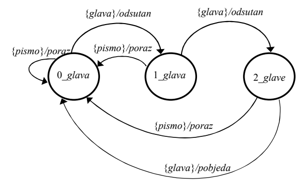
Podsjetimo se…

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

a danas ……

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

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

Još o Automatima

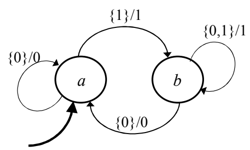
- ako više od jednog ulaznog znaka vodi na isti prijelaz stanja i isti izlaz, oznaka prijelaza "ulaz/izlaz" može sadržavati oznaku skupa ulaznih znakova
- ako za neki automat vrijedi da postoji točno jedan mogući prijelaz za svaku kombinaciju trenutnog stanja i ulaza tada govorimo o determinističkom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

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

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 $s(n)=b$ i ako je $x(n)=0$ naredno stanje $s(n+1)$ 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 $x(n)$ i trenutno stanje $s(n)$ *mogucaFunkcijaPrijelaza* namiče skup mogućih narednih stanja $s(n+1)$ i izlaza $y(n)$

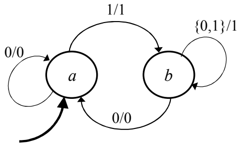
$$mogucaFunkcijaPrijelaza : Stanja \times Ulazi \to P(Stanja \times Izlazi)$$

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

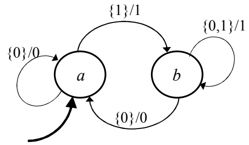


Stanja = {*a*,*b*}
Ulazi = {0,1,*odsutan*}
Izlazi = {0,1,*odsutan*}
pocetnoStanje = *a*

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

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

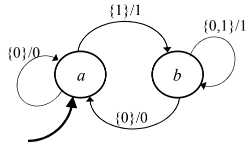
Nedeterministički Automati



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

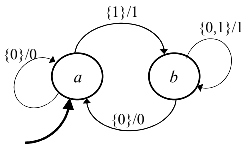
ulazni niz znakova (0,1,0,1,0,1,……)
stanja sustava (*a,a,b,a,b,a,b,*…….)
izlazni niz znakova (0,1,0,1,0,1,……)

Nedeterministički Automati



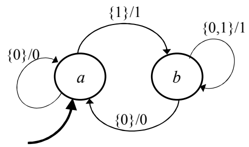
ali i slijedeće mogućnosti
ulazni niz znakova (0,1,0,1,0,1,……)
stanja sustava (*a,a,b,b,b,b,b,*…….)
izlazni niz znakova (0,1,1,1,1,1,……)

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

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

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

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

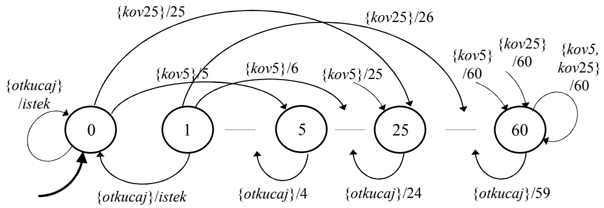
Konačni automati – primjer parkirnog sata

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

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

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

Deterministički model parkirnog sata



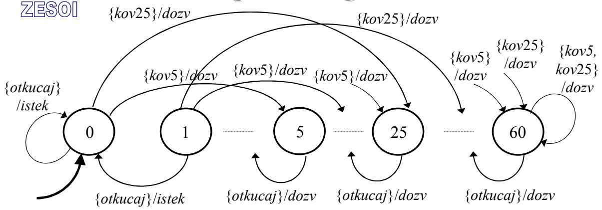
- primjer: neka je dan ulazni niz ($kov25$, $otkucaj^{18}$, $kov5$, $otkucaj^{10}$, $otkucaj^{4}$,….) izlazni niz je (*istek*, 25, 24, ..., 8, 7, 12, 11, 10,…., 3, 2, 1, *istek*, ...)

