deljivosti na skupu A. Drugim rečima, ako a, b ∈ A, onda a ≤ b ako i samo ako a|b.

Nacrtati Hasov dijagram poseta (A, ≤).

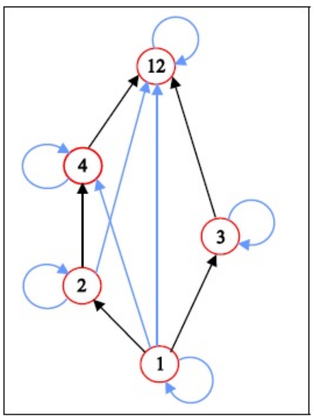
Rešenje:

Hasov dijagram je prikazan na slici 4.

Da bi naglasili jednostavnost Hasovog dijagrama, na slici 5 prikazujemo digraf istog poseta.



Slika 3.1.4 Hasov dijagram poseta iz primera [Izvor: Autor]



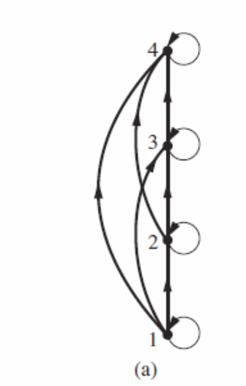
Slika 3.1.5 Digraf relacije iz primera [Izvor: Autor]

PRIMER 2

Razmotrimo usmereni graf za parcijalno uređenje {(a, b) | a ≤ b} nad skupom {1, 2, 3, 4}

Primer

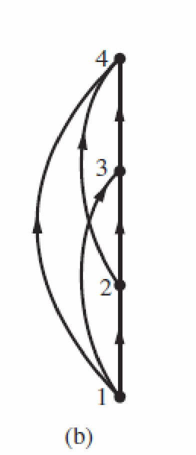
Razmotrimo usmereni graf za parcijalno uređenje {(a, b) | a ≤ b} nad skupom {1, 2, 3, 4} prikazanim na slici



Slika 3.1.6 Usmereni graf za ({1, 2, 3, 4},=) [Izvor: Autor]

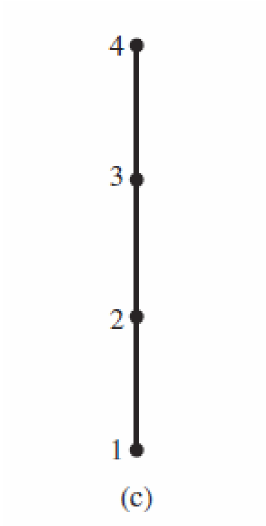
Pošto se radi o parcijalnom uredenju, ova relacija je refleksivna, pa svaki cvor na prikazanoj slici ima petlju.

Prema pravilima Hasovog dijagrama ove petlje ne moramo da prikažemo, pošto znamo da one moraju postojati.Tako, u sledecem koraku uklanjamo petlje kao što je prikazano na slici



Slika 3.1.7 Drugi korak u izradi Hasovog dijagrama [Izvor: Autor]

Pošto je parcijalno uređenje takođe i tranzitivno, uklanjamo i sve grane koje postoje kao posledica tranzitivnosti. Na primer,
grane (1, 3), (1, 4), i (2, 4) nisu prikazane pošto one postoje zbog tranzitivnosti. Ako pretpostavimo da nije potrebno da pokažemo ni usmerenje grana, tada dobijamo Hasov dijagram kao što je prikazano na sledećoj slici



Slika 3.1.8 Hasov dijagram za ({1, 2, 3, 4},≤). [Izvor: Autor]

PRIMER - VIDEO

Razmotrimo usmereni graf za parcijalno uređenje {(a, b) | a ≤ b} nad skupom {1, 2, 3, 4}.

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

3.2 TOPOLOŠKO SORTIRANJE

DEFINICIJA TOPOLOŠKOG SORTIRANJA

Proces konstruisanja linearnog uređenja se zove topološko sortiranje

Ako je A poset sa parcijalnim uređenjem , nekad je potrebno naći linearno uređenje < zataj skup koje će prosto biti proširenje datog uređenja u smislu da **ako je a ≤ b onda mora da bude a < b.** Proces konstruisanja takvog linearnog uređenja se zove topološko sortiranje (engl. topological sorting).

Primer

Ovaj problem može nastati kada treba učitati konačan poset A u kompjuter. Elementi skupa A se moraju učitati u nekom redosledu, a mi želimo da ih učitamo tako da se očuva parcijalno uređenje. To jest, ako je a ≤ b, onda a treba učitati pre b.

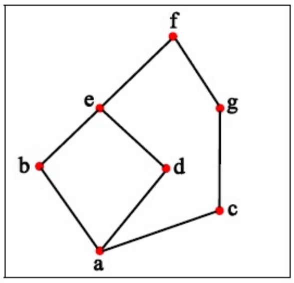
Topološko sortiranje < će dati redosled učitavanja elemenata koji zadovoljava ovaj uslov.

Primer

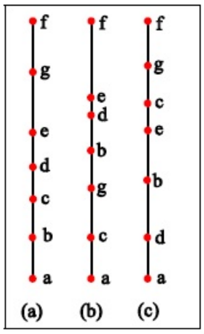
Odrediti topološko sortiranje poseta čiji je Hasov dijagram prikazan na slici 1.

Rešenje:

Jasno je da je parcijalno uređenje čiji je dijagram prikazan na slici 2 (a) linearno uređenje. Lako je pokazati da svaki par koji je u relaciji ≤ je takođe u relaciji ≺ pa je ≺ topološko sortiranje parcijalnog uređenja ≤.
Slike 2 (b) i 2 (c) prikazuju druga dva rešenja ovog problema.



Slika 3.2.1 Hasov dijagram poseta iz primera [Izvor: Autor]



Slika 3.2.2 Dijagrami topološkog sortiranja parcijalnog uređenja sa slike 1 [Izvor: Autor]

PRIMER

Topološko sortiranje poseta ({1, 2, 4, 5, 12, 20}, |).

Primer

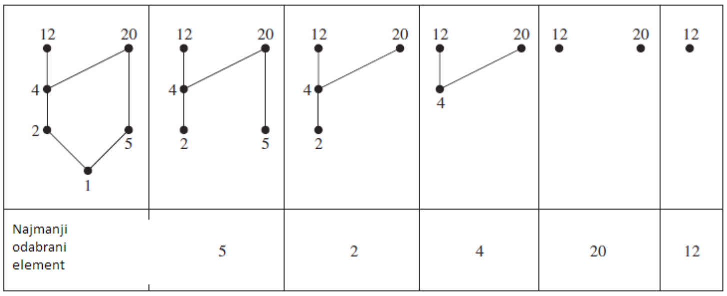
Obaviti topološko sortiranje za poset ({1, 2, 4, 5, 12, 20}, |).

Rešenje

Prvi korak je da se izabere minimalni element. U ovom primeru to mora biti 1, pošto je to jedini minimalni element. U sledećem koraku se bira minimalni element od ({2, 4, 5, 12, 20}, |). Pošto postoje dva minimalna elementa, a to su 2 i 5, biramo jedan. U ovom slučaju biramo 5. Ostaju nam elementi {2, 4, 12, 20}. U ovom koraku jedini minimalni element je 2. Sledeći je 4 minimalni element od ({4, 12, 20}, |). Pošto preostaju 12 i 20 kao minimalni elementi, može se izabrati bilo koji od ova dva , ({12, 20}, |). Biramo 20, što ostavlja 12 kao poslednji element. Ovaj sled koraka proizvodi redosled

$1 \prec 5 \prec 2 \prec 4 \prec 20 \prec 12.$

Koraci ovog sortiranja su prikazani na slici



Slika 3.2.3 Topološko sortiranje poseta ({1, 2, 4, 5, 12, 20}, |). [Izvor: Rosen]

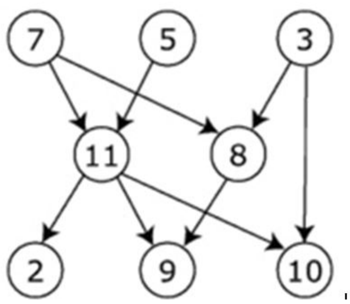
PRIMER TOPOLOŠKOG SORTIRANJA

Prikaz različitih primera topoloških sortiranja

Primer

Graf prikazan na slici ima više različitih mogućih načina i rešenja za topološko sortiranje

- 7,5,3,11,8,2,9,10 (vizuelno s leva u desno, odozgo na dole)
- 7,5,11,2,3,8,9, 10 (odozgo na dole, s leva u desno)
- 3, 5, 7, 8, 11, 2, 9, 10 (prvi najmanji dostupni čvor)



Slika 3.2.4 Prikaz topološkog sortiranja iz primera [Izvor: Autor]

3.3 EKSTREMNI ELEMENTI

MAKSIMALNI I MINIMALNI ELEMENTI POSETA

Element $a \in A$ se zove maksimalni element poseta A, ako ne postoji element $c \in A$ takav da je $a \prec c$.

Neki elementi poseta su od specijalnog značaja za mnoge osobine i primene poseta. Ovde ćemo razmatrati takve elemente.

Pretpostavimo da je (A, ≤) poset.

Element a ∈ A se zove maksimalni element poseta A,ako ne postoji element c ∈ A takav da je a < c.

Element b ∈ A se zove minimalni element poseta A, ako ne postoji element c ∈ A takav da c < b.

Direktno sledi da ako je (A, ≤)poset i (A, ≥)njegov dualni poset, element a ∈ A je maksimalni element poseta (A, ≥), ako i samo ako je a minimalni element poseta (A, ≤). Takođe, a je minimalni element poseta (A, ≥), ako i samo ako je a maksimalni element poseta (A, ≤)

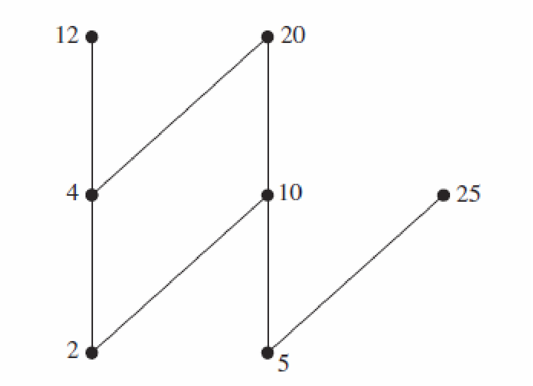
Teorema

Neka je A proizvoljan konačan neprazan skup sa parcijalnim uređenjem. Tada A ima bar jedan maksimalni i bar jedan minimalni element.

PRIMER - MAKSIMALNI I MINIMALNI ELEMENTI

Odrediti minimalne i maksimalne elemente poseta ({2, 4, 5, 10, 12, 20, 25}, |)

Primer
Koji su elementi maksimalni, a koji minimalni poseta ({2, 4, 5, 10, 12, 20, 25}, |) ?



