

SINTEZA AUTOMATA

- **APSTRAKTNA SINTEZA**
 - zadavanje automata
 - minimizacija
- **STRUKTURNA SINTEZA**
 - kodiranje stanja, ulaza i izlaza
 - uvrštavanje kodova, prepoznavanje:
 - * tablica prijelaza za pojedine bistabile
 - * tablica istine za izlazne varijable
 - realizacija automata
 - * općim bistabilima i logičkim vratima
 - * mux-demux strukturom i D bistabilima (MDD)

APSTRAKTNA SINTEZA AUTOMATA

- **ZADAVANJE AUTOMATA:** tri pristupa
 - transformator sekvence
 - pravila: ulazna sekvenca → izlazna sekvenca
(matematičke gramatike)
 - akceptor sekvence
 - pravila: ulazna sekvenca → izlazni simbol
(jezik regularnih izraza)
 - korak po korak
 - pravila: ulazni simbol → izlazni simbol
(metoda potpunog stabla)

REGULARNI IZRAZI

- **PROBLEM POČETNOG ZAPISA**
 - tablice i grafovi neprikladni
 - to je primitivan formalni jezik (zapisa automata)
 - potpuno stablo uzrokuje eksploziju stanja
- **AKCEPTOR SEKVENCE**
 - preslikavanje ulazne sekvence na izlazni simbol
 - uvodimo viši formalni jezik (zapisa automata)
 - to su regularni izrazi

REGULARNI IZRAZI

- **ULAZI U AUTOMAT**
 - ulazi u automat daju mu informaciju o okolini
 - ulazni simboli dolaze u vremenskim nizovima
 - to su sekvence u diskretnom vremenu
 - te smo sekvence nazivali nizovima DOGAĐAJA
- **ALGEBRA DOGAĐAJA**
 - algebra koja uzima u obzir vremenske odnose
 - bavi se nizovima DOGAĐAJA
 - uvodi ALGEBARSKE OPERACIJE među događajima

ALGEBRA DOGAĐAJA

- **VRSTE DOGAĐAJA**
 - elementarni događaj
 - složeni događaj
- **ELEMENTARNI DOGAĐAJ**
 - nedjeljivi događaj, desi se kao cjelina
 - desi se samo JEDAN u trenutku diskretnog vremena
- **SLOŽENI DOGAĐAJ**
 - nastaje kao kombinacija elementarnih događaja
 - u suštini je vremenski niz - sekvenca - elementarnih

ALGEBRA DOGAĐAJA

- **OPERACIJE MEĐU DOGAĐAJIMA:**
 - konjunkcija, disjunkcija, negacija i iteracija
 - operacijskim vezama među elementarnim događajima dobijemo složene događaje
 - pišemo ALGEBARSKIE IZRAZE u granicama algebre događaja
- **REGULARNI IZRAZI**
 - to su KONAČNI algebarski izrazi u okvirima algebre događaja

ALGEBRA DOGAĐAJA

- **DISJUNKCIJA**

$$R = R_1 \vee R_2$$

- složeni događaj od dva ili više (elementarnih) događaja
- desi se kad se desi JEDAN OD disjunktivno vezanih članova (događaja)
- traje koliko traje disjunktivni član koji se desio
- vrijedi svojstvo komutativnosti:

$$R_1 \vee R_2 = R_2 \vee R_1$$

ALGEBRA DOGAĐAJA

- **KONJUNKCIJA**

$$R = R_1 \& R_2$$

- složeni događaj od dva ili više (elementarnih) događaja
- desi se kad se dese SVI disjunktivno vezani članovi onim redom kojim su zapisani
- traje koliko traju SVI članovi
- opisuje SEKVENCU događaja
- ne vrijedi svojstvo komutativnosti

$$R_1 R_2 \neq R_2 R_1$$

ALGEBRA DOGAĐAJA

- **NEGACIJA**

$$R = \overline{R_1}$$

- događaj koji se desi kad se NE dese negirani događaj
- to je dakle skup SVIH događaja OSIM negiranog događaja
- problem je u određivanju POTPUNOG SKUPA događaja
- potpuni skup događaja može biti beskonačan
- stoga se u praksi ne koristi

