Podsjetimo se…

- zadnji puta:
  - bezmemorijski sustavi
  - eksplicitni i implicitni sustavi
  - spojna lista
  - opisivanje memorijskih sustava
  - definicija konačnih automata

1

a danas ……

- danas ćemo razmotriti:
  - nedeterminističke automate
  - ekvivalenciju automata
  - kaskadu automata
  - povratnu vezu automata

2

Automati

- automat se definira uređenom petorkom

$$Automat = (Stanja, Ulazi, Izlazi, FunkcijaPrijelaza, pocetnoStanje)$$

značenje ovih imena je

Stanja - *prostor stanja*

Ulazi - ulazni skup znakova ili *ulazni alfabet*

Izlazi – izlazni skup znakova ili *izlazni alfabet*

$pocetnoStanje \in Stanja$ je pocetno stanje

$$FunkcijaPrijelaza : Stanja \times Ulazi \to Stanja \times Izlazi$$

3

Automati

- $Ulazi$ i $Izlazi$ su skupovi mogućih ulaznih odnosno izlaznih znakova
- skup $UlazniSignali$ sastoji se od svih beskonačnih nizova ulaznih znakova

$$UlazniSignali = [Prirodni_0 \to Ulazi]$$

- skup $IzlazniSignali$ sastoji se od svih beskonačnih nizova izlaznih znakova

$$IzlazniSignali = [Prirodni_0 \to Izlazi]$$

4

Automati

- neka je ulazni signal

$$u \in UlazniSignali$$

- pojedini znak u signalu može se označiti

$$u(n), \quad \forall n \in Prirodni_0$$

n ovdje nužno ne predstavlja trenutak u vremenu već korak (poziciju) u nizu

- cijeli ulazni signal je niz

$$(u(0), u(1), u(2), ..., u(n), ...)$$

5

Automati

- automat opisan petorkom

$$Automat = (Stanja, Ulazi, Izlazi, FunkcijaPrijelaza, pocetnoStanje)$$

definira funkciju

$$F : UlazniSignali \to IzlazniSignali$$

dakle

$$\forall u \in UlazniSignali \quad \Rightarrow \quad y = F(u)$$

6

Automati

- ali i definira postupak za izračunavanje ove funkcije za određeni ulazni signal
- niz stanja u pojedinim koracima, *odziv stanja*, $(x(0), x(1), \ldots.)$ i izlazni signal y se konstruiraju, korak po korak, kako slijedi

$$x(0) = pocetnoStanje$$

i

$$\forall n \geq 0, \quad \left(x(n+1), y(n)\right) = FunkcijaPrijelaza\left(x(n), u(n)\right)$$

- svaki gornji izračun se naziva *reakcija* ili *odziv*

Automati

- pogodno je funkciju *FunkcijaPrijelaza* razložiti u dvije funkcije

  funkciju narednog stanja

  $$narednoStanje : Stanja \times Ulazi \rightarrow Stanja$$

  i *izlaznu funkciju*

  $$izlaz : Stanja \times Ulazi \rightarrow Izlazi$$

  pa pišemo $x(n+1) = narednoStanje\left(x(n), u(n)\right)$

  odnosno $y(n) = izlaz\left(x(n), u(n)\right)$

Automati

- zaključno

$$\forall x(n) \in Stanja \wedge \forall u(n) \in Ulazi,$$

$$\left(x(n+1), y(n)\right) = FunkcijaPrijelaza\left(x(n), u(n)\right)$$

$$= \left(narednoStanje\left(x(n), u(n)\right), izlaz\left(x(n), u(n)\right)\right)$$

Automati

- definira se specijalni, “ne čini ništa”, ulazni znak koji nazivamo *odsutan* (*absent*)
- pri čemu je ovaj znak uvijek mogući ulaz i izlaz pa je

$$odsutan \in Ulazi \qquad odsutan \in Izlazi$$

- kada je $u(n)=odsutan$
  - tada se stanje ne mijenja pa je $x(n+1)=x(n)$
  - tada je izlaz također odsutan tj. $y(n)=odsutan$

10

Automati - primjer

- primjer: igra bacanja novčića u kojoj se postiže *pobjeda* nakon tri uzastopne *glave* a gubi se (*poraz*) dobivanjem *pisma* prije tri glave u nizu
- da bi se definiralo automat koji u potpunosti opisuje ovu igru treba definirati svih pet elemenata petorke