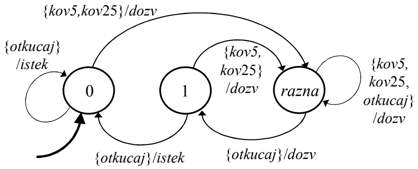
Konačni automati – primjer parkirnog sata

- 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*

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

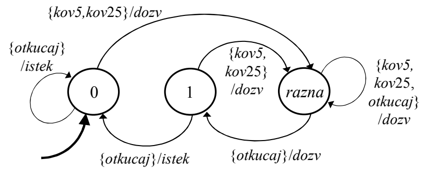
Nedeterministički model parkirnog sata



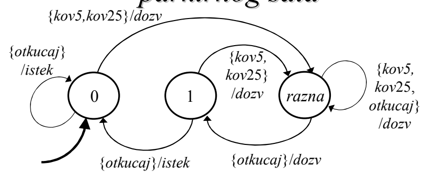


Nedeterministički model parkirnog sata

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



Nedeterministički model parkirnog sata



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

ulazni niz znakova	(*kov*5,*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

Vladanja (ponašanja) automata

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

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

Vladanja (ponašanja) automata

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

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

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

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

Relacije simulacije automata

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

Relacije simulacije automata

- formalno kažemo da A simulira B ako postoji simulacijska relacija

  takva da

1. $(pocetStanje_B, pocetStanje_A) \in S$, i
2. $\forall x(n) \in Ulazi, \quad \forall (s_B(n), s_A(n)) \in S$,

   i $\forall (s_B(n+1), y_B(n)) \in mogucaFunkcijaPrijelaza(s_B(n), x(n))$,

   $\exists (s_A(n+1), y_A(n)) \in mogucaFunkcijaPrijelaza(s_A(n), x(n))$

   takav da

   $(s_B(n+1), s_A(n+1)) \in S$ i $y_A(n) = y_B(n)$

Relacije simulacije automata

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

Relacije simulacije automata

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

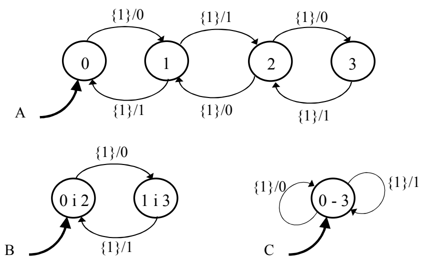
A simulira B $\Rightarrow Vladanja_B \subset Vladanja_A$

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

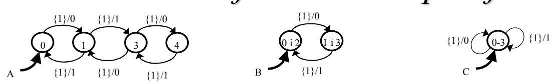
$$(x, y) \neq Vladanja_A \Rightarrow (x, y) \neq Vladanja_B$$

Simulacije automata - primjer

- dana su tri automata A, B, C

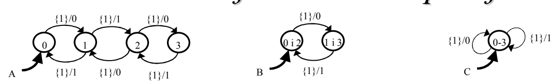


Simulacije automata - primjer



- 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

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

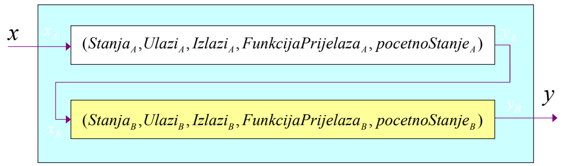
- ako automat A iz stanja 2 uvijek izabere povratak u stanje 1 relacija simulacije je

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

- inače

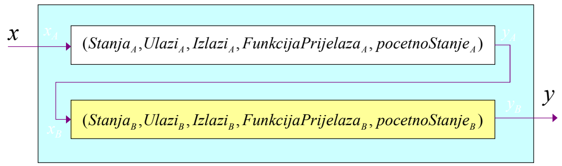
$$S_{B,A} = \{(0 \text{ i } 2, 0), (1 \text{ i } 3, 1), (0 \text{ i } 2, 2), (1 \text{ i } 3, 3)\}$$