ALGEBRA DOGAĐAJA

- **ITERACIJA** (iteracijska zagrada)

$$R = R_1^* = (R_1)^*$$

- događaj koji se desi kad se događaj pod iteracijom desi PROIZVOLJAN BROJ puta, čak i NI JEDNOM
- dakle, to je skup svih sekvenci koje nastaju proizvoljnim ponavljanjem događaja pod iteracijom
- iteracija uključuje i praznu sekvencu Λ :

$$(R_1)^* = (\Lambda \vee R_1 \vee R_1 R_1 \vee R_1 R_1 R_1 \vee R_1 R_1 R_1 R_1 \vee \dots)$$

$$(R_1 \vee R_2)^* = \Lambda \vee R_1 \vee R_2 \vee R_1 R_1 \vee R_1 R_2 \vee R_2 R_1 \vee R_2 R_2 \vee R_1 R_1 R_1 \dots$$

ALGEBRA DOGAĐAJA

- **REGULARNI IZRAZI**
 - to su **KONAČNI** algebarski izrazi u okvirima algebre događaja
- **REGULARNI DOGAĐAJ**
 - svaki složeni događaj koji možemo izraziti konačnim algebarskim izrazom - regularnim izrazom
- **VEZA AUTOMATA I ALGEBRE:**
 - ulazni simbol automata je elementarni događaj
 - sekvenca ulaznih simbola je složeni događaj

JEZIK REGULARNIH IZRAZA

- **PROBLEM ZADAVANJA AUTOMATA**
 - promatramo automat kao akceptor sekvence
 - trebamo opisati traženu sekvencu
 - automat mora odbaciti sekvence koje nisu tražene
 - trebamo opisati i sekvence koje nisu tražene
- **REGULARNIM IZRAZOM**
 - opisujemo ulaznu sekvencu tako da izdvojimo
 - * dio sekvence koji nije tražena sekvenca
 - * dio sekvence koji je tražena sekvenca

JEZIK REGULARNIH IZRAZA

- **VRSTA SEKVENCE DOGAĐAJA**
 - sekvenca sa strukturom
 - sekvenca bez strukture
 - koristimo dvije različite tehnike pisanja regularnog izraza
- **PROBLEM BROJA IZLAZNIH SIMBOLA**
 - jedan ili više akceptorskih izlaznih simbola
 - pišemo zasebni regularni izraz za svaki simbol
 - zajednički dio NE uključuje tražene sekvence

JEZIK REGULARNIH IZRAZA

- **SEKVENCA SA STRUKTUROM**
 - iteracijskom zagradom obuhvatimo sekvence koje NISU tražene
 - običnom zagradom obuhvatimo sekvence koje su tražene za promatrani AKCEPTORSKI (AKTIVNI) izlazni simbol
 - pišemo više izraza ako je više AKCEPTORSKIH izlaznih simbola
 - automat u svim diskretnim trenucima, u kojima NIJE donio odluku, na izlazu daje NEUTRALNI (NEAKTIVNI) izlazni simbol

JEZIK REGULARNIH IZRAZA

• SEKVENCA SA STRUKTUROM - 1. PRIMJER

- među sekvencama po TRI simbola u_1, u_2 tražimo sekvence $u_1u_2u_1$ i $u_1u_2u_2$ za izlazni simbol i_2 , automat daje i_1 u svim ostalim slučajevima
- za sekvence koje nisu tražene

$$\begin{aligned} R_0 &= (u_1 u_1 u_1 \vee u_1 u_1 u_2 \vee u_2 u_1 u_1 \vee u_2 u_1 u_2 \vee u_2 u_2 u_1 \vee u_2 u_2 u_2) = \\ &= (u_1 u_1 (u_1 \vee u_2) \vee u_2 (u_1 \vee u_2) (u_1 \vee u_2)) |_{i_1} \end{aligned}$$