$$(Stanja, Ulazi, Izlazi, FunkcijaPrijelaza, pocetnoStanje)$$

11

Primjer: definiranje automata

$$Stanja = \{0\_glava, 1\_glava, 2\_glave\}$$

$$Ulazi = \{glava, pismo\}$$

$$Izlazi = \{pobjeda, poraz, odsutan\}$$

$$pocetnoStanje = \{0\_glava\}$$

funkcija *FunkcijaPrijelaza* može se definirati tablicom

12

Primjer - FunkcijaPrijelaza

$(x(n+1), y(n)) = FunkcijaPrijelaza(x(n), u(n))$

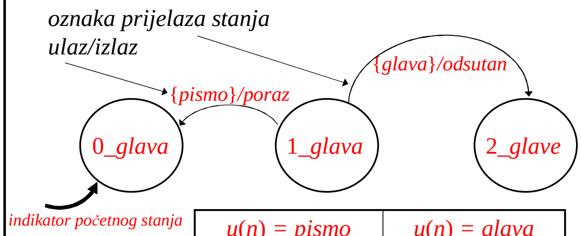
	$u(n)$=glava	$u(n)$=pismo
$x(n)$=0_glava	(1_glava,odsutan)	(0_glava,poraz)
$x(n)$=1_glava	(2_glave,odsutan)	(0_glava,poraz)
$x(n)$=2_glave	(0_glava,pobjeda)	(0_glava,poraz)

Dijagram prijelaza stanja

- da bi se kreirao dijagram prijelaza stanja, kraće dijagram stanja, za automat, prvo se ucrtaju krugovi koji predstavljaju moguća stanja

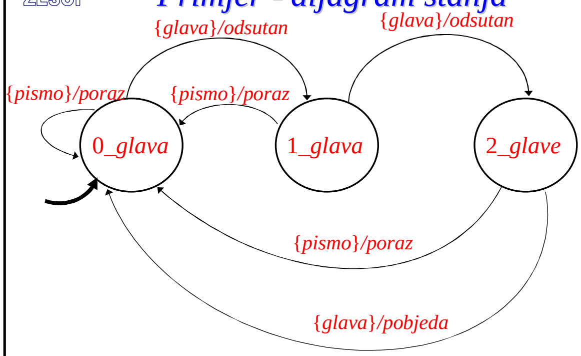
Primjer - dijagram stanja

- za svaku kombinaciju ulaza i stanja nacrtati strelicu od trenutnog stanja u naredno stanje



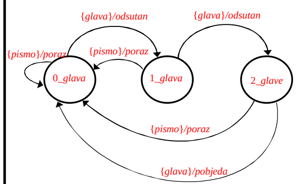
	$u(n)$ = pismo	$u(n)$ = glava
$x(n)$=1_glava	(0_glava,poraz)	(2_glave,odsutan

Primjer - dijagram stanja



16

Primjer - dijagram stanja



stanja
0_glava - postignuto nula glava
1_glava – postignuta jedna glava
2_glave – postignuto dvije glave

ulazi
pismo – rezultat bacanja pismo
glava – rezultat bacanja glava

izlazi
poraz – nisu postignute tri glave u nizu
pobjeda – postignute tri glave u nizu
odsutan – ni pobjeda ni poraz

17

Automati - činjenice

- automati o kojima smo do sada govorili nazivaju se Mealyevi automati i za njih je karakteristično da izlazni znak ovisi o ulaznom znaku i znaku stanja
- definiraju se i Mooreovi automati kod kojih je izlaz samo funkcija trenutnog stanja i neovisan je vanjskom ulazu

18

Još o Automatima

- svaki prijelaz i izlaz ovise samo o trenutnom stanju i trenutnom ulazu
- prethodni ulazni znakovi utječu na prijelaz i izlaz samo u onoj mjeri u kojoj određuju trenutno stanje
- prijelaz će biti određen za svaku moguću kombinaciju ulaza i trenutnih stanja
- ako prijelaz nije prikazan za pojedini ulaz, pretpostavlja se da je prijelaz u isto stanje i da je izlaz *odsutan*

19

Još o Automatima

- ako više od jednog ulaznog znaka vodi na isti prijelaz i izlaz oznaka prijelaza “ulaz/izlaz” može sadržavati oznaku skupa ulaznih znakova
- ako za neki automat vrijedi da postoji točno jedan mogući prijelaz za svaku kombinaciju trenutnog stanja i ulaza tada govorimo o determinističkom automatu

20

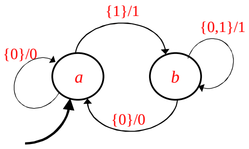
Još o Automatima

- automat je *receptivan* ako postoji barem jedan prijelaz za svaku kombinaciju trenutnog stanja i ulaza
- nedeterministički automat može imati više od jednog mogućeg prijelaza za svaku kombinaciju trenutnog stanja i ulaza

21

Nedeterministički automati

- dakle nedeterministički automati se razlikuju od determinističkih po činjenici da dopuštaju više nego jedan prijelaz za dano trenutno stanje i ulaz



  - kada je stanje $x(n)=b$ i ako je $u(n)=0$ naredno stanje $x(n+1)$ može biti ili a ili b
  - izlaz $y(n)$ može biti ili 0 ili 1
- model ne kazuje kako je izbor prijelaza načinjen

