Podsjetimo se…

- primjeri slike
- primjeri video signala
- nizovi simbola
- vremenski diskretni signali i otipkavanje
- kvantizacija po amplitudi i vremenu
- funkcije i kompozicija funkcija
- sustavi kao funkcije
- primjer sustava s povratnom vezom
- opis sustava pomoću blok dijagrama
- složeni sustavi

uh!!!! dobro je prošlo

- danas ćemo razmotriti:
  - bezmemorijske sustave
  - eksplicitne i implicitne sustave
  - spojnu listu
  - opisivanje memorijskih sustava
  - definiciju i spajanje konačnih automata

Bezmemorijski sustavi

- bezmemorijski sustav ima svojstvo da odziv sustava ovisi samo o trenutnoj vrijednosti ulaznog signala a ne o njihovim prethodnim vrijednostima
- sustav

$$F:[Realni \to Y] \to [Realni \to Y]$$

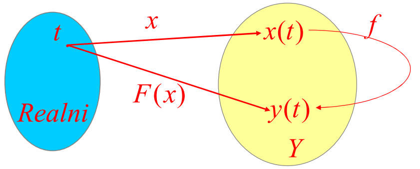
je bezmemorijski ako postoji funkcija

$$f: Y \to Y$$

tako da vrijedi

$$\forall t \in Realni \wedge \forall x \in [Realni \to Y], (F(x))(t) = f(x(t))$$

Bezmemorijski sustavi



$$y = F(x) \Rightarrow \quad y(t) = (F(x))(t) = f(x(t))$$

- bezmemorijski sustav ima svojstvo da izlaz u trenutku *t* ovisi samo o vrijednosti ulaznog signala u trenutku *t*.

Bezmemorijski sustavi

- razmotrimo sustav

$$Kvadrat:[Realni \to Realni] \to [Realni \to Realni]$$

gdje ako je

$$y = Kvadrat(x)$$

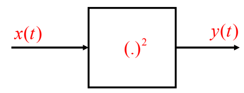
tada

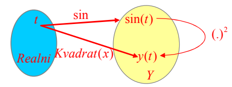
$$\forall t \in Realni,\ y(t) = (x(t))^2$$

- neka je ulazni signal

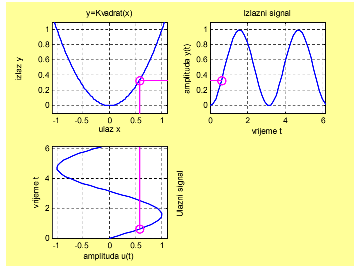
$$\forall t \in Realni,\ x(t) = \sin(t)$$

Bezmemorijski sustavi





$y(t) = (Kvadrat(x))(t) = (\sin(t))^2$



$$y(t) = (\sin(t))^2$$

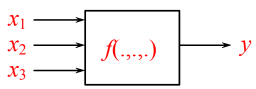
$$= \frac{1}{2}(1 - \cos(2t))$$

Funkcijski blok

- kod bezmemorijskih sustava izlaz u trenutku *t* ovisi samo o vrijednosti ulaznog signala u trenutku *t*.
- elementi sustava su prikazani funkcijskim blokom.
- funkcijski blok je opisan funkcijom.

$$y(t) = f(x_1(t), x_2(t), \ldots, x_m(t))$$

$$y(t), x_i(t) \in Realni$$



Funkcijski blok s jednim ulazom i jednim izlazom

- $y(t) = f(x(t))$, za svaki t
- f je funkcija koja broju pridružuje broj.
- blok označen s f nazivamo funkcijski blok.
- funkcijska veza ulaza i izlaza može se dati:
  - analitičkim izrazom pomoću poznatih funkcija
  - krivuljom u x-y ravnini
  - tablicom diskretnih vrijednosti

MATLAB

Primjeri funkcijskih blokova s jednim ulazom i jednim izlazom

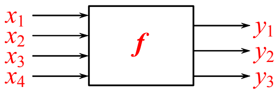
Spajanje funkcijskih blokova u sustav

- sustav s više ulaza i više izlaza:

$y_1 = f_1(x_1, x_2, x_3, x_4)$

$y_2 = f_2(x_1, x_2, x_3, x_4)$

$y_3 = f_3(x_1, x_2, x_3, x_4)$



uvođenjem vektora:

ulaz: $[x_1, x_2, \ldots x_m]^T$

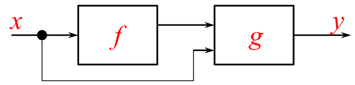
izlaz: $[y_1, y_2, \ldots y_r]^T$

$\mathbf{y} = \boldsymbol{f}(\mathbf{x})$ ... gdje je $\boldsymbol{f}$ vektorska funkcija

Eksplicitni i implicitni sustavi

- dvije grupe sustava bez memorije:
  - eksplicitni sustavi,
  - implicitni sustavi.
- podjela prema tome da li signal na svom putu kroz sustav čini petlju:
  - eksplicitni sustav – nema petlje,
  - implicitni sustav – ima jednu ili više petlji.