JEZIK REGULARNIH IZRAZA

• SEKVENCA SA STRUKTUROM - 1. PRIMJER

- za tražene sekvence

$$\begin{aligned} R_1 &= (u_1 u_2 u_1 \vee u_1 u_2 u_2) |_{i_2} = \\ &= u_1 u_2 (u_1 \vee u_2) |_{i_2} \end{aligned}$$

- sve skupa, nakon niza netraženih nađe tražena sekvenca:

$$R = (R_0)^* \cdot R_1 |_{i_2}$$

$$R = (u_1 u_1 (u_1 \vee u_2) \vee u_2 (u_1 \vee u_2) (u_1 \vee u_2))^* (u_1 u_2 (u_1 \vee u_2)) |_{i_2}$$

JEZIK REGULARNIH IZRAZA

• SEKVENCA SA STRUKTUROM - 2. PRIMJER

- među sekvencama po DVA simbola u_0, u_1 tražimo sekvencu u_0u_1 za izlazni simbol i_1 , i sekvencu u_1u_0 za izlazni simbol i_2 , automat daje i_0 u svim ostalim slučajevima

- za sekvencu u_0u_1

$$R_1 = (u_0u_0 \vee u_1u_1)^* u_0u_1 \Big|_{i_1}$$

- za sekvencu u_1u_0

$$R_2 = (u_0u_0 \vee u_1u_1)^* u_1u_0 \Big|_{i_2}$$

- dio za netražene sekvence je isti!

JEZIK REGULARNIH IZRAZA

• SEKVENCA BEZ STRUKTURE

- analiziramo dijelove ulazne sekvence
- iteracijskom zagradom obuhvatimo sekvence do duljine tražene koje NISU POČETAK tražene
- upišemo PRVI simbol tražene sekvence
- iteracijskom zagradom obuhvatimo ostatke sekvenci nakon prvog simbola koje NISU POČETAK tražene, ali čiji je ZADNJI simbol PRVI simbol tražene sekvence (ako takvih ima)
- upišemo DRUGI simbol tražene sekvence

JEZIK REGULARNIH IZRAZA

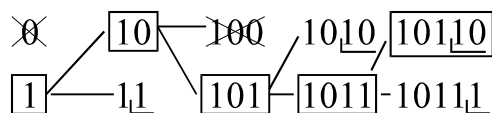
• SEKVENCA BEZ STRUKTURE - NASTAVAK

- iteracijskom zagradom obuhvatimo ostatke sekvenci nakon drugog simbola koje NISU POČETAK tražene, ali čija su ZADNJA DVA simbola PRVA DVA simbola tražene sekvence (ako takvih ima)
- upišemo TREĆI simbol tražene sekvence
- iteracijskom zagradom obuhvatimo ostatke sekvenci nakon trećeg simbola koje NISU POČETAK tražene, ali čija su ZADNJA TRI simbola PRVA TRI simbola tražene sekvence (ako takvih ima)
- itd. do ispisa tražene sekvence

JEZIK REGULARNIH IZRAZA

• SEKVENCA BEZ STRUKTURE - PRIMJER

- napisi regularni izraz za automat koji na izlazu daje 1 ako je posljednjih 5 bita sekvence bez strukture bilo 10110 (dozvoljeno preklapanje):



$$R = (u_0 \vee u_1 u_0 u_0)^* u_1 (u_1 \vee u_0 u_1 u_1 u_1)^* u_0 (u_1 u_0)^* u_1 u_1 u_0 \Big|_{i_1}$$

JEZIK REGULARNIH IZRAZA

- **SINTEZA AUTOMATA**

- označimo MJESTA unutar regularnog izraza
- mjesta mogu biti osnovna i predosnovna
- indeksiramo osnovna mjesta
- rasprostiremo indekse koristeći pravila o rasprostiranju
- dodijelimo IZLAZNI simbol akceptorskim mjestima
- reduciramo indekse koristeći pravila o redukciji
- ispišemo primitivnu tablicu automata!