Nedeterministički automati

- slično determinističkim i nedeterministički automati se prikazuju petorkom

$$(Stanja, Ulazi, Izlazi, mogucaFunkcijaPrijelaza, pocetnoStanje)$$

- razlika je u definiciji prijelazne funkcije koja je ovdje nazvana *mogucaFunkcijaPrijelaza*
- *mogucaFunkcijaPrijelaza* definira se kako slijedi

Nedeterministički automati

- za dani ulaz $u(n)$ i trenutno stanje $x(n)$ *mogucaFunkcijaPrijelaza* namiče skup mogućih narednih stanja $x(n+1)$ i izlaza $y(n)$

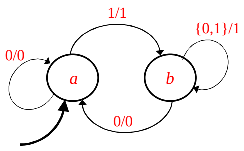
$$mogucaFunkcijaPrijelaza: Stanja \times Ulazi \rightarrow P(Stanja \times Izlazi)$$

- pri čemu je $P(Stanja \times Izlazi)$ partitivni skup od $(Stanja \times Izlazi)$ dakle skup svih podskupova od spomenutog skupa

Nedeterministički automati

- prema tome, svaki podskup od $(Stanja \times Izlazi)$ je element od $P(Stanja \times Izlazi)$ pa je područje vrijednosti funkcije *mogucaFunkcijaPrijelaza* je skup uređenih parova iz $P(Stanja \times Izlazi)$

Nedeterministički automati

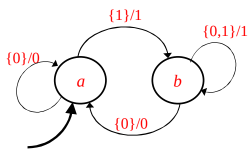


$Stanja = \{a,b\}$
$Ulazi = \{0,1,odsutan\}$
$Izlazi = \{0,1,odsutan\}$
$pocetnoStanje = a$

$(x(n+1), y(n)) = mogucaFunkcijaPrijelaza(x(n), u(n))$

	$u(n)=0$	$u(n)=1$
$x(n) = a$	$\{(a,0)\}$	$\{(b,1)\}$
$x(n) = b$	$\{(b,1),(a,0)\}$	$\{(b,1)\}$

Nedeterministički Automati



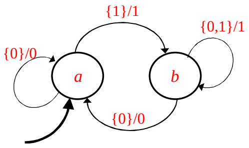
za ovaj automat postoji, za isti ulazni signal, više nizova mogućih stanja i izlaza:

ulazni niz znakova (0,1,0,1,0,1,……)

stanja sustava (*a*,*a*,*b*,*a*,*b*,*a*,*b*,…….)

izlazni niz znakova (0,1,0,1,0,1,……)

Nedeterministički Automati



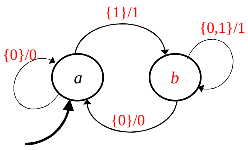
ali i slijedeće mogućnosti

ulazni niz znakova (0,1,0,1,0,1,……)

stanja sustava (*a,a,b,b,b,b,b,*…….)

izlazni niz znakova (0,1,1,1,1,1,……)

Nedeterministički Automati



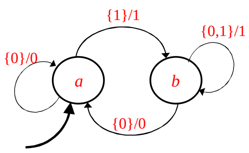
te treća mogućnost

ulazni niz znakova (0,1,0,1,0,1,……)

stanja sustava (*a,a,b,b,b,a,b,*…….)

izlazni niz znakova (0,1,1,1,0,1,……)

Nedeterministički Automati



i četvrta mogućnost

ulazni niz znakova (0,1,0,1,0,1,……)

stanja sustava (*a,a,b,a,b,b,b,*…….)

izlazni niz znakova (0,1,0,1,1,1,……)

Konačni automati – primjer parkirnog sata

- nedeterministički automati, se često koriste pri simuliranju automata složene strukture s automatom jednostavnije strukture
- razmotrimo ovdje primjer 60 minutnog parkirnog sata čiji ćemo rad opisati uz pomoć jednog konačnog automata

31

Konačni automati – primjer parkirnog sata

- tri ulazna znaka *kov*5, *kov*25 i *otkucaj*
  - *kov*5 – ubacivanje kovanice u vrijednosti 5 minuta parkiranja
  - *kov*25 – ubacivanje kovanice u vrijednosti 25 minuta parkiranja
  - *otkucaj* – protek jedne minute parkirnog sata
- sat pokazuje preostalo vrijeme prije "*istek*"-a 60 minuta

32

Konačni automati – primjer parkirnog sata

- kada se pojavi ulazni znak *kov*5, vrijeme se uveća za 5 minuta (do maksimalno 60)
- kada se pojavi ulazni znak *kov*25, vrijeme se uveća za 25 minuta (do maksimalno 60)
- kada se pojavi ulazni znak *otkucaj* vrijeme se umanjuje za 1 minutu (do minimuma od 0 minuta)
- kada preostalo vrijeme postane jednako 0 parkirni sat postavlja poruku vrijeme *istek*lo