Prikaz sustava listom spajanja



g: f, x

f: x

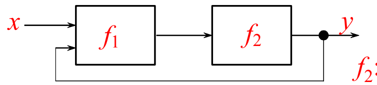
- svaki funkcijski blok ima jedan redak u listi.
- izlazne varijable označene su oznakom funkcijskog bloka.
- ulazne varijable označene su:
  - ulazima u dotični blok,
  - oznakama bloka čiji izlaz ulazi u dotični blok.

Prikaz sustava listom spajanja

- Sortirana lista:
  - kada su redci u listi spajanja složeni tako da u svakom retku ime funkcije ili varijable desno od dvotočke možemo naći lijevo od dvotočke negdje iznad tog retka ili je to ulaz sustava, kažemo da je lista sortirana.

nesortirana lista		sortirana lista
g: f, x	⇒	f: x
f: x		g: f, x

Prikaz sustava listom spajanja



f_2: f_1
f_1: f_2, x

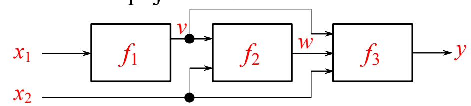
ili

f_1: f_2, x
f_2: f_1

- lista se ne može sortirati ⇒ sustav je implicitan
- implicitni sustav je sustav s povratnom vezom
- spojna lista je način ustanovljavanja da li je sustav eksplicitni ili implicitni u slučaju da to nije moguće ustanoviti vizualnom inspekcijom

Formulacije i rješenje jednadžbi sustava (eksplicitni sustav)

- Ulazno izlazne jednadžbe na osnovu sortirane spojne liste:



f_1: x_1
f_2: f_1, x_2
f_3: f_1, f_2, x_2

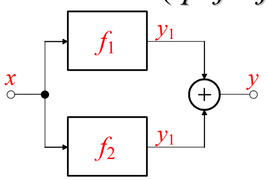
$v = f_1(x_1)$

$w = f_2(v, x_2)$

$y = f_3(v, w, x_2)$

$y = f_3\{f_1(x_1), f_2[f_1(x_1), x_2], x_2\}$

Formulacije i rješenje jednadžbi sustava (spajanje u paralelni slog)



$y = y_1 + y_2$

$y_1 = f_1(x)$

$y_2 = f_2(x)$

$y = f(x) = f_1(x) + f_2(x)$

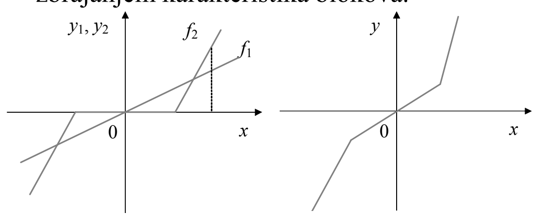
$f = f_1 + f_2$

- paralelni spoj ili slog.
- veći broj sustava složenih paralelno:

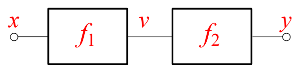
$$y = f(x) = \sum_1^n f_i(x)$$

Formulacije i rješenje jednadžbi sustava (spajanje u paralelni slog)

- karakteristika paralelnog sloga dobiva se zbrajanjem karakteristika blokova.



Formulacije i rješenje jednadžbi sustava (spajanje u kaskadu)



$y = f_2(v)$

$v = f_1(x)$

$y = f_2(f_1(x))$

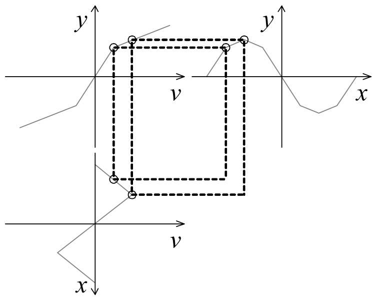
- kaskada sustava.
- funkcija kaskade je kompozicija funkcija:
  - $f = f_2 \circ f_1$
- za kaskadu s većim brojem blokova vrijedi:

$$y = f_n(f_{n-1}(f_{n-2}(\ldots(f_1(x))\ldots))).$$

$$f = f_n \circ f_{n-1} \circ f_{n-2} \circ \ldots \circ f_1.$$

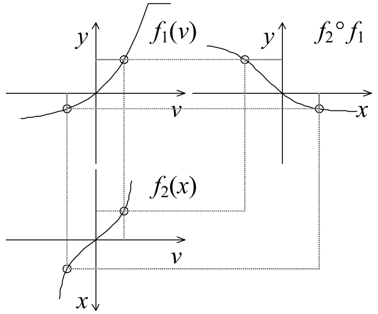
Formulacije i rješenje jednadžbi sustava (spajanje u kaskadu)

- Funkcija kaskade ovisi o redoslijedu blokova.