Slika 3.3.1 Hasov dijagram poseta [Izvor: Autor]

Rešenje
Na slici je prikazano da su maksimalni elementi 12, 20 i 25, a minimalni 2 i 5. Kao što i sam primer pokazuje poset može imati više maksimalnih i minimalnih elemenata.

NAJVEĆI I NAJMANJI ELEMENT POSETA

Element a ∈ A se zove najveći element poseta A ako x ≤ a za svako x ∈ A.Element b ∈ A se zove najmanji element poseta A ako b≤ x za svako x ∈ A.

Definicija
Element a $\in$ A se zove najveći element poseta A
ako x $\leq$ a za svako x $\in$ A.
Element b $\in$ A se zove najmanji element poseta A
ako b $\leq$ x za svako x $\in$ A.

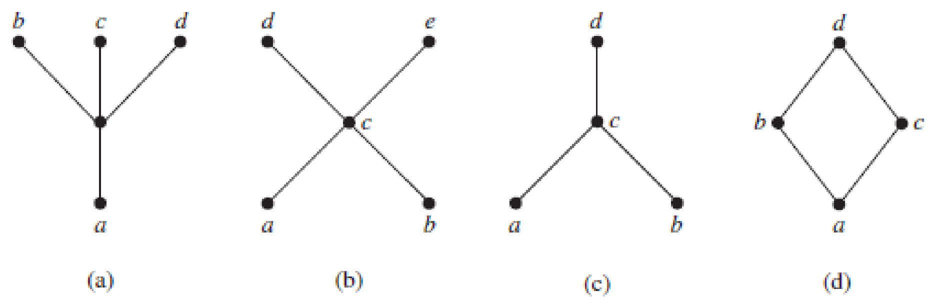
Primer
Neka je A poset nenegativnih realnih brojeva sa uobičajenim parcijalnim uređenjem ≤. Tada je 0 najmanji element, a ovaj poset nema najveći element.
Neka je S = {a, b, c} i posmatramo poset A = P(S) sa relacijom biti podskup. Tada imamo
A = { $\emptyset$, {a}, {b}, {c}, {a, b}, {a, c}, {b, c}, S }
$\emptyset$ je najmanji element, a S je najveći element poseta A.
Poset Z sa uobičajenim parcijalnim uređenjem nema ni najmanji ni najveći element

Poset ima najviše jedan najveći i najviše jedan najmanji element .

PRIMER - NAJVEĆI I NAJMANJI ELEMENT

Odrediti da li poseti prikazani Hasovim dijagramima na slici imaju najveći i najmanji element

Primer
Odrediti da li poseti prikazani Hasovim dijagramima na slici imaju najveći i najmanji element. Odredite ih ako postoje.



Slika 3.3.2 Hasovi dijagrami za četiri poseta [Izvor: Autor]

Rešenje

- Najmanji element Hasovog dijagrama (a) je a. Ovaj poset nema najveći element.
- Poset sa Hasovim dijagramom (b) nema ni najmanji ni najveći element.
- Poset sa Hasovim dijagramom (c) nema najmanji element, a njegov najveći element je d.
- Najmanji element poseta sa Hasovim dijagramom (d) je a, a najveći d.

Poglavlje 4

MAŠINE KONAČNIH STANJA

PRIMENA MAŠINA KONAČNIH STANJA

Mašine konačnih stanja se dosta koriste u računarskim naukama i mrežama

Mnogi tipovi mašina, uključujući i komponente računara, se mogu modelovati koristeći strukturu pod nazivom mašine konačnih stanja. Mašine konačnih stanja se dosta koriste u računarskim naukama i mrežama. Na primer, mašine konačnih stanja čine osnovu za “spell check”, proveru gramatike, indeksiranje u tekstu, prepoznavanju govora, transformacijama markap jezika kao što su XML i HTML, kao i mrežnih protokola koji omogućavaju komunikaciju računara. Na ovom predavanju ćemo izučiti mašine konačnih stanja koje imaju neki izlaz.

Sve mašine sa konačnim stanjima imaju skup stanja, uključujući početno stanje, ulaze i prelaznu funkciju koja dodeljuje sledeće stanje za svaki par stanja i ulaza. Neke mašine sa konačnim stanjima proizvode izlaze (izlazni simbol) za svaki prelaz iz jednog stanja u drugi; ove mašine se mogu koristiti za modelovanje mnogih vrsta mašina, uključujući automate (engl. vending machines), mašine sa kašnjenjem (engl. delay machines), binarne sabirače (engl. binary adder) i prepoznavače jezika. Takođe ćemo proučavati mašine konačnih stanja koje nemaju izlaz, ali imaju konačni broj stanja. Takve mašine se koriste u prepoznavanju jezika. Nizovi karaktera se koriste da se prepoznaju početna i konačna stanja.

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

KARAKTERISTIKE MAŠINE KONAČNIH STANJA

Mašina konačnih stanja počinje svoj životni vek sa početnim stanjem, a dalji prelazi iz jednog u drugo stanje se odvijaju kao posledica spoljnih ulaza

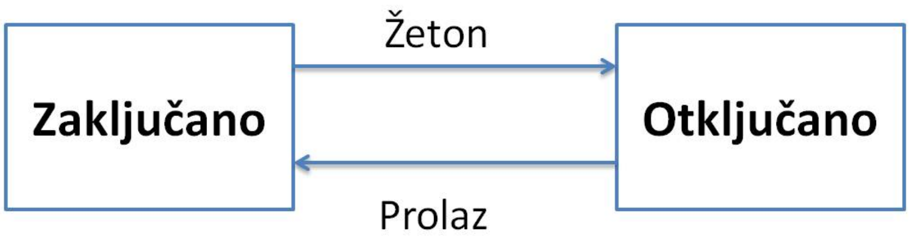
Mašina konačnih stanja (engl. finite state machine - FSM) je uređaj ili apstraktni model uređaja koji ima konačan broj stanja, počevši od tzv. početnog stanja, ulaznih informacija i funkcija prelaza koje koriste ulazne informacije za mapiranje trenutnog stanja u sledeće stanje.

Dakle, FSM počinje svoj životni vek sa početnim stanjem, a dalji prelazi iz jednog u drugo stanje se odvijaju kao posledica spoljnih ulaza, i pri tom mogu da se vrše određene akcije (tj. mogu imati izlaze kojim deluju na okolinu).
Prema tome, ideja FSM je da se ponašanje realnog objekta dekomponuje u konačni broj stanja i prelaz koji se dešavaju pod određenim uslovima.
Funkcije prelaza (engl. Transition function) su obično podeljene među stanjima i lake su za razumevanje jer svako stanje na osnovu ulaznih informacija zna šta može prouzrokovati prelaz u drugo stanje.

MODELOVANJE RADA ROTIRAJUĆE KAPIJE

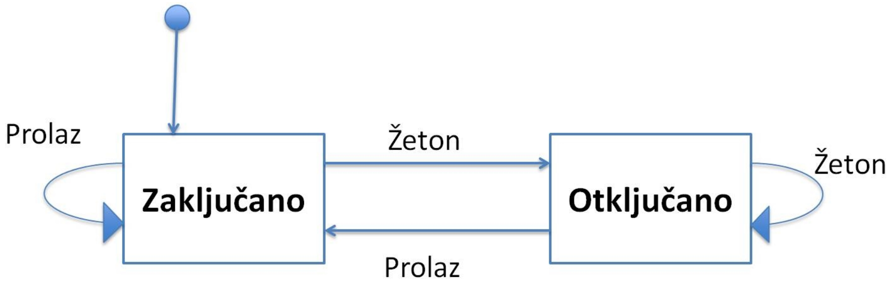
Rotirajuća kapija dozvoljava prolaz putnicima samo ako ubace

Razmotrimo kontrolu ulaza na stanicu autobuske stanice, gde rotirajuća kapija dozvoljava prolaz putnicima samo ako ubace žeton. Kapija je u prvobitno "zaključana" i dok je zaključana ona se ne okreće i ne dozvoljava putnicima da prođu. Kada se ubaci žeton, kapija je "otključana", ali se ne okreće sama, već se okreće tek kad kroz nju prođe osoba. Kada osoba prođe, kapija se ponovo zaključa. Iz ovog opisa bi trebalo da je jasno da rotirajuća kapija uvek može biti u 1 od 2 stanja: zaključana ili otključana. Postoje i samo dva tranziciona pokretača: deponovanje žetona i prolaz kroz kapiju.



Slika 4.1 Jednostavna mašina konačnih stanja za rotirajuću kapiju [Izvor: Autor]

Iako je ovo dobra aproksimacija funkcionisanja sistema, vidimo da određene informacije o ponašanju rotirajuće kapije nisu jasno definisane. Šta se dešava kada se žeton deponuje pre nego što neko prođe kroz kapiju? Slično tome, šta se dešava ako osoba pokuša proći kroz kapiju bez deponovanja žetona? U oba slučaja, naš jednostavan model bi trebao ostati u svom trenutnom stanju. Pokazujemo ovo ponašanje sa prelazima pomoću "petlji" na sledećoj slici. Takođe, dodajemo indikaciju početnog stanja sistema pomoću punog kruga i strelice koja pokazuje početno stanje.