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 $x_A(n)$ propagira istovremeno kroz kaskadu za svaki korak - sinkronost

Kaskada automata - definicija



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

Stanja= *Ulazi*=

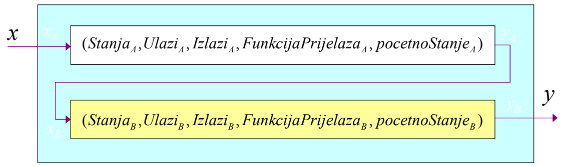
pocetnoStanje= *Izlazi*=

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

gdje $\left((s_A(n+1), y_A(n)\right) = FunkcijaPrijelaza_A\left((s_A(n), x(n)\right)$

i $\left((s_B(n+1), y(n)\right) = FunkcijaPrijelaza_B\left((s_B(n), y_A(n)\right)$

Kaskada automata - definicija

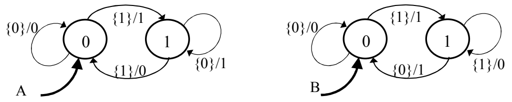


- važno je uočiti da "interni" izlaz $y_A(n)$ se koristi kao "interni" ulaz $x_B(n)$ u automat B
- prema tome da bi kaskadni spoj bio valjan mora biti

$$Izlazi_A \subset Ulazi_B$$

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

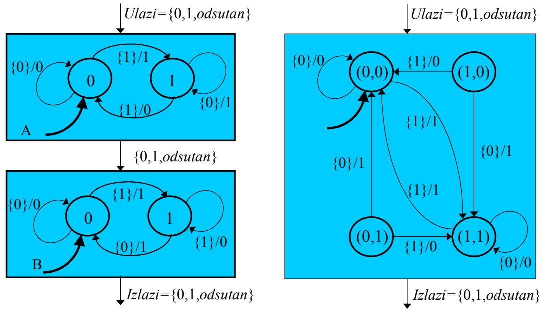


Kaskada automata - primjer

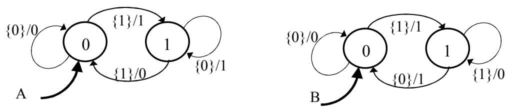
- za dva automata spojena u kaskadu može se nacrtati jedinstveni dijagram stanja:

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 $(s_A(n+1), s_B(n+1))$
   f) označi prijelaznu strelicu s ulazom u automat A i izlazom iz automata B

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

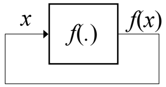
- iz prethodne tablice ili iz dijagrama stanja je vidljivo da stanja (0,1) i (1,0) nisu upravljiva ili dostupna iz početnog stanja
- stanje se naziva neupravljivim ili nedostupnim ako se nekim nizom ulaznih znakova početno stanje ne može prevesti u to stanje

Povratna veza

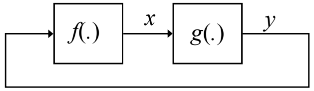
- elementarni spoj automata u povratnoj vezi je spoj u kojem je izlaz iz automata ujedno i ulaz u isti automat
- složenija je situacija spoja više automata koji mogu biti spojeni u petlje povratnih veza
- razmatramo slaganje sinkronih modela automata u povratnu vezu
- kod sinkronih automata izlazni znak 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