33

Konačni automati – primjer parkirnog sata

- definiramo konačni automat parkirnog sata

$Stanja = \{0, 1, 2, ..., 60\}$

$Ulazi = \{kov5, kov25, otkucaj, odsutan\}$

$Izlazi = \{istek, 1, 2, ..., 60, odsutan\}$

$pocetnoStanje = 0$

$FunkcijaPrijelaza : Stanja \times Ulazi \to Stanja \times Izlazi$

gdje je funkcija prijelaza pobliže definirana kako slijedi

34

Konačni automati – primjer parkirnog sata

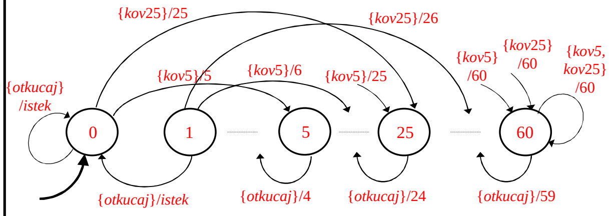
$\forall x(n) \in Stanja, \quad u(n) \in Ulazi$

$FunkcijaPrijelaza(x(n), u(n)) =$

$$= \begin{cases} (0, istek) & u(n) = otkucaj \wedge (x(n) = 0 \vee x(n) = 1) \\ (x(n)-1, x(n)-1) & u(n) = otkucaj \wedge x(n) > 1 \\ (\min(x(n)+5, 60), \min(x(n)+5, 60)) & u(n) = kov5 \\ (\min(x(n)+25, 60), \min(x(n)+25, 60)) & u(n) = kov25 \\ (x(n), odsutan) & u(n) = odsutan \end{cases}$$

35

Deterministički model parkirnog sata



- primjer: neka je dan ulazni niz

$(kov25, otkucaj^{18}, kov5, otkucaj^{10}, otkucaj^{4}, ...)$

izlazni niz je

$(istek, 25, 24, ..., 8, 7, 12, 11, 10, ..., 3, 2, 1, istek, ...)$

36

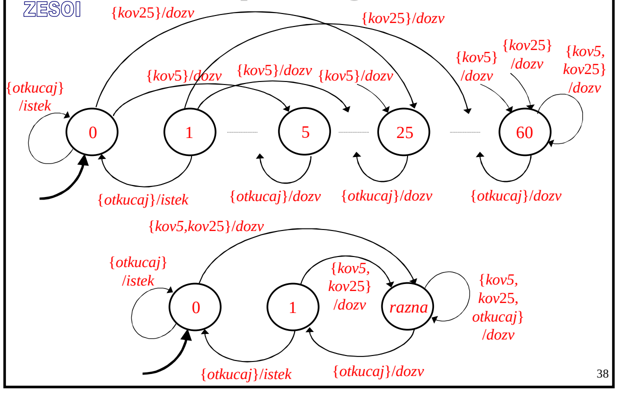
Konačni automati – primjer parkirnog sata

- definiramo konačni automat parkirnog sata sa stajališta nadziratelja parkiranja
- za nadziratelja parkiranja nezanimljiv je podatak o preostalom vremenu jer njega zanima samo je li parkiranje unutar *dozv*oljenog vremena ili je ono *istek*lo
- sukladno tom pristupu redefiniramo model na način da su sada *Izlazi*

$$Izlazi = \{dozv, istek, odsutan\}$$

37

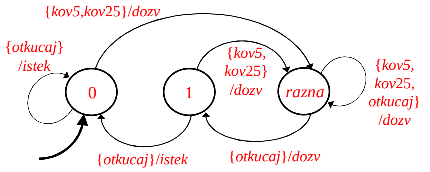
Nedeterministički model parkirnog sata



38

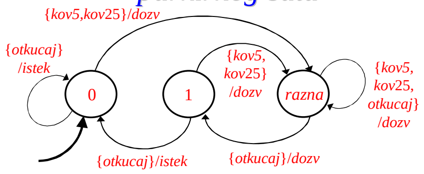
Nedeterministički model parkirnog sata

- nedeterministički model parkirnog sata sadrži manje detalja i predstavlja apstrakciju izvornog determinističkog modela



39

Nedeterministički model parkirnog sata



- ovdje su, međutim, mogući odzivi koji su u slučaju determinističkog modela nemogući

ulazni niz znakova	(*kov5*,	*otkucaj*,	*otkucaj*,	*otkucaj*,	…)	
stanja sustava	(*razna*,	*razna*,	1,	0,	0,	…)
izlazni niz znakova	(*dozv*,	*dozv*,	*istek*,	*istek*,	…)	

- a sada vi objasnite policajcu da je ovo samo nedeterministički model stvarnog parkirnog sata

40

Vladanja (ponašanja) automata

- *Vladanja* automata opisujemo parom (u,y) gdje je u ulazni niz a y odgovarajući izlazni niz
- definiramo

$$Vladanja = \{(u,y) \in [Prirodni_0 \to Ulazi] \times [Prirodni_0 \to Izlazi] \mid y \text{ je mogući izlazni niz za ulaz } u \}$$