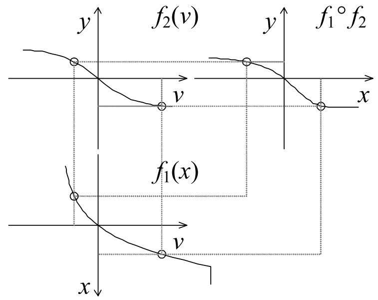
Formulacije i rješenje jednadžbi sustava (spajanje u kaskadu)

- Funkcija kaskade ovisi o redoslijedu blokova.

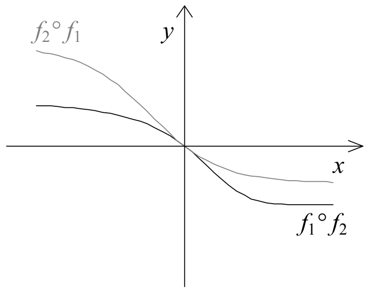


Formulacije i rješenje jednadžbi sustava (spajanje u kaskadu)

- Obrnuti redoslijed kaskada.



Formulacije i rješenje jednadžbi sustava (spajanje u kaskadu)



Bezmemorijski sustavi

- ponovimo:

  bezmemorijski sustav ima svojstvo da odziv sustava ovisi samo o trenutnoj vrijednosti ulaznog signala a ne o njihovim prethodnim vrijednostima i možemo pisati:

$$\forall t \in Realni, \quad y(t) = f(x(t))$$

Memorijski sustavi

- memorijski kauzalni sustav s beskonačnom memorijom definiran je s

$$\forall t \in Realni, \quad y(t) = F(x_{(-\infty,t]})$$

- pri čemu bi trivijalni slučaj $x_{(-\infty,t]} = x(t)$ učinio ovaj sustav bezmemorijskim

Memorijski sustavi u konačnom intervalu

- vladanje sustava uglavnom pratimo na konačnom vremenskom intervalu $(t_0, t]$ koji nazivamo interval promatranja
- zanima nas, dakle, odsječak odziva

$$y_{(t_0,t]}$$

kao posljedica odsječka pobude

$$x_{(t_0,t]}$$

Kontinuirrani memorijski sustavi - primjer

- neka je zadan vremenski kontinuirani sustav s ulazom x i izlazom y gdje je

$$\forall t \in Realni, \quad y(t) = \frac{1}{M} \int_{t-M}^{t} x(\tau) d\tau$$

- uz zamjenu varijabli slijedi

$$y(t) = \frac{1}{M} \int_{0}^{M} x(t-\tau) d\tau$$

- očigledno radi se o memorijskom sustav koji inače ima svojstvo izglađivanja ulaznog signala

Diferencijalne jednadžbe

- sustavi koji su opisani funkcijom

$$KontSustavi : KontSignali \to KontSignali$$

- $KontSignali$ je skup vremenski kontinuiranih signala koji ovisno o konkretnom sustavu mogu biti

$$KontSignali = [Vrijeme \to Realni] \quad \text{ili}$$

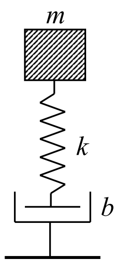
$$KontSignali = [Vrijeme \to Kompleksni]$$

uz $Vrijeme = Realni$ ili $Vrijeme = Realni_+$

Diferencijalne jednadžbe

- ovako definirana klasa sustava naziva se vremenski kontinuirani sustavi
- vrlo često se vremenski kontinuirani sustavi opisuju diferencijalnim jednadžbama

Diferencijalne jednadžbe



- podsjetimo se primjera

$$m \cdot y''(t) + b \cdot y'(t) + k \cdot y(t) = F(t)$$

- uz zamjenu

$$\zeta = \frac{b}{2m} \qquad \omega_0 = \sqrt{\frac{k}{m}} \qquad x(t) = F(t)$$

$$y''(t) + 2\zeta \cdot y'(t) + \omega_0^2 y(t) = x(t)$$

Vremenski diskretni sustavi

- klasa sustava opisanih funkcijom
  $DiskrSustavi: DiskrSignali \to DiskrSignali$
  naziva se diskretni sustavi
- dakle vremenski diskretni sustavi djeluju s klasama diskretnih signala koji mogu biti