JEZIK REGULARNIH IZRAZA

- **MJESTA U REGULARNOM IZRAZU**

- OSNOVNA mjesta su ona koja se nalaze s lijeve strane obične i iteracijske zagrade i s desne strane slova
- PREDOSNOVNA su ona koja se nalaze s desne strane obične i iteracijske zagrade i s lijeve strane slova
- između dva slova (simbola) mjesto je istovremeno osnovno i predosnovno, smatramo ga OSNOVNIM
- između dvije zagrade mjesto je istovremeno osnovno i predosnovno, smatramo ga PREDOSNOVNIM

JEZIK REGULARNIH IZRAZA

- **ODNOS MJESTA I AUTOMATA**
 - OSNOVNA mjesta odgovaraju STANJIMA automata
 - to su stanja u koja dolazimo nakon određene sekvence ulaznih događaja
 - PREDOSNOVNA mjesta odgovaraju PRELAZIMA automata
 - to su stanja iz kojih idemo u prelaze

JEZIK REGULARNIH IZRAZA

- **INDEKSIRANJE MJESTA**
 - osnovna mjesta označimo KRATKIM okomitim crtama i indeksiramo u PRVOM redu ispod izraza
 - predosnovna mjesta označimo DUGIM okomitim crtama i indeksiramo u DRUGOM redu ispod izraza
 - osnovnim mjestima dodijelimo VLASTITE indekse, npr. Redom 1, 2 ...
 - na predosnovna mjesta rasprostiremo indekse osnovnih mjesta koristeći 5 pravila (4+pravilo za završno mjesto izraza)

JEZIK REGULARNIH IZRAZA

- **PRAVILA ZA RASPROSTIRANJE INDEKSA**

- 1 indeks mjesta ispred obične i iteracijske zgrade rasprostiremo na POČETNA PREDOSNOVNA mjesta disjunktivno vezanih članova unutar zgrade (zato jer se zagrada - disjunkcija desi, kad se desi jedan od disjunktivno vezanih članova)
- 2 indeks mjesta ispred iteracijske zgrade rasprostiremo na predosnovno mjesto IZA zgrade (zato jer iteracija sadrži i praznu sekvencu)

JEZIK REGULARNIH IZRAZA

- **PRAVILA ZA RASPROSTIRANJE INDEKSA**

- 3 indeks ZAVRŠNIH mjesta disjunktivno vezanih članova unutar obične i iteracijske zgrade rasprostiremo na predosnovno mjesto IZA zgrade (zato jer se zagrada - disjunkcija desi, kad se desi jedan od disjunktivno vezanih članova)
- 4 indekse svih mjesta koja smo rasprostirali na predosnovno mjesto IZA iteracijske zgrade rasprostiremo na POČETNA PREDOSNOVNA mjesta disjunktivno vezanih članova unutar zgrade (zato jer iteracija dozvoljava proizvoljno ponavljanje)

JEZIK REGULARNIH IZRAZA

- **PRAVILA ZA RASPROSTIRANJE INDEKSA**

- završno mjesto regularnog izraza odgovara akceptorskom stanju
- rasprostiranje tog indeksa osigurava rad automata nakon prepoznavanja tražene sekvence

5 indeks ZAVRŠNOG mjesta regularnog izraza rasprostiremo na ona predosnovna mjesta, na koja se je rasprostirao indeks početnog mjesta regularnog izraza iznimno, ako za sekvencu bez strukture dozvoljavamo preklapanje, indeks završnog mjesta rasprostiremo na mjesta na koja se rasprostirao indeks mjesta u koje dolazimo nakon dijela sekvence koji se preklapa

JEZIK REGULARNIH IZRAZA

- **DODJELJIVANJE IZLAZNIH SIMBOLA**

- za MOORE automat akceptorski izlazni simbol dodijelimo završnom OSNOVNOM mjestu regularnog izraza
- za MEALY automat, akceptorski izlazni simbol dodijelimo posljednjem PREDOSNOVNOM mjestu regularnog izraza, a to je uvijek mjesto ISPRED posljednjeg simbola tražene sekvence
- za sva ostala mjesta podrazumijevamo NEUTRALNI izlazni simbol