41

Vladanja (ponašanja) automata

- za determinističke automate postoji samo jedan izlazni niz y za svaki ulazni niz u
- *Vladanja* automata je tada graf funkcije tj. svaki element domene $[Prirodni_0 \to Ulazi]$ se preslikava u jedan element u $[Prirodni_0 \to Izlazi]$

42

Vladanja (ponašanja) automata

- za nedeterminističke automate za jedan ulazni niz iz [$Prirodni_0 \rightarrow Ulazi$] postoji više mogućih izlaznih nizova u [$Prirodni_0 \rightarrow Izlazi$]
- *Vladanja* automata tada nije funkcija već relacija

43

Ekvivalencija automata

- dva različita automata mogu biti ekvivalentna na način da za isti ulazni niz generiraju isti izlazni niz
- definiraju se dvije relacije ekvivalencije
  - simulacija
  - bisimulacija

44

Ekvivalencija automata

- kažemo da automat A simulira automat B ako za bilo koji ulazni niz svaki izlazni niz automata B je također mogući izlazni niz automata A
- kažemo da A bisimulira B ako A simulira B i B simulira A

45

Relacije simulacije automata

- simulacijske relacije povezuju skupove dvaju automata
- one su skup uređenih parova koje uparuju stanje automata A s “ekvivalentnim” stanjem automata B

46

Relacije simulacije automata

- formalno kažemo da A simulira B ako postoji simulacijska relacija $S \subset Stanja_B \times Stanja_A$ takva da

1. $(pocetStanje_B, pocetStanje_A) \in S$, i
2. $\forall u(n) \in Ulazi, \quad \forall (x_B(n), x_A(n)) \in X$,

   i $\forall (x_B(n+1), y_B(n)) \in mogucaFunkcijaPrijelaza(x_B(n), u(n))$,

   $\exists (x_A(n+1), y_A(n)) \in mogucaFunkcijaPrijelaza(x_A(n), u(n))$

   takav da

   $(x_B(n+1), x_A(n+1)) \in X \quad$ i $\quad y_A(n) = y_B(n)$

47

Relacije simulacije automata

- izravno kazano A simulira B kada postoji takav skup parova stanja da bilo koji ulaz $u(n)$ prevodi oba automata iz ekvivalentnih stanja $(x_B(n), x_A(n))$ u ekvivalentna stanja $(x_B(n+1), x_A(n+1))$ generirajući pri tome isti izlaz $y_B(n) = y_A(n)$

48

Relacije simulacije automata

- ako A simulira B tada A ima sva *Vladanja* koja ima i B a možda i više

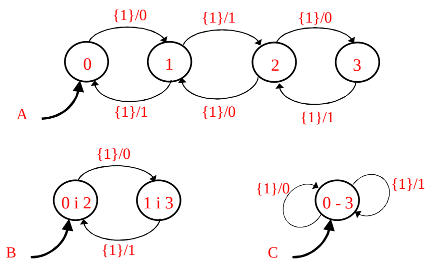
A simulira B $\Rightarrow$ $Vladanja_B \subset Vladanja_A$

- ako A simulira B tada svako *Vladanje* koje je nemoguće za A je također nemoguće za B dakle

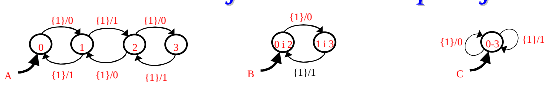
$$(u, y) \notin Vladanja_A \Rightarrow (u, y) \notin Vladanja_B$$

Simulacije automata - primjer

- dana su tri automata A, B, C

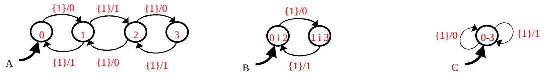


Simulacije automata - primjer



- C simulira A i B
- B simulira A ali i A simulira B
- B može pratiti svaku promjenu stanja (vladanje) A ali i A, koji je nedeterministički u dva stanja, može, na dva načina, pratiti svaku promjenu stanja B
- dakle simulacijske relacije nisu jednoznačne

Simulacije automata - primjer



- ako automat A iz stanja 1 uvijek izabere povratak u stanje 0 relacija simulacije je

$$S_{B,A} = \{(0 \text{ i } 2,0),(1 \text{ i } 3,1)\}$$

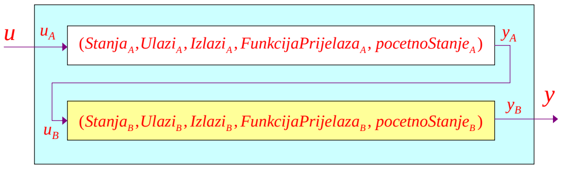
- ako automat A iz stanja 2 uvijek izabere povratak u stanje 1 relacija simulacije je

$$S_{B,A} = \{(0 \text{ i } 2,0),(1 \text{ i } 3,1),(0 \text{ i } 2,2)\}$$

- inače

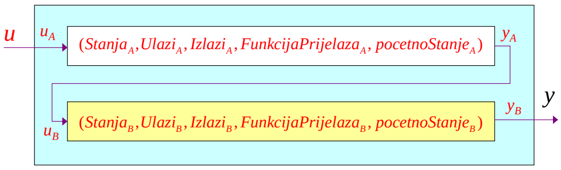
$$S_{B,A} = \{(0 \text{ i } 2,0),(1 \text{ i } 3,1),(0 \text{ i } 2,2),(1 \text{ i } 3,3)\}$$