$DiskrSignali = [Cjelobrojni \to Realni]$ ili

$DiskrSignali = [Cjelobrojni \to Kompleksni]$ ili

$DiskrSignali = [Prirodni_0 \to Realni]$ ili

$DiskrSignali = [Prirodni_0 \to Kompleksni]$

Jednadžbe diferencija

- vremenski diskretni sustavi često se opisuju uz pomoć jednadžbi diferencija
- razmotrimo vremenski diskretni sustav

$$S : [Prirodni_0 \to Realni] \to [Prirodni_0 \to Realni]$$

gdje je za

$$\forall x \in [Prirodni_0 \to Realni], \quad y = S(x)$$

dan s

$$\forall n \in Cjelobrojni, \quad y(n) = \frac{1}{3}\big(x(n) + x(n-1) + x(n-2)\big)$$

Jednadžbe diferencija

- izlaz za svaki indeks n je srednja vrijednost tri uzlazne vrijednosti (tri uzorka)
- ovo je jednostavni sustav za usrednjavanje i naziva se *moving average*

Jednadžbe diferencija

- neka je $x = u$, jedinični skok, funkcija definirana kao

$$\forall n \in Cjelobrojni, \quad u(n) = \begin{cases} 1 & \text{za} \quad n \geq 0 \\ 0 & \text{za ostale } n \end{cases}$$

- rješavanjem zadane jednadžbe diferencija izračunava se izlaz y

$$\forall n \in Cjelobrojni, \quad y(n) = \begin{cases} 1 & \text{za} \quad n \geq 2 \\ 2/3 & \text{za} \quad n = 1 \\ 1/3 & \text{za} \quad n = 0 \\ 0 & \text{inače} \end{cases}$$

Jednadžbe diferencija

- opći oblik sustava za usrednjavanje tzv. *moving average system* je

$$\forall n \in Cjelobrojni, \quad y(n) = \frac{1}{M}\sum_{j=0}^{M-1} x(n-j)$$

- za svaki n ovaj sustav daje srednju vrijednost od zadnjih M vrijednosti ulaznog signala
- ovaj sustav predstavlja diskretnu realizaciju sustava opisanog s

$$y(t) = \frac{1}{M}\int_0^M x(t-\tau)\,d\tau$$

Deklarativne i imperativne definicije sustava

- dani primjeri su bili primjeri deklarativne definicije sustava
- imperativna definicija predstavlja postupak za izračunavanje izlaznog signala za zadani ulazni signal

Deklarativne i imperativne definicije sustava

- imperativni opis kontinuiranih sustava moguć je uz pomoć analognih el. mreža ili ostalih fizikalnih sustava koji su kontinuirani sami po sebi
- imperativni opis sustava može biti u obliku niza naredbi odgovarajućeg programa u nekom od programskih jezika

Deklarativne i imperativne definicije sustava

- u slučaju kontinuiranih sustava biti će potrebne odgovarajuće aproksimacije koja se koriste pri rješavanju diferencijalnih jednadžbi
- vremenski diskretni sustavi često se definiraju kao *automati* engleski *state machines*

Automati

- do sada korišten model *ulaz - izlaz* koji je definirao izlazni niz za zadani ulazni niz
- uvodi se prikaz sustava temeljen na poznavanju stanja sustava i ulaznog niza i naziva se model s varijablama stanja
- ideja se temelji na činjenici da sustav u svom djelovanju prolazi kroz niz promjena stanja
- ovo je imperativni opis sustava

Automati

- često je korisno opisati dio sklopovlja ili dio računalnog programa kao sustav i to korištenjem modela s varijablama stanja
- ovaj pristup omogućava bolju analizu od drugih formalnih opisa
- za početak razmatramo model s varijablama stanja za automate (sustave) s konačnim (i relativno malim) brojem stanja – konačni automati

Uvod u automate

- konačni automati uglavnom su vezani uz nizove događaja
- stanje sustava predstavlja (sažimlje) povijest sustava
- prema tome automat će generirati znak (uzorak) izlaznog signala na temelju ulaznog znaka (uzorka) i stanja sustava

Automati

- preciznije
  1. *razmatra se ulazni znak $x(n)$*
  2. *automat generira izlazni znak uračunavajući znak $x(n)$ i trenutno stanje $s(n)$*
  3. *na temelju znaka $x(n)$ i trenutnog stanja $s(n)$ automat izračunava novo stanje $s(n+1)$. Ovo se naziva prijelaz stanja, ažuriranje ili engleski* ***update***
  4. *automat se vraća na točku 1. postupka i uzima u razmatranje znak $x(n+1)$*
- na ovaj način automat specificira niz izlaznih znakova za niz znakova ulaznog signala