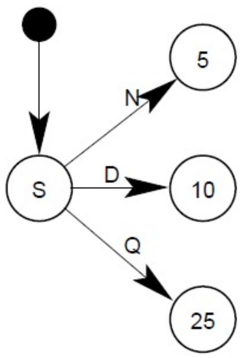


Slika 4.2 Unapređen model mašine konačnih stanja za rotirajuću kapiju [Izvor: Autor]

MODELOVANJE RADA APARATA ZA KAFU

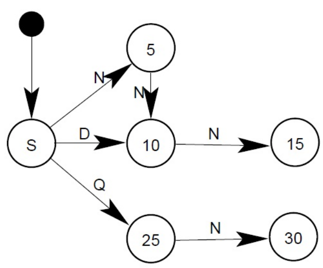
Analizu počinjemo tako što prvo definišemo moguće ulaze

Razmotrimo primer mašine konačnih stanja koja modeluje rad automata za kafu. Ova mašina sipa kafu samo jedne vrste i po ceni od 0,30. Mašina prima novčinje u vrednosti od 0,05, 0,10 i 0,25. Mašina ima dva dugmeta – jednu za sipanje kafe, drugo da vrati novac. Pošto je ovo čisto mašinski sistem, on nema memoriju, niti može da obavlja direktno aritmetičke operacije kao što su sabiranje i oduzimanje.
Počnimo našu analizu tako što ćemo prvo definisati moguće ulaze 0,05 (N), 0,10 (D), ,.25 (Q), dugme za sipanje (DB) i dugme za povraćaj novca (R). Kako bismo identifikovali stanja, počnimo od stanja koja dobijamo unošenjem kovanica. Ova stanja možemo imenovati kao vrednost unetog novca do tog momenta: 5, 10 ili 25. Ilustracija ovog modela je prikazana na slici



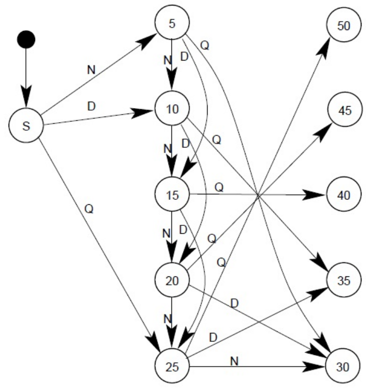
Slika 4.3 Početak modela mašine konačnih stanja za aparat za kafu [Izvor: Rosen]

Sledeće, razmotrimo šta se događa kada unesemo 0,05, nakon prvog novčića. Rezultujući model je prikazan na slici



Slika 4.4 Druga faza modela mašine konačnih stanja za aparat za kafu [Izvor: Rosen]

Dalje nastavljamo razvoj modela dok ne dođemo do stanja koje će reprezentovati ukupan novac potreban za kafu, odnosno 0,30, ali koji je manji od ,55 (pošto unos 0,25 posle 0,30 će automatski vratiti novac, bez da menja stanja). Ova verzija je prikazana na sledećoj slici



Slika 4.5 Model mašine konačnih stanja za aparat za kafu sa svim mogućim ulazima [Izvor: Rosen]

Poglavlje 5

MAŠINE KONAČNIH STANJA SA IZLAZOM

ELEMENTI MAŠINE KONAČNIH STANJA

Mašina konačnih stanja se sastoji od konačnog skupa stanja, nepraznog skupa ulaznog alfabeta, nepraznog skupa izlaznog alfabeta, funkcije prelaza, izlaznog i početnog stanja

Mašina konačnih stanja $M = (S, I, O, f, g, s_0)$se sastoji od konačnog skupa S stanja, neprazni skup ulazni alfabet I, neprazni skup izlazni alfabet O, funkcije prelaza f koja dodeljuje svakom paru ulaza i stanja, novo stanje. Izlazna funkcija g dodeljuje svakom paru stanja i ulaza njihov izlaz, kao i inicijalno stanje s_0 .

Ukoliko je ulazni niz $x = x_1x_2...x_k$, tada kada mašina pročita ulaz ona prelazi iz stanja s_0 u stanje s_1 , gde je $s_1 = f(s_0, x_1)$, pa prelazi u stanje s_2 , gde je $s_2 = f(s_1, x_2)$ i tako dalje ka $s_j = f(s_{j-1}, x_j)$za $j = 1, 2, \ldots, k$, a završava se sa stanjem $s_k = f(s_{k-1}, x_k)$. Ovaj niz prelaza proizvodi izlaz $y_1y_2...y_k$, gde je $y_1 = g(s_0, x_1)$. y_1 je izlaz koji odgovara prelazu iz stanja s_0 u stanje s_1 . Izlaz $y_2 = g(s_1, x_2)$ je izlaz prelaza iz stanja s_1 u stanje s_2 . Uopšteno gledano, $y_j = g(s_{j-1}, x_j)$ za $j = 1, 2, \ldots, k$.

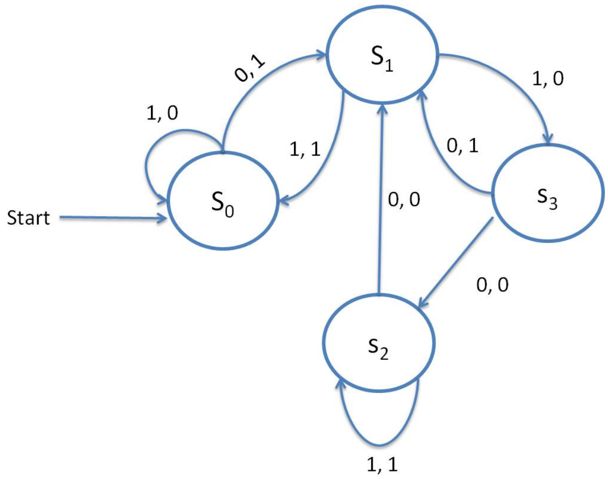
Primer

Tablica stanja prikazana u tabeli opisuje mašinu konačnih stanja sa $S = \{s_0, s_1, s_2, s_3\}$, $I = \{0, 1\}$ i $O = \{0, 1\}$. Vrednosti funkcije prelaza f su prikazane u prve dve kolone, a vrednosti izlazne funkcije g su prikazane u poslednje dve kolone. Napravite dijagram stanja za mašinu sa brojem stanja i tranzicijama koji su definisani sledećom tabelom

Stanje	**f**		**g**	
	Ulaz		**Ulaz**	
	0	**1**	**0**	**1**
s_0	s_1	s_0	1	0
s_1	s_3	s_0	1	1
s_2	s_1	s_2	0	1
s_3	s_2	s_1	0	0

Slika 5.1 Tabela-1 Tablica stanja [Izvor: Autor]

Rešenje:

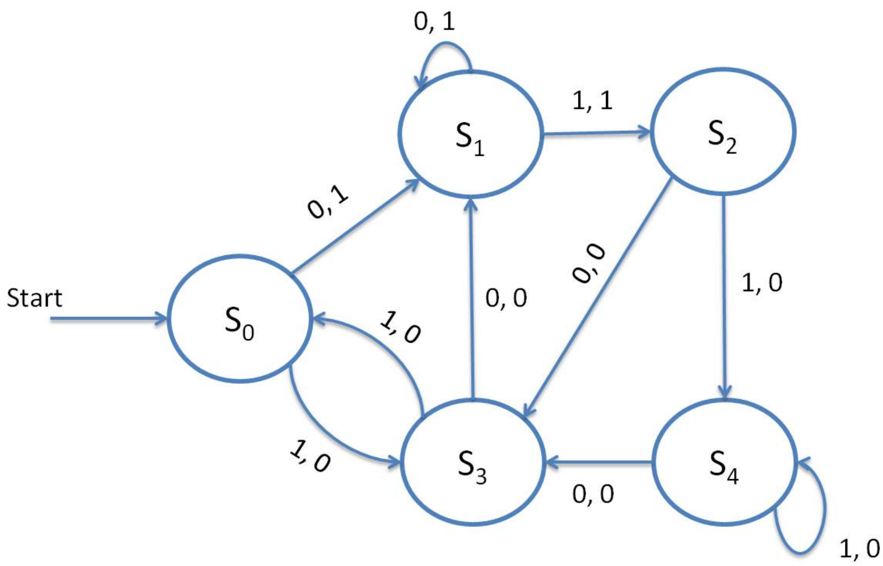


Slika 5.2 Dijagram stanja [Izvor: Autor]

ZADATAK 1

Odrediti izlazni niz generisan od strane mašine konačnih stanja

Odredite tablicu stanja za mašinu konačnih stanja opisanu sledećom slikom. Odrediti izlazni niz generisan od strane mašine konačnih stanja ako je ulazni niz 101011.



Slika 5.3 Mašina konačnih stanja [Izvor: Autor]

Rešenje:

Stanje	f		g	
	Ulaz		Ulaz	
	0	1	0	1
s_0	s_1	s_3	1	0
s_1	s_1	s_2	1	1
s_2	s_3	s_4	0	0
s_3	s_1	s_0	0	0
s_4	s_3	s_4	0	0

Slika 5.4 Tabela-2 Tablica stanja [Izvor: Autor]

Ulaz	1	0	1	0	1	1	-
Stanje	s_0	s_3	s_1	s_2	s_3	s_0	s_3
Izlaz	0	0	1	0	0	0	-

Slika 5.5 Tabela-3 Određivanje izlaznog niza [Izvor: Autor]

ZADATAK 2

Konstruišite mašinu konačnih stanja koja daje 1 kao svoj trenutni izlazni bit ako i samo ako su poslednja tri primljena bita 1.

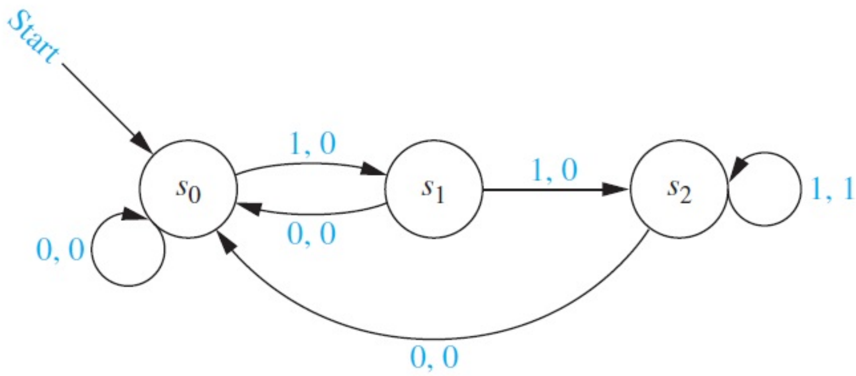
Zadatak

U određenoj šemi šifriranja, kada se u poruci pojavljuju tri uzastopne 1, prijemnik poruke zna da je došlo do greške pri prenosu. Konstruišite mašinu konačnih stanja koja daje 1 kao svoj trenutni izlazni bit ako i samo ako su poslednja tri primljena bita 1.

Rešenje:

Za ovu mašinu je potrebno definisati tri stanja. Početno stanje s_0 pamti da je prethodna ulazna vrednost, ako uopšte postoji, nije bila 1. Stanje s1 pamti da je prethodni ulaz bio 1, ali ulaz pre prethodnog unosa, ako postoji, nije bio 1. Stanje s_2 pamti da su prethodna dva ulaza bila 1.

Ulazna vrednost 1 dovodi do prelaza iz stanja s_0 u stanje s_1, jer je primljena prva 1. Ulazna vrednost 1 dalje dovodi do prelaza iz stanja s_1 u s_2, jer su sada učitane dve uzastopne 1. Stanje s_2 vodi u samo sebe, pošto su već učitane najmanje dve uzastopna 1. Ulazna vrednost 0 dovodi do prelaza svih stanja u stanje s_0.



Slika 5.6 Mašina konačnih stanja koja modeluje detektovanje greške kada se prime tri uzastopna bita vrednosti 1 [Izvor: Rosen]

Poglavlje 6

MAŠINE KONAČNIH STANJA BEZ IZLAZA

PRIMER MAŠINE KONAČNIH STANJA BEZ IZLAZA

Mašina konačnih stanja bez izlaza se zove i konačni automat

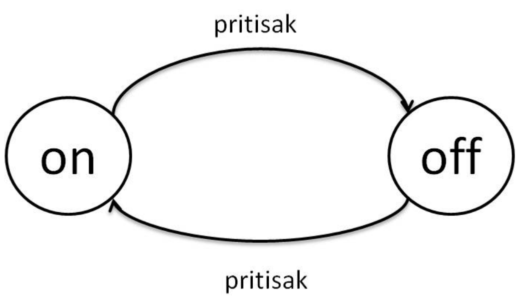
Jedna od najvažnijih primena mašina konačnih stanja je prepoznavanje jezika. Ova primena igra osnovnu ulogu u dizajniranju i konstrukciji kompajlera za programerske jezike.