Povratna veza

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



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



Povratna veza

- razmotrimo tri slučaja primjera $x = f(x)$ za dane f
- za f : *Realni* → *Realni* slijedi uz
- slučaj 1. – jedinstveno rješenje

$$f(x) = 1 - x \Rightarrow x = 1 - x \Rightarrow x = 0,5$$

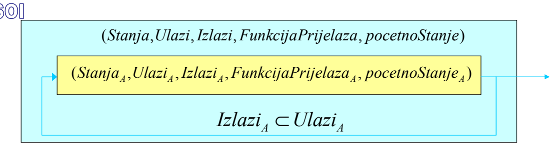
- slučaj 2. – nema realnog rješenja

$$\sqrt{\ }$$

- slučaj 3. – više rješenja

$$f(x) = x^2 \Rightarrow x = x^2 \Rightarrow x_1 = 0, \quad x_2 = 1$$

Povratna veza – automati bez ulaza

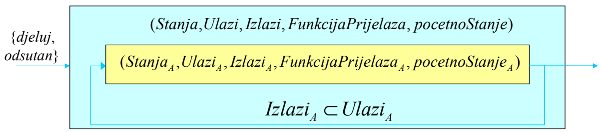


- razmatramo dakle spoj u kojem je izlaz automata A spojen je na njegov ulaz
- želimo odrediti složeni automat označen petorkom (*Stanja*, *Ulazi*, *Izlazi*, *FunkcijaPrijelaza*, *pocetnoStanje*) svjetloplavim blokom – automat bez ulaza
- on se ne uklapa u naš model automata koji pretpostavlja postojanje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

Povratna veza – automati bez ulaza

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

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



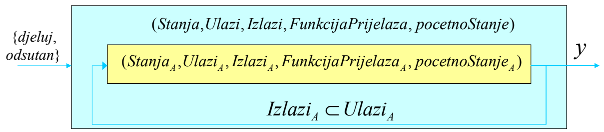
- ulazni znak *djeluj* interpretiramo kao nalog unutarnjem automatu za djelovanje

Povratna veza – automati bez ulaza

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

$$\text{za} \quad s(n) \in Stanja_A, \quad \text{i} \quad y(n) \in Izlazi_A$$

$$\big(s(n+1), y(n)\big) = FunkcijaPrijelaza_A\big(s(n), y(n)\big)$$



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 $s(n+1) = narednoStanje\big(s(n), y(n)\big)$

odnosno $y(n) = izlaz_A\big(s(n), y(n)\big)$

Povratna veza – automati bez ulaza

- u jednadžbi $y(n) = izlaz_A\big(s(n), y(n)\big)$

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

Povratna veza – automati bez ulaza

- za *dobro-formirani* automat vrijedi

$$Stanja = Stanja_A$$

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

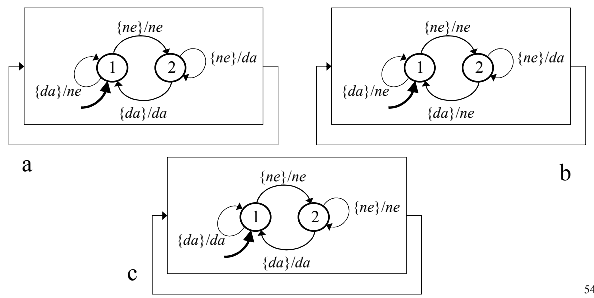
$$Izlazi = Izlazi_A$$

$$pocetnoStanje = pocetnoStanje_A$$

$$FunkcijaPrijelaza\big(s(n), y(n)\big) = \begin{cases} FunkcijaPrijelaza_A\big(s(n), y(n)\big), \\ \qquad \text{gdje je } y(n) \text{ jedinstveno rješenje ako } x(n) = djeluj \\ \big(s(n), y(n)\big) \quad \text{ako } x(n) = odsutan \end{cases}$$

Povratna veza – automati bez ulaza primjeri

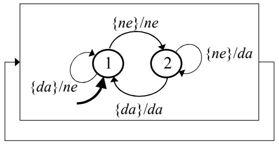
- tri primjera slaganja automata u povratnu vezu



Povratna veza – automati bez ulaza (primjer a)

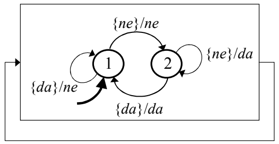
- u sva tri primjera ulazi i izlazi osnovnih automata koje spajamo u povratnu vezu su