Kaskada automata



- oba automata djeluju istovremeno (oba u koraku n)
- oba automata imaju svoja vlastita stanja, ulaze i izlaze
- izlaz automata A je ulaz u automat B
- djelovanje ulaza $u_A(n)$ propagira istovremeno kroz kaskadu za svaki korak - sinkronost

Kaskada automata - definicija



- definira se 5-torka za složeni automat (kaskadu)

Stanja= *Ulazi=*

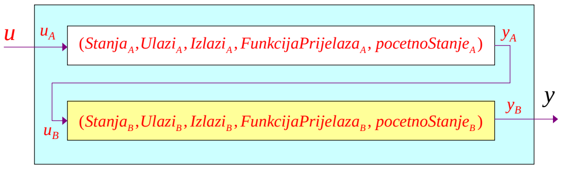
pocetnoStanje= *Izlazi=*

$$FunkcijaPrijelaza((x_A(n), x_B(n), u(n)) = ((x_A(n+1), x_B(n+1), y(n))$$

gdje $((x_A(n+1), y_A(n)) = FunkcijaPrijelaza_A((x_A(n), u(n))$

i $((x_B(n+1), y(n)) = FunkcijaPrijelaza_B((x_B(n), y_A(n))$

Kaskada automata - definicija



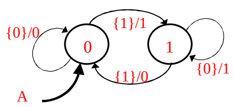
- važno je uočiti da "interni" izlaz $y_A(n)$ se koristi kao "interni" ulaz $u_B(n)$ u automat B
- prema tome da bi kaskadni spoj bio valjan mora biti

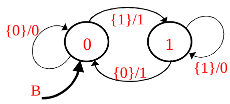
$$Izlazi_A \subset Ulazi_B$$

55

Kaskada automata - primjer

- neka su zadana dva automata A i B i spojimo ih u kaskadu tako da je izlaz iz automata A ulaz u automat B





56

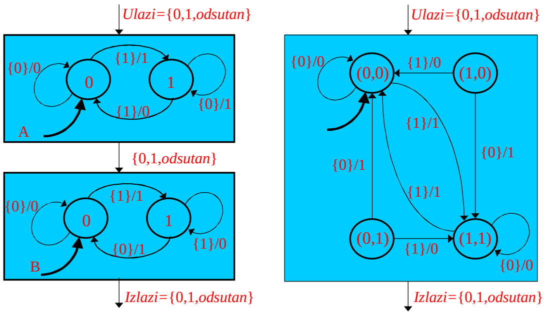
Kaskada automata - primjer

- za dva automata spojena u kaskadu može se nacrtati jedinstveni dijagram stanja:

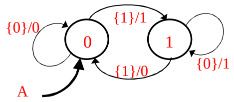
1. nacrtaj krugove za svako stanje u $Stanja_A \times Stanja_B$
2. za svako stanje razmotri svaki mogući ulaz u A
   a) odredi odgovarajuće naredno stanje automata A
   b) odredi izlaz automata A koji tvori ulaz u automat B
   c) odredi odgovarajuće naredno stanje automata B
   d) odredi izlaz automata B
   e) ucrtaj prijelaznu strelicu u $(x_A(n+1), x_B(n+1))$
   f) označi prijelaznu strelicu s ulazom u automat A i izlazom iz automata B

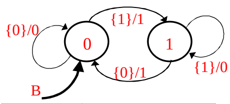
57

Kaskada automata - primjer



Kaskada automata





Automat A	(*narednoStanje*$_A$, *izlaz*$_A$) *za ulaz*$_A$		
trenutnoStanje$_A$	0	1	*odsutan*
0	(0,0)	(1,1)	(0, *odsutan*)
1	(1,1)	(0,0)	(1, *odsutan*)

Automat B	(*narednoStanje*$_B$, *izlaz*$_B$) *za ulaz*$_B$		
trenutnoStanje$_B$	0	1	*odsutan*
0	(0,0)	(1,1)	(0, *odsutan*)
1	(0,1)	(1,0)	(1, *odsutan*)

	(*narednoStanje*, *izlaz*) *za ulaz*		
trenutnoStanje	0	1	*odsutan*
(0,0)	((0,0),0)	((1,1),1)	((0,0),*odsutan*)
(0,1)	((0,0),1)	((1,1),0)	((0,1),*odsutan*)
(1,0)	((1,1),1)	((0,0),0)	((1,0),*odsutan*)
(1,1)	((1,1),0)	((0,0),1)	((1,1),*odsutan*)