Mašina konačnih stanja bez izlaza se zove i konačni automat. Ove mašine ne proizvode izlaz, ali imaju krajnje stanje.
Verovatno najjednostavniji primer konačnog automata je”on/off” prekidač koji koristimo za paljenje i gašenje svetla.
Prekidač može da ima dva 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 – pritiskom na dugme se iz bilo kog od ta dva stanja prelazi u ono drugo.

Jedan od načina da se prikažu automati je da se koriste usmereni grafovi. U ovim grafovima je svako stanje prikazano krugom. Grane su imenovane ulazom i izlazom za svaki prelaz između stanja. Rad prekidača se grafički može prikazati na sledeći način:



Slika 6.1 Model rada on/off prekidača [Izvor: Autor]

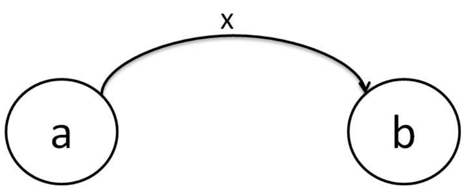
FORMALNA DEFINICIJA

Automat se definiše kao uređena trojka $A = (A, X, \delta)$

Automat se definiše kao uređena trojka $A = (A, X, \delta)$ 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, tamo gde ne postoji opasnost od zabune. Princip rada ovako definisanog automata je sledeći:
  • automat A se u određenom trenutku nalazi u stanju a ∈ A, a na njegov ulaz dospeva ulazni signal x ∈ X;
  • pod dejstvom tog signala automat menja svoje stanje, i u narednom trenutku prelazi u stanje $\delta(a, x) \in A$.

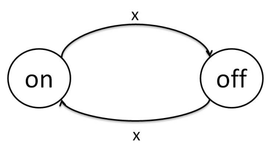
Ovo ćemo grafički predstavljati na sledeći način:



Slika 6.2 Grafički prikaz konačnog automata [Izvor: Autor]

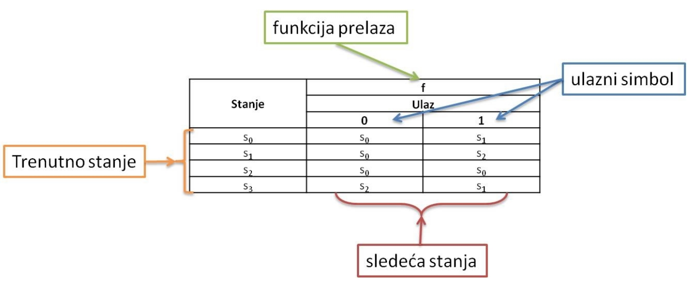
Razmotrimo”on/off” prekidač. Ako ulazni signal ”pritisak na dugme” kraće označimo sa x, onda imamo da je A = {on, off}, X = {x}, a funkcija prelaza $\delta : A \times X \to A$ je zadata sa $\delta(on, x) = off, \delta(off, x) = on$. To se može dati i tabelarno, odnosno grafički, na sledeće načine:

	off	on
x	on	off



Slika 6.3 Konačni automat za model on/off prekidača [Izvor: Autor]

Automat je takođe veoma zgodno zadavati i takozvanim tablicama prelaza. Tablica prelaza automata $A = (A, X, \delta)$ je pravougaona tablica sa vrstama koje odgovaraju svakom ulaznom simbolu i kolonama koje odgovaraju stanjima. Na mestu u tablici koje odgovara vrsti određenoj koloni određenim stanjem a ∈ A i ulaznim simbolom x ∈ X upisuje se stanje $\delta(a, x)$. To je prikazano u sledećoj tablici



Stanje	f	
	Ulaz	
	0	1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_0
s_3	s_2	s_1

Slika 6.4 Prikaz tablice prelaza [Izvor: Autor]

PRIMER

Odrediti dijagram stanja za konačni automat

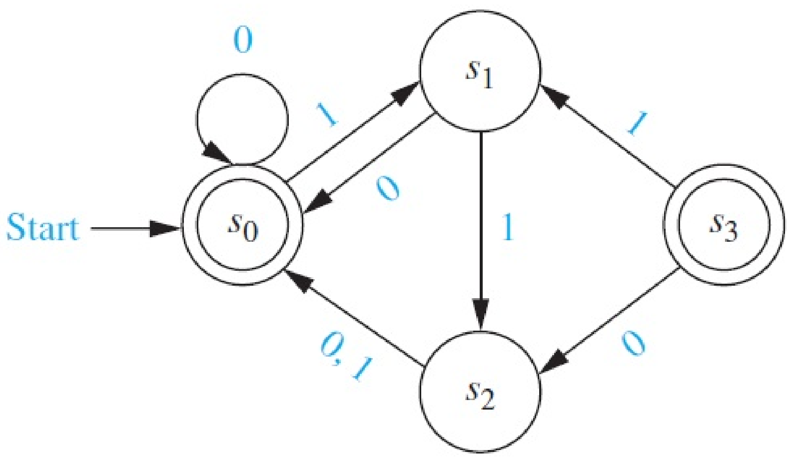
Zadatak

Odrediti dijagram stanja za konačni automat M = (S, I, f, s_0, F), gde je S = {s_0, s_1, s_2, s_3}, I = {0, 1}, F = {s_0, s_3}, a funkcija prelaza f je data u tabeli

Stanje	f	
	Ulaz	
	0	1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_0
s_3	s_2	s_1

Slika 6.5 Tabela-1 Tablica prelaza [Izvor: Autor]

Rešenje:



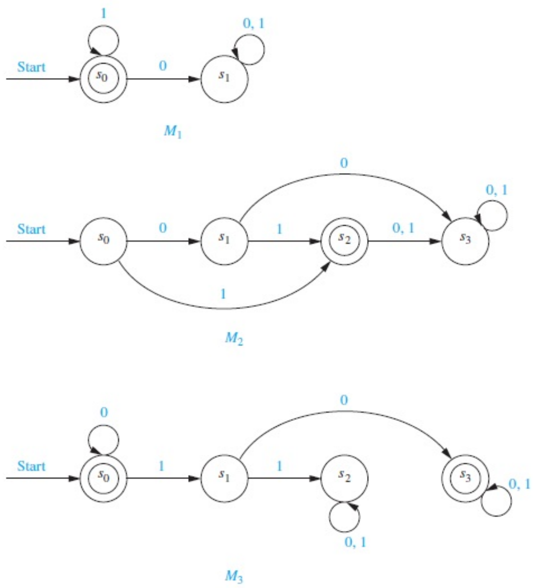
Slika 6.6 Dijagram stanja za konačni automat [Izvor: Rosen]

PREPOZNAVANJE JEZIKA

Jezik koji je prepoznat ili prihvaćen od strane mašine M, označava se sa L (M)

Niz x je k prepoznat ili prihvaćen od strane mašine M = (S, I, f, s_0, F) ako postoje prelazi koji vode iz početnog stanja s0 do konačnog stanja, odnosno ako je f (s_0, k) stanje u F. Jezik koji je prepoznat ili prihvaćen od strane mašine M, označava se sa L (M) i predstavlja skup svih nizova koji su prepoznati od strane M. Dva konačna automata su ekvivalentna ako prepoznaju isti jezik.

Odredite jezike koje prepoznaju automati M_1, M_2 i M_3



Slika 6.7 Automati M_1 , M_2 i M_3 [Izvor: Rosen]

Rešenje:
Jedino konačno stanje u M_1 je s_0. Niz koji vodi stanje s0 u samog sebe se sastoje od 0 ili više uzastopnih 1. Tako, $L(M_1) = \{1^n \mid n = 0,1,2,...\}$.
Jedino konačno stanje M_2 je s_2. Nizovi koji dovode do prelaza iz stanja s_0 u stanje s_2 su 1 i 01. Tako, $L(M_2) = \{1, 01\}$.
Konačna stanja mašine M_3 su s_0 i s_3. Jedini nizovi koji vode stanje s_0 u samog sebe su 0, 00, 000, . . . , odnosno nizovi koji sadrže nijednu ili više 0. Jedini niz koji vodi iz stanja s_0 u stanje s_3 su nizovi koji sadrže nijednu ili više uzastopnih 0, nakon čega sledi 10, a zatim bilo koji niz. Tako, $L(M_3) = \{0^n, 0^n10x \mid n = 0,1,2,..., \text{ i } x \text{ je bilo koji niz}\}$.

PROJEKTOVANJE KONAČNIH AUTOMATA

Konstruišite deterministički konačni automat

Često možemo konstruisati konačan automat koji prepoznaje određeni niz pažljivim dodavanjem stanja i prelazaka i određivanjem koje od ovih stanja bi trebalo da bude konačno stanje. Kada je to potrebno, uključujemo i stanja koja mogu pratiti neke od osobina ulaznog niza.

Zadatak
Konstruišite deterministički konačni automat koji prepoznaje svaki od ovih jezika.
(a) skup nizova bitova koji počinju sa dve 0
(b) skup nizova bitova koji sadrže dve uzastopne 0
(c) skup nizova bitova koji ne sadrže dve uzastopne 0

(d) skup nizova bitova koji se završavaju sa dve 0
(e) skup nizova bitova koji sadrže najmanje dve 0.