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



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

Povratna veza – automati bez ulaza (primjer a)

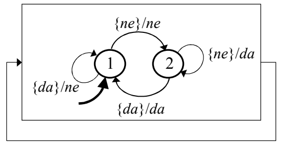


- budući je izlazni znak *ne* tada je ulazni znak isto *ne* i to je jedinstveno rješenje od

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

- 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

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

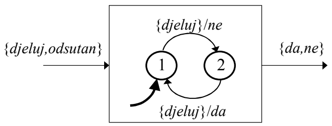
Povratna veza – automati bez ulaza (primjer a)

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

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

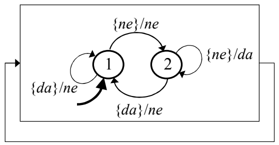
za ulazni niz znakova

- 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*



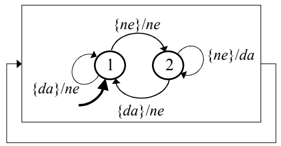
- budući je izlazni znak *ne* tada je ulazni znak isto *ne* i to je jedinstveno rješenje od

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

- 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*



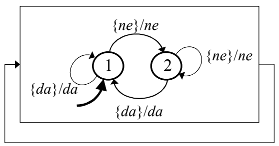
- to pokazuje da ne postoji rješenje izlazne jednadžbe

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

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

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*



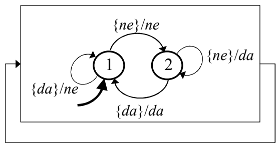
- to pokazuje da postoje dva rješenje izlazne jednadžbe

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

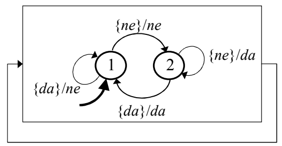
- i ovdje govorimo o *loše-formiranom* automatu s povratnom vezom

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



Povratna veza – izlaz određen stanjem



- ovo je ujedno i primjer automata za koji vrijedi da je izlaz određen stanjem jer je jednak za oba moguća ulazna znaka
- dakle *y*(*n*)=*ne* za *s*(*n*)=1 i *y*(*n*)=*da* za *s*(*n*)=2

Povratna veza – izlaz određen stanjem

- kažemo da automat A ima izlaz određen stanjem ako za svako dostupno stanje

$$s(n) \in Stanja_A$$

postoji jedinstveni izlazni znak $y(n)=b$ koji ovisi samo o $s(n)$ a ne ovisi o ulaznom znaku

- dakle

$$izlaz_A\left(s(n), x(n)\right) = b$$

- očigledno je da je automat s povratnom vezom dobro-formiran

Povratna veza – izlaz određen stanjem

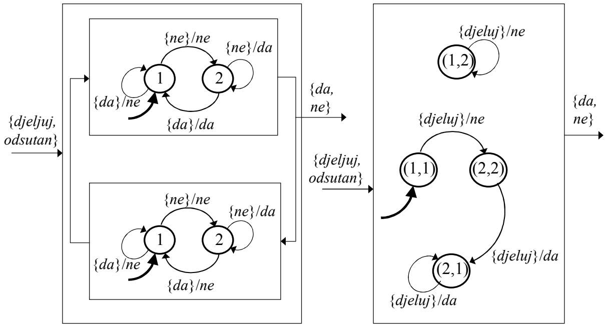
- za ovaj specijalni slučaj automat opisujemo

$$Stanja = Stanja_A$$
$$Ulazi = \{djeluj, odsutan\}$$
$$Izlazi = Izlazi_A$$
$$pocetnoStanje = pocetnoStanje_A$$
$$FunkcijaPrijelaza\left(s(n), x(n)\right) = \begin{cases} FunkcijaPrijelaza_A\left(s(n), b\right), \\ \quad \text{gdje je } b \text{ jedinstveni izlazni znak u stanju } s(n) \text{ ako } x(n) = djeluj \\ \left(s(n), y(n)\right) \quad \text{ako } \ x(n) = odsutan \end{cases}$$