Kaskada automata - primjer

- iz prethodne tablice ili iz dijagrama stanja je vidljivo da stanja (0,1) i (1,0) nisu upravljiva ili dostupna iz početnog stanja
- stanje se naziva neupravljivim ili nedostupnim ako se nekim nizom ulaznih znakova početno stanje ne može prevesti u to stanje

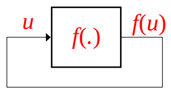
Povratna veza

- elementarni spoj automata u povratnoj vezi je spoj u kojem je izlaz iz automata ujedno i ulaz u isti automat
- složenija je situacija spoja više automata koji mogu biti spojeni u petlje povratnih veza
- razmatramo slaganje sinkronih modela automata u povratnu vezu
- kod sinkronih automata izlazni znak je istodoban s ulaznim znakom pa će izlazni znak automata u povratnoj vezi ovisiti o ulaznom znaku koji opet ovisi o svom vlastitom izlaznom znaku

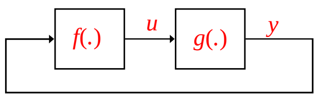
61

Povratna veza

- problem je sličan onom kod bezmemorijskih sustava
- primjer $u = f(u)$ za dani f



- primjer $u = f(y)$ i $y = g(u)$ za dane f i g



62

Povratna veza

- razmotrimo tri slučaja primjera $u = f(u)$ za dane f
- za $f : Realni \rightarrow Realni$ slijedi uz $\forall u \in Realni,$
- slučaj 1. – jedinstveno rješenje

$$f(u) = 1 - u \Rightarrow u = 1 - u \Rightarrow u = 0{,}5$$

- slučaj 2. – nema realnog rješenja

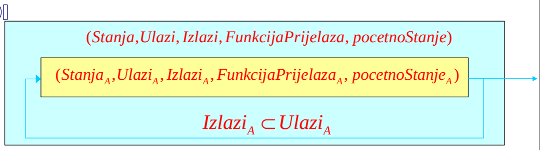
$$f(u) = 1 + u^2 \Rightarrow u = 1 + u^2 \Rightarrow u = 0{,}5 \pm j0{,}5\sqrt{3}$$

- slučaj 3. – više rješenja

$$f(u) = u^2 \Rightarrow u = u^2 \Rightarrow u_1 = 0, \quad u_2 = 1$$

63

Povratna veza – automati bez ulaza



- razmatramo dakle spoj u kojem je izlaz automata A spojen je na njegov ulaz
- želimo odrediti složeni automat označen petorkom *(Stanja, Ulazi, Izlazi, FunkcijaPrijelaza, pocetnoStanje)* svjetloplavim blokom – automat bez ulaza
- on se ne uklapa u naš model automata koji pretpostavlja postojanje ulaza na koje automat djeluje (reagira)

64

Povratna veza – automati bez ulaza

- zato se uvodi nadomjesni ulazni znak, *djeluj* pa je ulazni alfabet

$$Ulazi = \{djeluj, odsutan\}$$

- ulazni znak *djeluj* interpretiramo kao nalog unutarnjem automatu za djelovanje

65

Povratna veza – automati bez ulaza

- problem je naći $y(n)$ koji je ujedno i ulazni znak

za $x(n) \in Stanja_A$, i $y(n) \in Izlazi_A$

$$(x(n+1), y(n)) = FunkcijaPrijelaza_A(x(n), y(n))$$

66

Povratna veza – automati bez ulaza

- pogodno je, i ovdje funkciju *FunkcijaPrijelaza* razložiti u dvije funkcije

funkciju narednog stanja

$$narednoStanje : Stanja_A \times Ulazi_A \to Stanja_A$$

i *izlaznu funkciju*

$$izlaz_A : Stanja_A \times Ulazi_A \to Izlazi_A$$

pa pišemo $x(n+1) = narednoStanje(x(n), y(n))$

odnosno $y(n) = izlaz_A(x(n), y(n))$

67

Povratna veza – automati bez ulaza

- u jednadžbi $y(n) = izlaz_A(x(n), y(n))$

$x(n)$ je konstanta i u slučaju *dobro-formiranog* automata daje jedinstveno rješenje

68

Povratna veza – automati bez ulaza

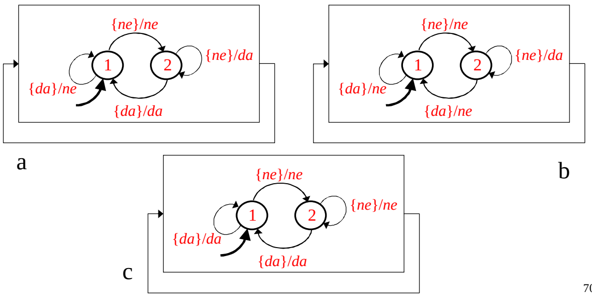
- za *dobro-formirani* automat vrijedi

$$Stanja = Stanja_A$$
$$Ulazi = \{djeluj, odsutan\}$$
$$Izlazi = Izlazi_A$$
$$pocetnoStanje = pocetnoStanje_A$$
$$FunkcijaPrijelaza(x(n), y(n)) = \begin{cases} FunkcijaPrijelaza_A(x(n), y(n)), & \\ \quad \text{gdje je } y(n) \text{ jedinstveno rješenje ako } u(n) = djeluj & \\ (x(n), y(n)) \quad \text{ako } u(n) = odsutan & \end{cases}$$