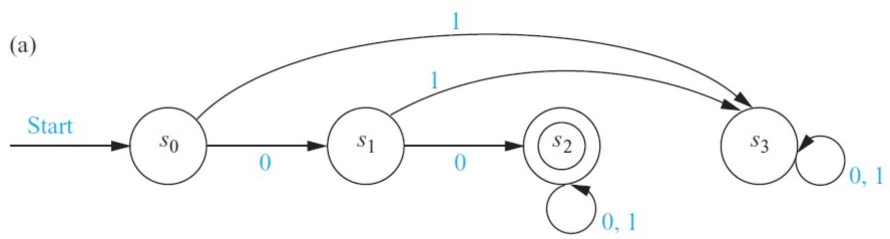
KONAČNI AUTOMAT KOJI PREPOZNAJE SKUP NIZOVA BITOVA KOJI POČINJU SA DVE 0

Naš cilj je da konstruišemo deterministički konačni automat koji prepoznaje skup nizova bitova koji počinju sa dve 0

(a) skup nizova bitova koji počinju sa dve 0

Rešenje:

(a) Naš cilj je da konstruišemo deterministički konačni automat koji prepoznaje skup nizova koji počinju sa dve 0. Pored početnog stanja s_0, uključujemo i ne-konačno stanje s_1. Iz stanja s_1 prelazimo u stanje s_0 ako je prvi bit 0. Zatim dodajemo konačno stanje s_2, u koje dolazimo iz stanja s_1 i to kada je drugi bit 0. Kada smo dosegli s_2 znamo da su prva dva ulazna bitova oba 0, tako da ostajemo u stanju s_2 bez obzira na to koji su naredni bitovi (ako ih ima). Prelazimo u konačno stanje s_3 iz s_0 ako je prvi bit 1 i iz s_1 ako je drugi bit 1



Slika 6.8 Konačni automat koji prepoznaje skup nizova bitova koji počinju sa dve 0 [Izvor: Rosen]

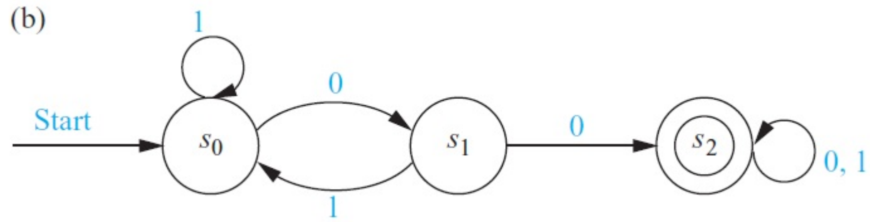
KONAČNI AUTOMAT KOJI PREPOZNAJE SKUP NIZOVA BITOVA KOJI SADRŽE DVE UZASTOPNE 0

Naš cilj je da konstruišemo deterministički konačni automat koji prepoznaje skup nizova bitova koji sadrže dve uzastopne 0

(b) skup nizova bitova koji sadrže dve uzastopne 0

Rešenje:

(b) Naš cilj je da konstruišemo deterministički konačni automat koji prepoznaje skup nizova koji sadrže dve uzastopne 0. Osim početnog stanja s_0, uključujemo i ne-konačno stanje s_1, što nam govori da je poslednji ulazni bit bio 0, a bit pre toga je ili bio 1 ili je ovo početni bit niza. Uključujemo konačno stanje s_2 u koje prelazimo iz stanja s_1 kada je ulazni bit nakon 0 takođe 0. Ako je primljen ulazni bit 1 nakon 0 u nizu (pre nego što učitamo dve uzastopne 0), u tom slučaju se vraćamo na stanje s_0 i započinjemo ponovo sa traženjem uzastopnih 0.



Slika 6.9 Konačni automat koji prepoznaje skup nizova bitova koji sadrže dve uzastopne 0 [Izvor: Rosen]

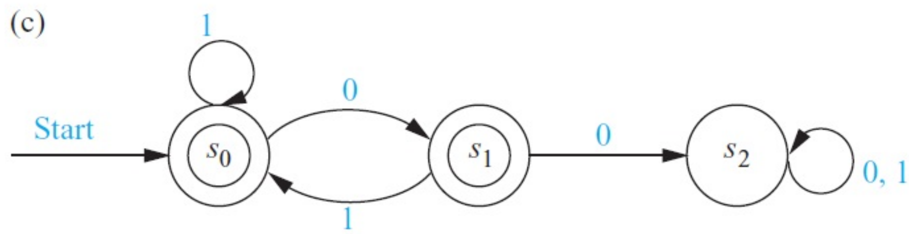
KONAČNI AUTOMAT KOJI PREPOZNAJE SKUP NIZOVA BITOVA KOJI NE SADRŽE DVE UZASTOPNE 0

Naš cilj je da konstruišemo deterministički konačni automat koji prepoznaje skup nizova bitova koji ne sadrže dve uzastopne 0

(c) skup nizova bitova koji ne sadrže dve uzastopne 0

Rešenje:

(c) Naš cilj je da konstruišemo deterministički konačni automat koji prepoznaje skup nizova koji ne sadrže dve uzastopne 0. Pored početnog stanja s_0, koji treba da bude i konačno stanje, uključujemo i konačno stanje s_1. U stanje s_1 se prelazi iz stanja s_0 kada je prvi ulazni bit 0. Kada je ulazni bit 1, tada ostajemo u stanju s_0 ili se u njega vraćamo. Dodajemo stanje s_2 u koje prelazimo iz stanja s_1 kada je ulazni bit 0. Dolazak u stanje s_2 nam govori da smo učitali dve uzastopna ulazna bita koji su 0. Ostajemo u stanju s_2 kada u njega stignemo; ovo stanje nije konačno.



Slika 6.10 Konačni automat koji prepoznaje skup nizova bitova koji ne sadrže dve uzastopne 0 [Izvor: Rosen]

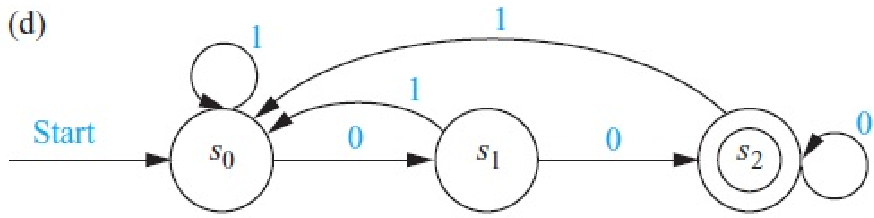
KONAČNI AUTOMAT KOJI PREPOZNAJE SKUP NIZOVA BITOVA KOJI SE ZAVRŠAVAJU SA DVE 0

Naš cilj je da konstruišemo deterministički konačni automat koji prepoznaje skup nizova bitova koji se završavaju sa dve 0

(d) skup nizova bitova koji se završavaju sa dve 0

Rešenje:

(d) Naš cilj je da konstruišemo deterministički konačni automat koji prepoznaje skup nizova koji se završavaju sa dve 0. Pored početnog stanja s_0, uključujemo i ne-konačno stanje s_1 u koje prelazimo kada je prvi učitani bit 0. Uključujemo konačno stanje s_2 do kojeg stižemo iz stanja s_1, ako je sledeći ulazni bit nakon 0 takođe 0. Ako posle prethodne 0 sledi još jedan ulazni bit koji je 0, tada ostajemo u stanju s_2, pošto su poslednja dva ulazna bitova i dalje 0. Kada smo u stanju s_2, ulazni bit vrednosti 1 će dovesti do prelaska u stanje s_0, kada se ponovo započinje traženje uzastopnih 0. Takođe se vraćamo na s0 iz stanja s_1, ako je sledeći ulazni bit 1.



Slika 6.11 Konačni automat koji prepoznaje skup nizova bitova koji se završavaju sa dve 0 [Izvor: Rosen]

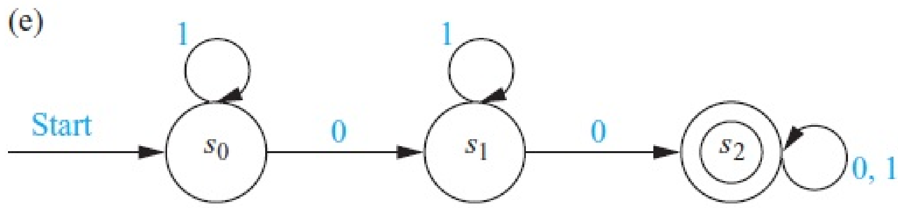
KONAČNI AUTOMAT KOJI PREPOZNAJE SKUP NIZOVA BITOVA KOJI SADRŽE NAJMANJE DVE 0

Naš cilj je da konstruišemo deterministički konačni automat koji prepoznaje skup nizova bitova koji sadrže najmanje dve 0

(e) skup nizova bitova koji sadrže najmanje dve 0

Rešenje:

(e) Naš cilj je da konstruišemo deterministički konačni automat koji prepoznaje skup nizova koji sadrže dve 0. Pored početnog stanja uključujemo i stanje s_1, koje nije konačno. Ostajemo u stanju s0 sve dok ulazni bit nije 0 i prelazimo u stanje s1 kada naiđemo na prvu 0. Dodajemo konačno stanje s_2 u koje prelazimo iz stanja s_1, što će se dogoditi kada se susretnemo sa drugom 0. Kad god je ulazni bit 1, tada ostajemo u trenutnom stanju. Kada stignemo do stanja s_2, tamo i ostajemo. Stanja s_1 i s_2 se koriste da nam signalizairaju da smo učitali jednu ili dve 0 (s_1 za jednu, a s_2 za dve 0).

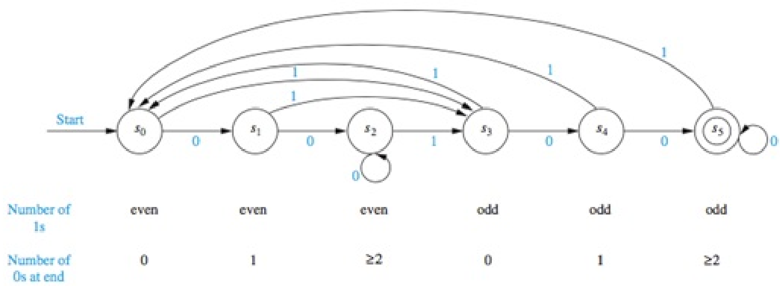


Slika 6.12 Konačni automat koji prepoznaje skup nizova bitova koji sadrže najmanje dve 0 [Izvor: Rosen]

DETERMINISTIČKI KONAČNI AUTOMAT KOJI PREPOZNAJE SKUP NIZA BITOVA KOJI SADRŽI NEPARAN BROJ 1

Definišimo stanja koja vode evidenciju i o paritetu bita 1 i broj 0 na kraju ulaznog niza

Zadatak
Odrediti deterministički konačni automat koji prepoznaje skup niza bitova koji sadrže neparan broj 1 i koji se završavaju sa barem dve uzastopne 0.
Rešenje:
Možemo da odredimo deterministički konačni automat koji prepoznaje tražene skupove nizova ako definišemo stanja koja vode evidenciju i o paritetu bita 1, kao i da li smo naišli na nijednu, jednu ili više 0 na kraju ulaznog niza.
Početno stanje s_0 se može koristiti da indikuje da li je očitan paran ili neparan broj 1 i da li se završava sa 0. Pored početnog stanja uključujemo još pet stanja:
S_1 - kada je ulazni niz sa parnim brojem 1 i kad se završava sa jednom 0
S_2 - kada je ulazni niz sa parnim brojem 1 i kad se završava sa barem dve 0
S_3 – kada ulazni niz sadrži neparan broj 1 i kad se ne završava sa 0
S_4 - kada ulazni niz sadrži neparan broj 1 i kad se završava sa jednom 0
S_5 - kada ulazni niz sadrži neparan broj 1 i kad se završava sa dve 0. S_5 je konačno stanje.