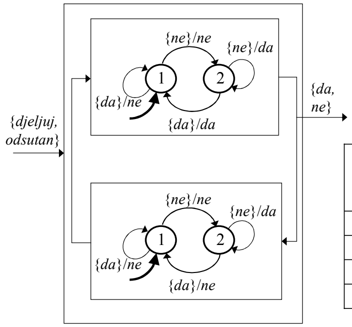
Povratna veza – izlaz određen stanjem

- ako se automat s izlazom određenim stanjem kombinira s bilo kojim drugim automatom u spoj s povratnom vezom rezultirajući spoj će biti *dobro-formiran*
- primjer: kombinacija automata iz prethodnih primjera a i b
- automat A ima izlaz određen stanjem a automat B ne
- ukupna kombinacija je *dobro-formirana*

Povratna veza – izlaz određen stanjem

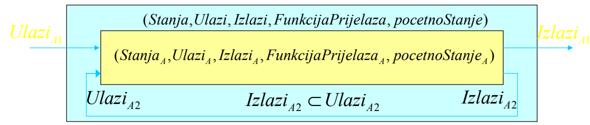


Povratna veza – izlaz određen stanjem



trenutno stanje	(naredno stanje, izlaz) za ulaz	
	djeluj	odsutan
(1,1)	((2,2),ne)	((1,1),odsutan)
(2,2)	((2,1),da)	((2,2),odsutan)
(1,2)	((1,2),ne)	((1,2),odsutan)
(2,1)	((2,1),da)	((2,1),odsutan)

Povratna veza – automati s ulazom



- razmatramo dakle automat s dva ulaza i dva izlaza u spoju s povratnom vezom pri čemu je drugi izlaz spojen na drugi ulaz
- želimo definirati složeni automat označen petorkom (*Stanja*, *Ulazi*, *Izlazi*, *FunkcijaPrijelaza*, *pocetnoStanje*) svjetloplavim blokom

Povratna veza – automati s ulazom

- ulazi i izlazi automata A su oblika

$$Ulazi_A = Ulazi_{A1} \times Ulazi_{A2}$$
$$Izlazi_A = Izlazi_{A1} \times Izlazi_{A2}$$

- izlazna funkcija od A je

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

- odnosno

$$izlaz_A = (izlaz_{A1}, izlaz_{A2})$$

Povratna veza – automati s ulazom

- gdje

$$izlaz_{A1} : Stanja_A \times Ulazi_A \rightarrow Izlazi_{A1}$$

daje izlazni znak na prvom izlazu a

$$izlaz_{A2} : Stanja_A \times Ulazi_A \rightarrow Izlazi_{A2}$$

na drugom

Povratna veza – automati s ulazom

- neka su za automat A u n-tom koraku $s(n) \in Stanja_A$ i trenutni vanjski ulazni znak $x_1(n) \in Ulazi_{A1}$
- naš problem je odrediti "nepoznati" izlazni znak

$(y_1(n), y_2(n)) \in Izlazi_A$ tako da vrijedi

$$izlaz_A(s(n), (x_1(n), y_2(n))) = (y_1(n), y_2(n))$$

- znak $y_2(n)$ se pojavljuje na obje strane jer je drugi ulaz $x_2(n)$ u automat jednak $y_2(n)$

Povratna veza – automati s ulazom

- izlaznu jednadžbu možemo pisati

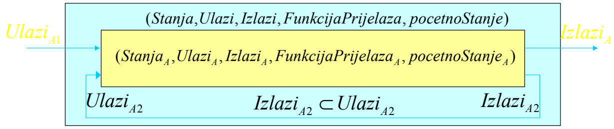
$$izlaz_{A1}(s(n),(x_1(n),y_2(n))) = y_1(n)$$

$$izlaz_{A2}(s(n),(x_1(n),y_2(n))) = y_2(n)$$

u ovim jednadžbama $s(n)$ i $x_1(n)$ su poznati a $y_1(n)$ i $y_2(n)$ su nepoznati