69

Povratna veza – automati bez ulaza primjeri

- tri primjera slaganja automata u povratnu vezu

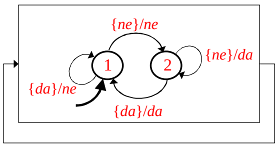


70

Povratna veza – automati bez ulaza (primjer a)

- u sva tri primjera ulazi i izlazi osnovnih automata koje spajamo u povratnu vezu su

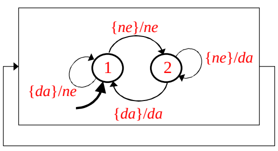
$$Ulazi_A = Izlazi_A = \{da, ne, odsutan\}$$



- za početno stanje 1 postoje dvije odlazne grane i za oba ulazna signala je $y(n)=ne$

71

Povratna veza – automati bez ulaza (primjer a)



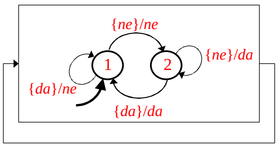
- budući je izlazni znak *ne* tada je ulazni znak isto *ne* i to je jedinstveno rješenje od

$$izlaz_A(1, ne) = ne$$

- prijelaz stanja je iz stanja 1 u stanje 2
- za tako dobiveno stanje 2 opet postoje dvije odlazeće grane i obje generiraju izlazni znak $y(n)=da$ za moguće ulazne znakove

72

Povratna veza – automati bez ulaza (primjer a)



- budući je izlazni znak *da* tada je ulazni znak isto *da* i to je jedinstveno rješenje od

$$izlaz_A(2, da) = da$$

- prijelaz stanja je iz stanja 2 u stanje 1
- kako za oba dostupna stanja postoje jedinstvena rješenja izlaznih jednadžbi govorimo o *dobro-formiranom* automatu s povratnom vezom

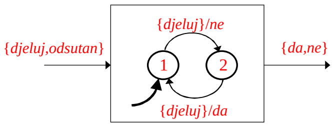
Povratna veza – automati bez ulaza (primjer a)

- automat s povratnom vezom mijenja svoja stanja na svaki znak *djeluj* i generira izlazni niz

$$(ne, da, ne, da, ne, da, ne, ...)$$

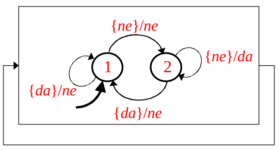
za ulazni niz znakova $(djeluj, djeluj, djeluj, ...)$

- dobiveni automat je



Povratna veza – automati bez ulaza (primjer b)

- za početno stanje 1 postoje dvije odlazne grane i za oba ulazna signala je *y*(*n*)=*ne*



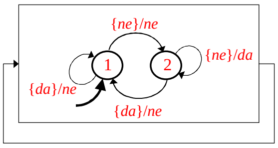
- budući je izlazni znak *ne* tada je ulazni znak isto *ne* i to je jedinstveno rješenje od

$$izlaz_A(1, ne) = ne$$

- prijelaz stanja je iz stanja 1 u stanje 2

Povratna veza – automati bez ulaza (primjer b)

- za tako dobiveno stanje 2 opet postoje dvije odlazeće grane i ako pretpostavimo ulazni znak *da* izlazni znak je *ne*; no ako je ulazni znak *ne* izlazni znak će biti *da*



- to pokazuje da ne postoji rješenje izlazne jednadžbe

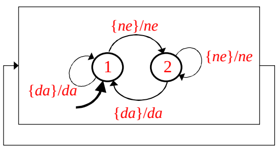
$$izlaz_A(2, y(n)) = y(n)$$

- govorimo o *loše-formiranom* automatu s povratnom vezom

76

Povratna veza – automati bez ulaza (primjer c)

- ako za početno stanje 1 pretpostavimo ulazni znak *da* izlazni znak je isto *da*, a za ulazni znak *ne* i izlazni znak je *ne*



- to pokazuje da postoje dva rješenje izlazne jednadžbe

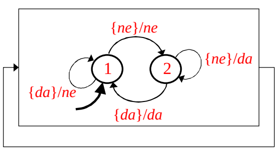
$$izlaz_A(1, y(n)) = y(n)$$

- i ovdje govorimo o *loše-formiranom* automatu s povratnom vezom

77

Povratna veza – automati bez ulaza (primjeri a,b i c - zaključak)

- zaključujemo da automati u primjerima b i c ne mogu biti spojeni u povratnu vezu kako je to prikazano
- jedina mogućnost ovako konstruirane povratne veze je za automat u primjeru a



78