Automati

- opis sustava kao funkcije uključuje tri dijela: skup ulaznih signala, skup izlaznih signala i funkciju samu,

$$F: UlazniSignali \to IzlazniSignali$$

- u slučaju automata ulazni i izlazni signali su oblika

$$NizDogadjaja: Prirodni_0 \to Znakovi$$

gdje su $Prirodni_0=\{0,1,2,\ldots\}$ a $Znakovi$ proizvoljni skup

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*

$$pocetnoStanje \in Stanja$$

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

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

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

Automati

- neka je ulazni signal

$$x \in UlazniSignali$$

- pojedini znak u signalu može se označiti

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

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

- cijeli ulazni signal je niz

$$\left(x(0), x(1), x(2), \ldots, x(n), \ldots\right)$$

Automati

- automat opisan petorkom

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

definira funkciju

$$F : UlazniSignali \rightarrow IzlazniSignali$$

dakle

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

Automati

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

$$s(0) = pocetnoStanje$$

i

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

- 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 $s(n+1) = narednoStanje(s(n), x(n))$

odnosno $y(n) = izlaz(s(n), x(n))$

Automati

- zaključno

$$\forall s(n) \in Stanja \wedge \forall x(n) \in Ulazi,$$
$$(s(n+1), y(n)) = FunkcijaPrijelaza(s(n), x(n))$$
$$= (narednoStanje(s(n), x(n)), izlaz(s(n), x(n)))$$

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

$$odsutan \in Ulazi \quad odsutan \in Izlazi$$

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

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

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

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

Primjer - FunkcijaPrijelaza

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

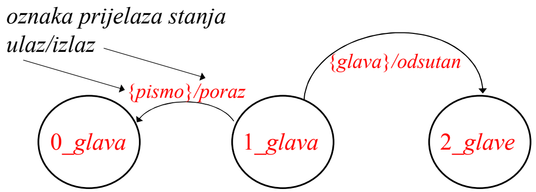
	$x(n)=glava$	$x(n)=pismo$
$s(n)=0\_glava$	$(1\_glava, odsutan)$	$(0\_glava, poraz)$
$s(n)=1\_glava$	$(2\_glave, odsutan)$	$(0\_glava, poraz)$
$s(n)=2\_glave$	$(0\_glava, pobjeda)$	$(0\_glava, poraz)$

Dijagram prijelaza stanja

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

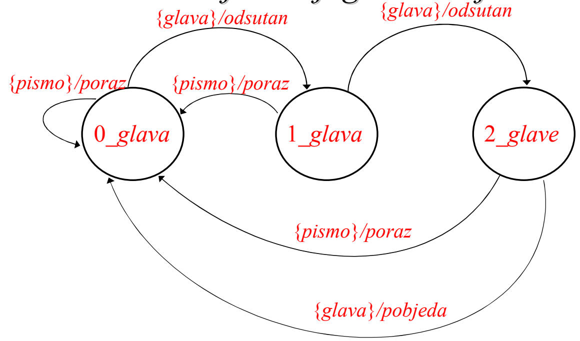
Primjer - dijagram stanja

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

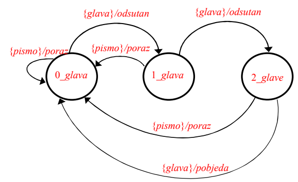


	$x(n)$ = *pismo*	$x(n)$ = *glava*
$s(n)$=1_*glava*	(0_*glava*,*poraz*)	(2_*glave*,*odsutan*

Primjer - dijagram stanja



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

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

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*

Još o Automatima

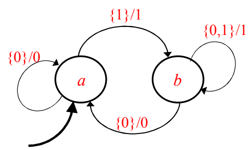
- ako više od jednog ulaznog znaka vodi na isti prijelaz i izlaz oznaka prijelaza “ulaz/izlaz” može sadržavati oznaku skupa ulaznih znakova
- ako za neki automat vrijedi da postoji točno jedan mogući prijelaz za svaku kombinaciju trenutnog stanja i ulaza tada govorimo o determinističkom automatu

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 $y(n)$

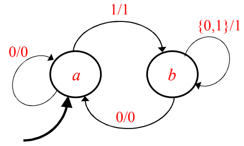
$$mogucaFunkcijaPrijelaza:$$
$$Stanja \times Ulazi \rightarrow P(Stanja \times Izlazi)$$

- 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



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

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

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