JEZIK REGULARNIH IZRAZA

- PRAVILA ZA REDUKCIJU INDEKSA

- indekse SVIH mjesta koji su se rasprostirali na ISTO predosnovno mjesto zamijenimo jednim indeksom, pod uvjetom da su im dodijeljeni ISTI izlazni simboli
- indekse SVIH mjesta u koja dolazimo iz ISTOG predosnovnog mjesta s ISTIM ulaznim simbolom zamijenimo jednim indeksom

- ISPIS PRIMITIVNE TABLICE AUTOMATA

- ispišemo primitivnu tablicu tako da preostale indekse osnovnih mjesta uzmemo za stanja automata, a njihove prelaze odredimo preko ulaznih simbola

JEZIK REGULARNIH IZRAZA

- PRIMJER (10110):

- označimo mjesta:

$$R = (|u_0| \vee |u_1| u_0 | u_0|)^* |u_1| (|u_1| \vee |u_0| u_1 | u_1|)^* |u_0| (|u_1| u_0 |)^* |u_1| u_1 | u_0 |$$

- indeksiramo osnovna mjesta redom:

$$R = (|u_0| \vee |u_1| u_0 | u_0|)^* |u_1| (|u_1| \vee |u_0| u_1 | u_1|)^* |u_0| (|u_1| u_0 |)^* |u_1| u_1 | u_0 |$$

- primijenimo prvo i drugo pravilo rasprostiranja:

$$R = (|u_0| \vee |u_1| u_0 | u_0|)^* |u_1| (|u_1| \vee |u_0| u_1 | u_1|)^* |u_0| (|u_1| u_0 |)^* |u_1| u_1 | u_0 |$$

JEZIK REGULARNIH IZRAZA

• PRIMJER (10110):

- nastavimo s MOORE automatom
- primijenimo prvo pravilo za redukciju indeksa:

$$R = (|u_0| \vee |u_1| u_0 | u_0|)^* |u_1| (|u_1| \vee |u_0| u_1 | u_1 | u_1 |)^* |u_0| (|u_1| u_0 |)^* |u_1| u_1 | u_0 |_{17/i}$$

- primijenimo drugo pravilo za redukciju indeksa:

$$R = (|u_0| \vee |u_1| u_0 | u_0|)^* |u_1| (|u_1| \vee |u_0| u_1 | u_1 | u_1 |)^* |u_0| (|u_1| u_0 |)^* |u_1| u_1 | u_0 |_{17/i}$$

JEZIK REGULARNIH IZRAZA

• PRIMJER (10110):

- ispišimo primitivnu tablicu za MOORE automat

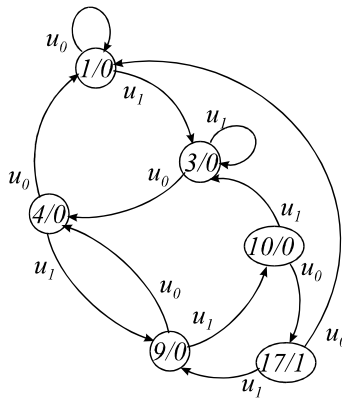
$$R = (|u_0| \vee |u_1| u_0 | u_0|)^* |u_1| (|u_1| \vee |u_0| u_1 | u_1 | u_1 |)^* |u_0| (|u_1| u_0 |)^* |u_1| u_1 | u_0 |_{17/i}$$

	s^{n+1}		i^n
	u_0	u_1	
1	1	3	0
3	4	3	0
4	1	9	0
9	4	10	0
10	17	3	0
17	1	9	1

JEZIK REGULARNIH IZRAZA

- PRIMJER (10110):

– nacrtajmo graf za MOORE automat (10110):



JEZIK REGULARNIH IZRAZA

- PRIMJER dva izlazna simbola:

– indeksiramo zajednički dio isto:

$$R_1 = (|u_0|_1 |u_0|_2 \vee |u_1|_4 |u_1|_5)^* |u_1|_6 |u_0|_7 / i_1$$

$$R_2 = (|u_0|_1 |u_0|_2 \vee |u_1|_4 |u_1|_5)^* |u_0|_8 |u_1|_9 / i_2$$