Slika 6.13 Deterministički konačni automat koji prepoznaje skup niza bitova koji sadrže neparan broj 1 i koji se završavaju sa barem dve uzastopne 0 [Izvor: Rosen]

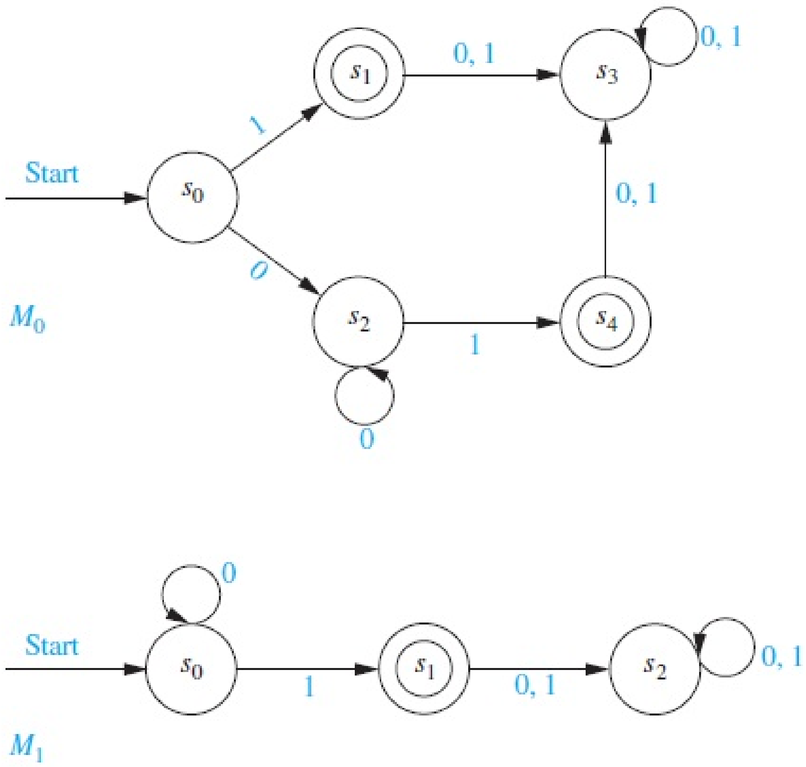
EKVIVALENTNI KONAČNI AUTOMATI

Dva automata su jedna ako prepoznaju isti jezik

Dva automata su jedna ako prepoznaju isti jezik.
Pokažimo da su dva konačna automata M_0 i M_1 jednaka.
Za niz x koji prepoznaje mašina M0, mašina mora proći od početnog stanja s_0 do konačnog stanja s_4. Jedini niz koji vodi od s_0 do s_1 je 1. Nizovi koji nas vode od s_0 do s_4 su oni nizovi koji počinju sa 0, koji dovode do prelaza od s_0 do s_2; nakon čega ako sledi nijedna ili više 0, drže mašinu u stanju s_2; nakon čega bit 1, dovodi do prelaza iz stanja s_2 u konačno stanje s_4. Svi drugi nizovi vode iz s_0 u stanja koja nisu konačna. Ostavljeno je da student pokaže ostatke detalja. Ovde zaključujemo da je L(M_0) skup nizova sa ni jednom ili više 0, nakon

kojih sledi 1.

Za niz x koji prepoznaje mašina M $_1$, x mora da nas sprovede od početnog stanja s0 do konačnog stanja s $_1$. Dakle, da bi x bio prepoznat mora da počinje sa nekim brojem 0, što mašinu ostavlja u stanju s $_0$; nakon čega ako sledi 1 to dovodi do prelaza u konačno stanje s1. Niz svih nula nije prepoznat zato što nas ostavlja u stanju s0, a koje nije konačno stanje. Svi nizovi koji sadrže 0 posle 1, nisu prepoznati zato što nas vode u stanje s $_2$, koje nije konačno. Sledi da je L(M $_1$) isti kao i L(M $_0$). Zaključujemo da su M $_0$ i M $_1$ ekvivalentni.



Slika 6.14 Dva ekvivalentna konačna automata [Izvor: Rosen]

Poglavlje 7

VEŽBA - PARCIJALNO UREĐENJE

ZADATAK 1.

U posetu (Z+, |) da li su 3 i 9 uporedivi elementi

Predviđeno vreme trajanja: 5 minuta

Zadatak:

U posetu ($\mathbf{Z}^+$, |) da li su 3 i 9 uporedivi elementi? Da li su 5 i 7 uporedivi?

Rešenje:

Brojevi 3 i 9 su uporedivi zato što 3 | 9, odnosno (3, 9).

Brojevi 5 i 7 nisu uporedivi zato što 5 ne deli 7 i 7 ne deli 5.

ZADATAK 2.

Primer leksikografskog uređenja

Predviđeno vreme trajanja: 5 minuta

Zadatak:

Odrediti da li su sledeći parovi poseta (ZxZ, $\preccurlyeq$), gde $\preccurlyeq$ predstavlja leksikografsko uređenje

$(3,5) \preccurlyeq (4,8)$

$(3,8) \preccurlyeq (4,5)$

$(4,9) \preccurlyeq (4,11)$

Rešenje:

Zato što je $3<4$ sledi da je $(3,5) \preccurlyeq (4,8)$.

Zato što je $3<4$ sledi da je $(3,8) \preccurlyeq (4,5)$.

Zato što je $4=4$, upoređujemo i drugu koordinatu, odnosno

je $9<11$ sledi da je $(4,9) \preccurlyeq (4,11)$.

ZADATAK 3.

Primer Hasovog dijagrama

Predviđeno vreme trajanja: 10 minuta

Zadatak:

Nacrtati Hasov dijagram poseta (S,R), tako da S={1,2,3,4,6,8,12}, a relacija R predstavlja parcijalno uređenje nad skupom S tako da a|b za a,b∈ S. Odrediti minimalne i maksimalne elemente poseta.

Rešenje:

Digraf ovog parcijalnog uređenja možemo prikazati na sledeći način:

(1, 1), (1, 2), (1, 3), (1, 4), (1, 6), (1, 8), (1, 12)

(2, 2), (2, 4), (2, 6), (2, 8), (2, 12)

(3, 3), (3, 6), (3, 12)

(4, 4), (4, 8), (4, 12)

(6, 6), (6, 8), (6, 12)

(8, 8)

(12, 12)

Prvo uklonimo sve petlje, a potom i sve grane koje impliciraju tranzitivnost, a to su:

(1, 4), (1, 6), (1, 8), (1, 12)

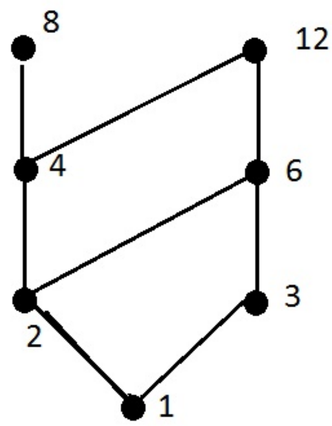
(2, 8), (2, 12)

(3, 12)

Zatim organizujemo čvorove tako da se grane usmere na gore i strelice uklonimo.

Rezultujući Hasov dijagram se nalazi na slici.

Minimalni element je 1. Maksimalni elementi su 8 i 12.



Slika 7.1 Prikaz rešenja primera 3 [Izvor: Autor]

ZADATAK 4.

Nacrtati Hasov dijagram poseta (P(S),R), tako da je P(S) partitivni skup

Predviđeno vreme trajanja: 15 minuta

Zadatak:

Nacrtati Hasov dijagram poseta (P(S),R), tako da je P(S) partitivni skup skupa S={a,b,c}, a relacija R predstavlja parcijalno uređenje nad P(S), tako da (A,B) kada je A⊂B za A,B∈ P(S).

Rešenje:

P(S)=$\{\emptyset, \{a\}, \{b\}, \{c\}, \{a,b\}, \{a,c\}, \{b,c\}, \{a,b,c\}\}$

Parcijalno uređenje definisano u zadatku definiše sledeće uređene parove

$(\emptyset,\emptyset)$, $(\emptyset,\{a\})$, $(\emptyset,\{b\})$, $(\emptyset,\{c\})$, $(\emptyset,\{a,b\})$, $(\emptyset,\{b,c\})$, $(\emptyset,\{a,c\})$, $(\emptyset,\{a,b,c\})$

$(\{a\},\{a\})$, $(\{a\},\{a,b\})$, $(\{a\},\{a,c\})$, $(\{a\},\{a,b,c\})$

$(\{b\},\{b\})$, $(\{b\},\{a,b\})$, $(\{b\},\{b,c\})$, $(\{b\},\{a,b,c\})$

$(\{c\},\{c\})$, $(\{c\},\{a,c\})$, $(\{c\},\{b,c\})$, $(\{c\},\{a,b,c\})$

$(\{a,b\},\{a,b,c\})$, $(\{a,c\},\{a,b,c\})$, $(\{b,c\},\{a,b,c\})$

$(\{a,b\},\{a,b\})$, $(\{a,c\},\{a,c\})$, $(\{b,c\},\{b,c\})$

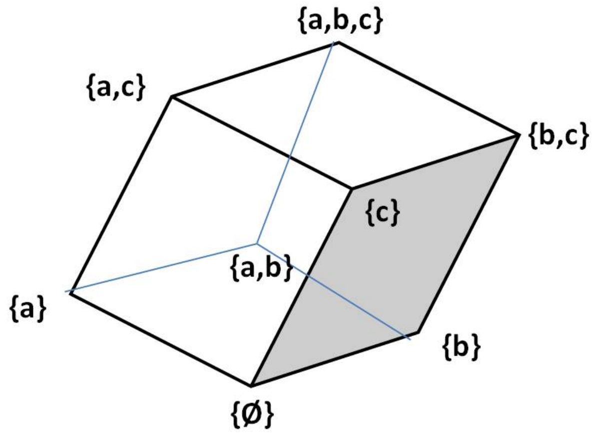
$(\{a,b,c\},\{a,b,c\})$

Grane koje se odnose na tranzitivnost su

$(\emptyset, \{a, b\})$, $(\emptyset, \{b, c\})$, $(\emptyset, \{a, c\})$, $(\emptyset, \{a, b, c\})$

$(\{a\}, \{a, b, c\})$, $(\{b\}, \{a, b, c\})$, $(\{c\}, \{a, b, c\})$

Hasov dijagram posete se nalazi na slici



Slika 7.2 Rešenje primera 4 [Izvor: Autor]

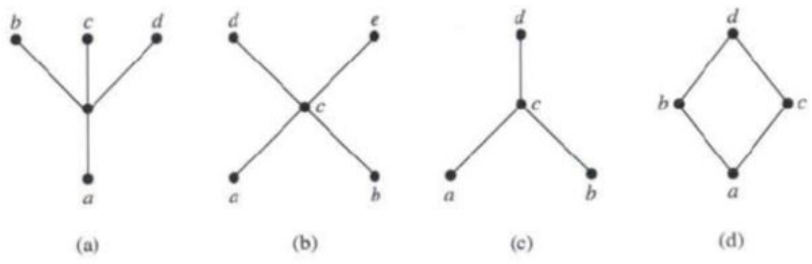
ZADATAK 5.