- druga jednadžba ukazuje da će jedinstveno rješenje biti moguće samo za *dobro-formirane* automate

Povratna veza – automati s ulazom



- kažemo da će automat s povratnom vezom biti dobro-formiran ako za svako dostupno stanje

$s(n) \in Stanja_A$ i za svaki vanjski znak $x_1(n) \in Ulazi_{A1}$

postoji jedinstveni izlazni simbol koji zadovoljava jednadžbu

$$izlaz_{A2}(s(n),(x_1(n),y_2(n))) = y_2(n)$$

Povratna veza – automati s ulazom

- za *dobro-formirani* automat vrijedi

$Stanja = Stanja_A$

$Ulazi = Ulazi_{A1}$

$Izlazi = Izlazi_{A1}$

$pocetnoStanje = pocetnoStanje_A$

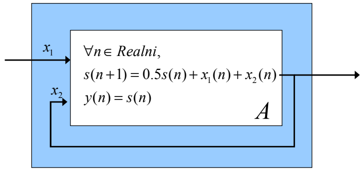
$FunkcijaPrijelaza\left(s(n),x(n)\right) = \left(narednoStanje\left(s(n),x(n)\right), izlaz\left(s(n),x(n)\right)\right)$:

$narednoStanje\left(s(n),x(n)\right) = narednoStanje_A\left(s(n),(x(n),y_2(n))\right)$ i

$izlaz\left(s(n),x(n)\right) = izlaz_A\left(s(n),(x(n),y_2(n))\right)$ gdje je $y_2(n)$ jedinstveno rješenje jednadžbe $izlaz_{A2}(s(n),(x(n),y_2(n))) = y_2(n)$

Povratna veza – automati s ulazom

- primjer



- A ima dva ulaza i jedan izlaz

$Ulazi_A = Realni \times Realni, \quad Izlazi_A = Realni$

i stanja $Stanja_A = Realni$

Povratna veza – automati s ulazom

- prema tome A ima beskonačni ulazni i izlazni alfabet te beskonačno mnogo stanja
- u n-tom koraku označavamo par ulaznih vrijednosti s $(x_1(n), x_2(n))$, trenutno stanje sa $s(n)$, naredno stanja sa $s(n+1)$ i izlaz s $y(n)$
- funkcija prijelaza je tada

$$\left(s(n+1), y(n)\right) = FunkcijaPrijelaza\left(s(n),(x_1(n),x_2(n))\right)$$
$$= \left(0.5s(n) + x_1(n) + x_2(n), s(n)\right)$$

Povratna veza – automati s ulazom

- ekvivalentno pišemo

$$\left(s(n+1), y(n)\right) = narednoStanje_A\left(s(n),(x_1(n),x_2(n))\right)$$
$$= 0.5s(n) + x_1(n) + x_2(n)$$

$$y(n) = izlaz_A\left(s(n),(x_1(n),x_2(n))\right) = s(n)$$

- očigledno je da ovaj automat ima izlaz određen stanjem

Povratna veza – automati s ulazom

- povratna veza povezuje izlaz i drugi ulaz, $x_2(n) = y(n)$ pa je
- iz čega slijedi
- kako je $x_1(n) = x(n)$
- potpuni je opis automata s povratnom vezom

$Ulazi = Realni, \quad Izlazi = Realni, \quad Stanja = Realni$

$FunkcijaPrijelaza\left(s(n),x(n)\right) =$

$$= \left(0.5s(n) + x(n) + s(n), s(n)\right) = \left(1.5s(n) + x(n), s(n)\right)$$

Povratna veza – automati s ulazom

- finalno