Najveći i najmanji element u Hasovom dijagramu

Predviđeno vreme trajanja: 5 minuta

Zadatak:

Odrediti najveće i najmanje elemente poseta određenim Hasovim dijagramima na slici:



Slika 7.3 Rešenje primera 4 [Izvor: Autor]

Rešenje:

a) Najmanji element – a

Najveći element – ne postoji

b) Najmanji element – ne postoji

Najveći element – ne postoji

c) Najmanji element – ne postoji

Najveći element – d

d) Najmanji element – a

Najveći element – d

ZADATAK 6.

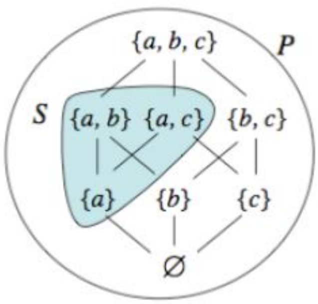
Posmatrajmo poset (P,R) čiji je Hasov dijagram prikazan na slici

Predviđeno vreme trajanja: 5 minuta

Zadatak:

Posmatrajmo poset (P,R) čiji je Hasov dijagram prikazan na slici. Neka je S podskup određen kao što je prikazano na slici.

Odrediti minimalni i maksimalni element, najmanji i najveći element podskupa S.



Slika 7.4 Rešenje primera 5 [Izvor: Rosen]

Rešenje:

Minimalni element – {a}

Maksimalni element – {a,b}, {a,c}

Najmanji element – {a}

Najveći elemenet – ne postoji

ZADATAK 7.

Odrediti minimalni i maksimalni element, najmanji i najveći element, gornje i donje međe

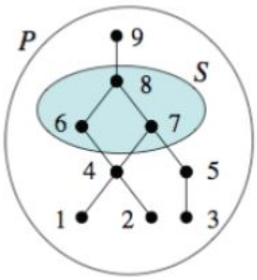
Predviđeno vreme trajanja: 5 minuta

Zadatak:

Posmatrajmo poset (P,R) čiji je Hasov dijagram prikazan na slici.

Neka je S podskup određen kao što je prikazano na Slici.

Odrediti minimalni i maksimalni element, najmanji i najveći element podskupa S.



Slika 7.5 Rešenje primera 7 [Izvor: Rosen]

Rešenje:

Minimalni element – 6, 7

Maksimalni element – 8

Najmanji element – ne postoji

Najveći element – 8

ZADATAK 8.

Koristeći topološko sortiranje, birajući prvo minimalne elemente, sortirajte elemente poseta

Predviđeno vreme trajanja: 5 minuta

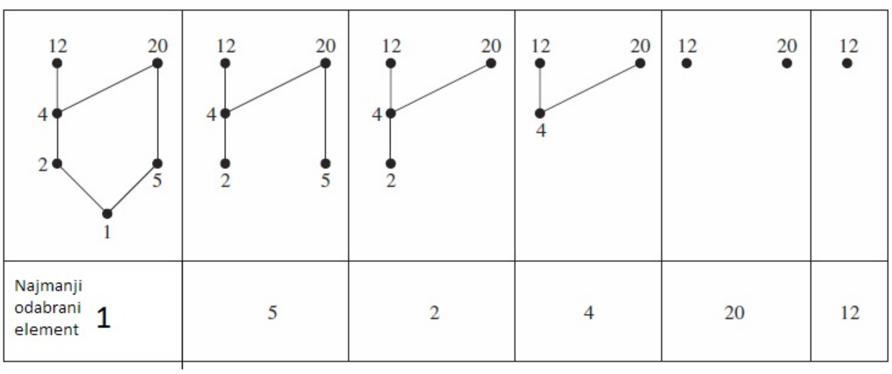
Zadatak:

Koristeći topološko sortiranje, birajući prvo minimalne elemente, sortirajte elemente poseta

({1,2,4,5,12,20}, |).

Rešenje:

Hasov dijagram posete i rešenje se nalaze na slici



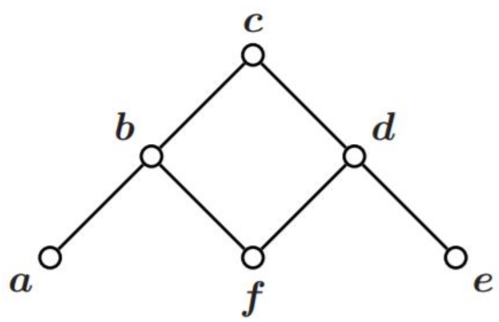
Slika 7.6 Rešenje primera 8 [Izvor: Autor]

ZADATAK 9.

Obnavljanje minimalnih i maksimalnih elemenata

Predviđeno vreme trajanja: 5 minuta

Na grafu na slici



Slika 7.7 Zadatak9 [Izvor: Autor]

Naći elemente koji nisu uporedivi sa elementom a.

Naći minimalne i maksimalne elemente.

Naći najmanji i najveći element.

REŠENJE:
(a) Elementi neuporedivi sa a su:
f, d, e

(b) Minimalni elementi su:
a, f, e

(c) Maksimalni element je:
c

(d) Najmanji elemnet ne postoji.

(e) Najveći element je c.

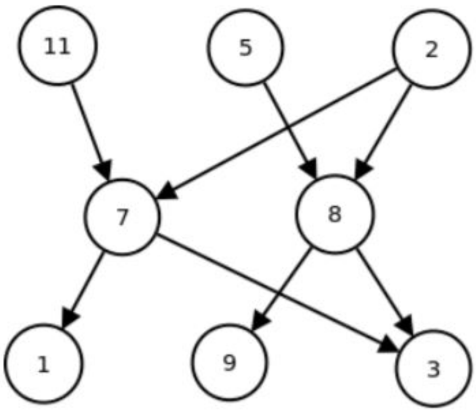
ZADATAK 10.

Odrediti topološko sortiranje datog grafa gde se uvek među dostupnim čvorovima bira onaj sa najmanjim brojem

Predviđeno vreme trajanja: 5 minuta

a) Odrediti topološko sortiranje datog grafa gde se uvek među dostupnim čvorovima bira onaj sa najmanjim brojem.

b) Odrediti još jedno, proizvoljno topološko sortiranje istog grafa



Slika 7.8 Zadatak 10 [Izvor: Autor]

Rešenje:

a)
dostupni: 2,5,11 izabran: 2
dostupni: 5,11 izabran: 5
dostupni: 8,11 izabran: 8
dostupni: 9,11 izabran: 9
dostupni: 11 izabran: 11
dostupni: 7 izabran: 7
dostupni: 1,3 izabran: 1
dostupni: 3 izabran: 3

b) 11 5 2 7 8 1 9 3

ZADATAK 11.

Dokazati da je ρ relacija poretka, nacrtati njen Hasov dijagram, i odrediti najveći, najmanji, minimalne i maksimalne elemente

Predviđeno vreme trajanja: 5 minuta

Data je binarna relacija

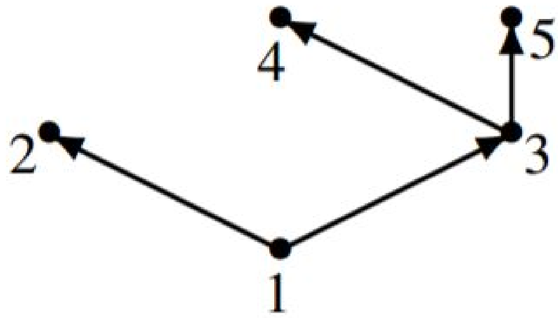
ρ = {(1,1),(1,2),(1,3),(1,4),(1,5),(2,2),(3,3),(3,4),(3,5),(4,4),(5,5)}

na skupu A = {1,2,3,4,5}.
Dokazati da je ρ relacija poretka, nacrtati njen Hasov dijagram, i odrediti najveći, najmanji, minimalne i maksimalne elemente.

Rešenje:

Utvrdujemo da ρ ima osobine R, A i T.

najmanji element: 1
najveci element: nema
minimalni element: 1
maksimalni element: 2,4,5



Slika 7.9 Rešenje zadatka 12 [Izvor: Autor]

Poglavlje 8

VEŽBA - MAŠINE KONAČNIH STANJA

ZADATAK 1.

Odrediti dijagram stanja za konačni automat

Predviđeno vreme trajanja: 5 minuta

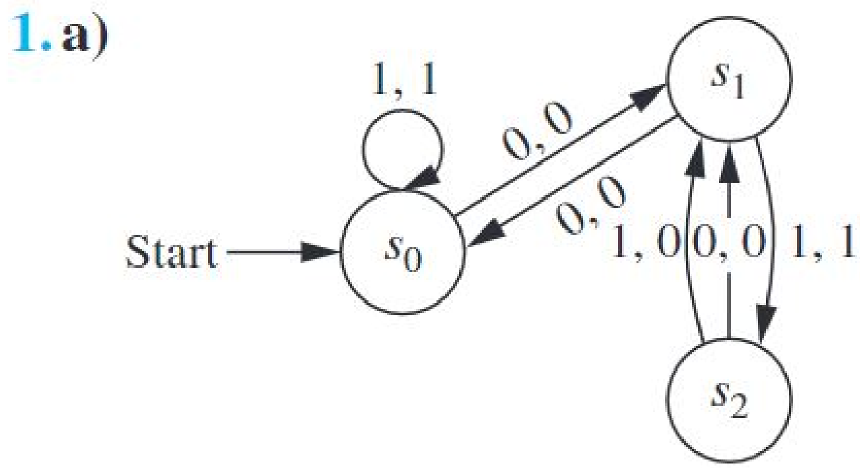
Zadatak

Odrediti dijagram stanja za konačni automat

a)

State	f Input 0	f Input 1	g Input 0	g Input 1
s_0	s_1	s_0	0	1
s_1	s_0	s_2	0	1
s_2	s_1	s_1	0	0

Slika 8.1 Tabela prelaza [Izvor: Rosen]



Slika 8.2 Dijagram stanja [Izvor: Rosen]

ZADATAK 2.

Odrediti dijagram stanja za konačni automat koji je predstavljen sledećom tabelom

Predviđeno vreme trajanja: 5 minuta

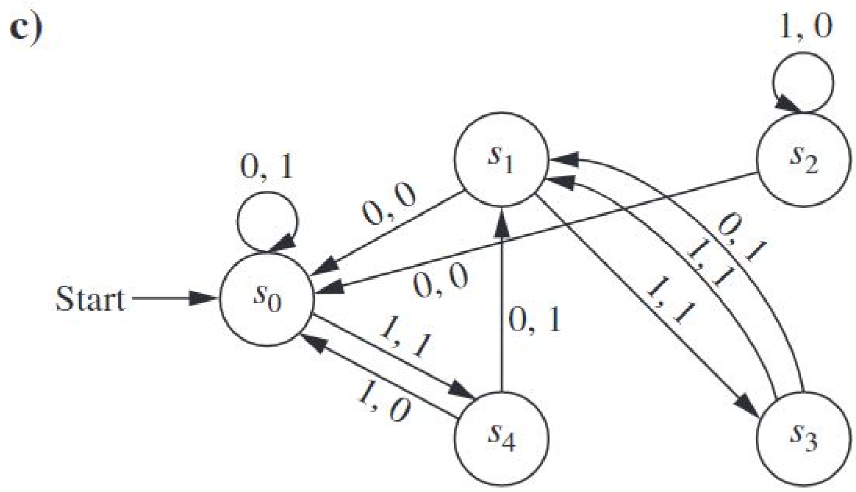
Zadatak

Odrediti dijagram stanja za konačni automat

c)

State	*f* Input 0	*f* Input 1	*g* Input 0	*g* Input 1
s_0	s_0	s_4	1	1
s_1	s_0	s_3	0	1
s_2	s_0	s_2	0	0
s_3	s_1	s_1	1	1
s_4	s_1	s_0	1	0

Slika 8.3 Tabela prelaza [Izvor: Rosen]

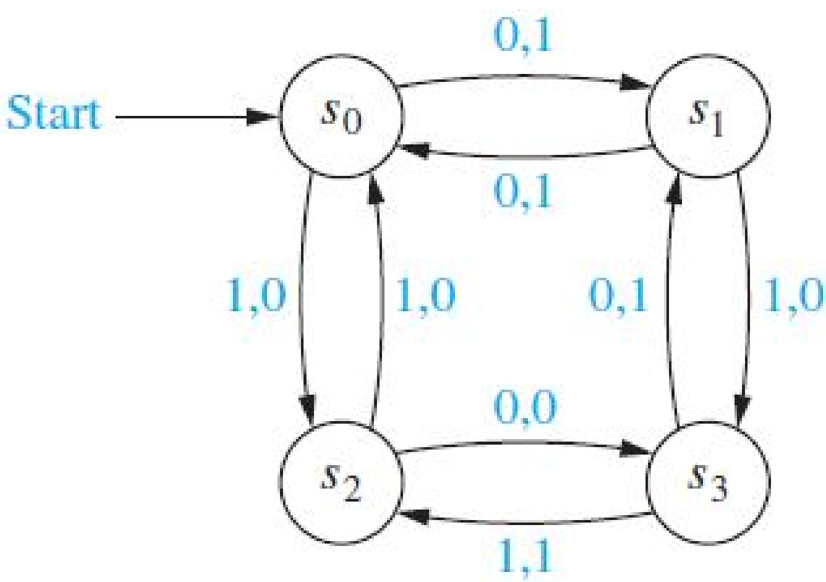


Slika 8.4 Dijagram stanja [Izvor: Rosen]

ZADATAK 3.

Odredite tablicu stanja za mašinu konačnih stanja opisanu sledećom slikom

Predviđeno vreme trajanja: 5 minuta



Slika 8.5 Mašina konačnih stanja [Izvor: Rosen]

STANJE	f		g	
	Ulaz		Ulaz	
	0	1	0	1
S_0	S_1	S_2	1	0
S_1	S_0	S_3	1	0
S_2	S_3	S_0	0	0
S_3	S_1	S_2	1	1

Slika 8.6 Rešenje [Izvor: Rosen]

ZADATAK 4.

Odrediti izlazni niz generisan od strane mašine konačnih stanja iz zadatka 3 ako je ulazni niz 101011

Predviđeno vreme trajanja: 5 minuta

Zadatak

Odrediti izlazni niz generisan od strane mašine konačnih stanja iz zadatka 3 ako je ulazni niz 101011.

Rešenje

Ulaz	**1**	**0**	**1**	**0**	**1**	**1**
Stanje	S_0	S_2	S_3	S_2	S_3	S_2
Izlaz	0	0	1	0	1	0

Slika 8.7 Rešenje [Izvor: Autor]

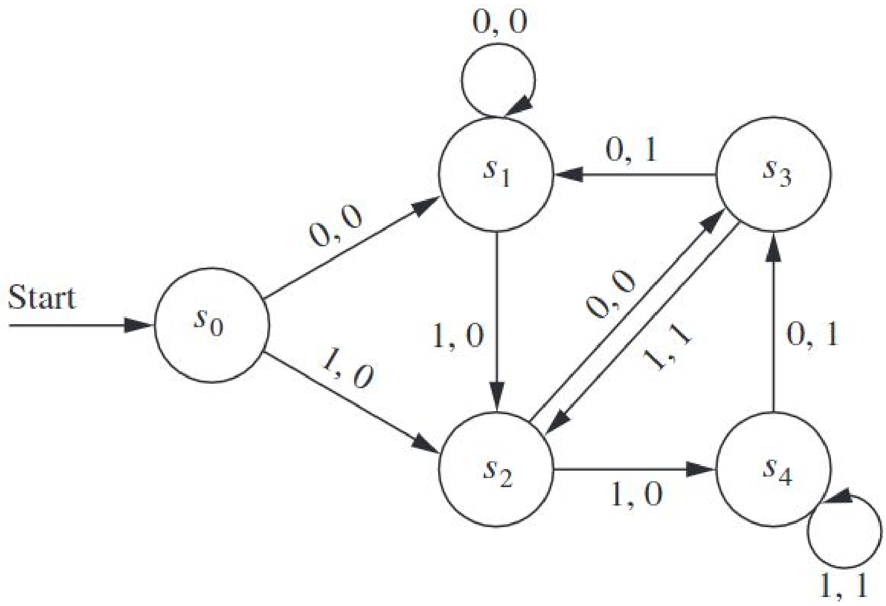
ZADATAK 5.

Konstruisati mašinu konačnih stanja koja ispunjava uslove zadatka

Predviđeno vreme trajanja: 5 minuta

Zadatak:

Konstruisati mašinu konačnih stanja koja "odlaze" ulaz za dva bita, 00, koja su prva dva bita izlaza.



Slika 8.8 Rešenje [Izvor: Rosen]

ZADATAK 6.

Konstruišemo konačni automat koji prepoznaje skup nizova bitova koji se završavaju sa 01

Predviđeno vreme trajanja: 5 minuta

Konstruisati konačni automat koji prepoznaje skup nizova bitova koji se završavaju sa 01

Rešenje:

(a) Naš cilj je da konstruišemo deterministički konačni automat koji prepoznaje skup nizova koji se završavaju sa 01.

Posmatrajmo problem "unazad". Da bismo završili u stanju s2, do njega treba doci iz stanja s1 i to preko bita 1. (1 je poslednji bit).

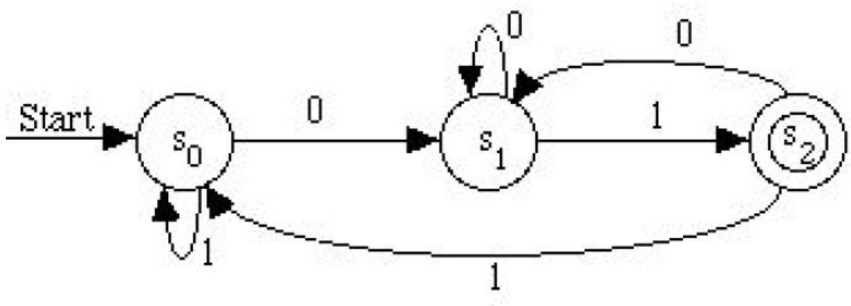
U stanje s1 mozemo doci preko bita 0 iz s0.

Time smo pokrili slucaj niza sa tacno dva elementa - 01.

Iz s0 po bitu 1 ostajemo u stanju S0 (primer 11101)

Iz s1 po bitu 0 ostajemo u stanju s1. (primer 000001)

Ukoliko sekvenca 01 nije na kraju niza, s2 ne mora da bude konacno izlazno stanje, po bitu 0 se treba vratiti u s1 (primer 0101), a po bitu 1 se vracamo na pocetno stanje s0 (primer 01101).



Slika 8.9 Rešenje [Izvor: Rosen]

Poglavlje 9

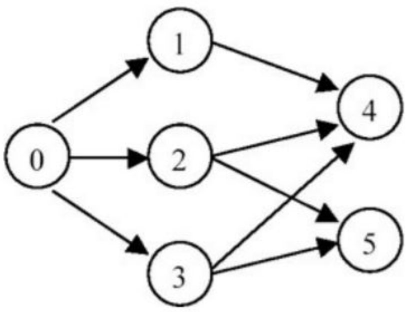
Zadaci za samostalni rad

ZADACI

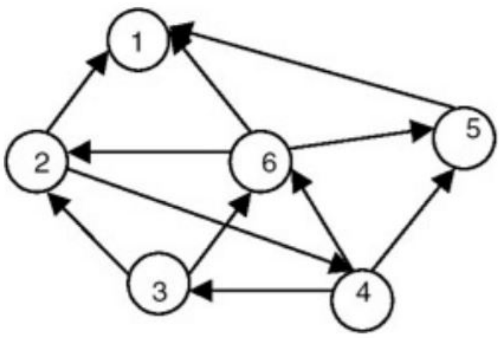
Zadaci za provežbavanje

Zadatak 1 - predviđeno vreme trajanja 15 minuta

Za grafove na slikama odrediti topološko sortiranje gde se uvek među dostupnim čvorovima bira onaj sa najmanjim brojem.



Slika 9.1 Zadatak 1a. [Izvor: Autor]



Slika 9.2 Zadatak 1b. [Izvor: Autor]

Zadatak 2 - predviđeno vreme trajanja 15 minuta

Konstruisati konačni automat koji prepoznaje skup nizova bitova koji se završavaju sa 10

Zadatak 3 - predviđeno vreme trajanja 15 minuta

Odrediti deterministički konačni automat koji prepoznaje skup niza bitova koji sadrže neparan broj 0 i koji se završavaju sa barem tri uzastopne 1.

ZAKLJUČAK

U ovoj lekciji su obrađeni su parcijalno i linearno uređeni skupovi. Specijalni fokus je dat na leksikografskom uređenju, tšopološkom sortiranju, kao i Hasovim dijagramima. U okviru Hasovih dijagrama su naglašeni njegovi izomorfni grafovi, kao i njegovi ekstremni elementi. Takođe, obrađene su i mreže i aksioma transfinitne indukcije.

Literatura

[1] Rosen, Kenneth H. "Discrete mathematics and its applications." AMC 10 (2007): 12.
[2] Epp, Susanna S. Discrete mathematics with applications. Cengage Learning, 2010.