MODELI RAČUNARSTVA - JEZIČNI PROCESORI 1

Siniša Srbljić, Sveučilište u Zagrebu

1. UVOD
2. REGULARNI JEZICI
3. KONTEKSTNO NEOVISNI JEZICI
4. REKURZIVNO PREBROJIVI JEZICI
5. KONTEKSTNO OVISNI JEZICI
6. RAZREDBA (TAKSONOMIJA) JEZIKA, AUTOMATA I GRAMATIKA

4. REKURZIVNO PREBROJIVI JEZICI

- 4.1. TURINGOV STROJ
- 4.2. GRAMATIKA NEOGRANIČENIH PRODUKCIJA
- 4.3. SVOJSTVA REKURZIVNIH I REKURZIVNO PREBROJIVIH JEZIKA

4. Rekurzivno prebrojivi jezici

- REKURZIVNO PREBROJIVI JEZICI
  - zasnivaju se na Turingovom stroju:
    - jezik **jest** rekurzivno prebrojiv
    - ako i samo ako postoji Turingov stroj koji ga prihvaća
  - time je definirana istovjetnost RPJ i TS:
    - za bilo koji rekurzivno prebrojiv jezik
    - **moguće** je izgraditi Turingov stroj koji ga prihvaća

4.1. Turingov stroj, TS (Turing Machine, TM)

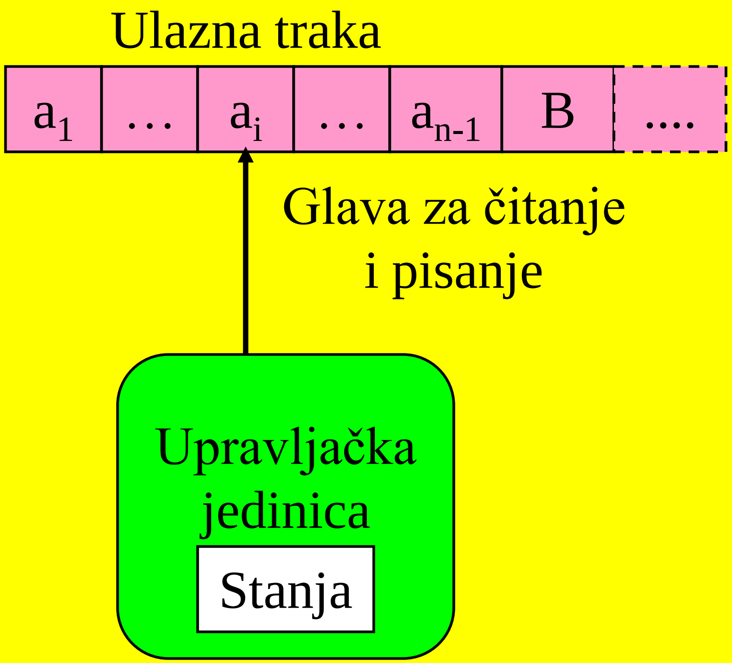
- 4.1.1. Osnovni model Turingovog stroja
- 4.1.2. Metode izrade Turingovog stroja
- 4.1.3. Prošireni modeli Turingovog stroja
- 4.1.4. Pojednostavljeni modeli Turingovog stroja
- 4.1.5. Generiranje jezika Turingovim strojem

4.1. Turingov stroj TS

- TURINGOV STROJ
  - je jednostavan
  - bez obzira na to, predstavlja opći model računanja
  - osnovna primjena TS je prihvaćanje jezika
  - obzirom da može pisati po traci
    - TS se koristi za generiranje jezika
    - TS se koristi za računanje cjelobrojnih funkcija

4.1.1. Osnovni model Turingovog stroja

- DEFINICIJA TURINGOVOG STROJA
  - TS nakon čitanja piše novi znak na traku
  - glava se pomiče lijevo i desno
  - traka je neograničena s desne strane
  - na početku, niz w je upisan krajnje lijevo ostatak je popunjen s B



Osnovni model Turingovog stroja

- DONOŠENJE ODLUKE I AKCIJA
  - upravljačka jedinica donosi odluku o
    - promjeni stanja
    - upisu znaka na traku
    - pomaku glave za čitanje i pisanje
  - odluku donosi na osnovu
    - stanja
    - znaka na traci

Osnovni model Turingovog stroja

- FORMALNA DEFINICIJA TS
  - formalno zadajemo sedmorkom:
    $TS = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$
  - gdje su
    - Q konačan skup stanja s početnim stanjem q_0
    - Γ konačan skup znakova trake s praznim znakom B
    - Σ konačan skup znakova trake bez praznog znaka B
    - δ funkcija prijelaza $Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$
    - $F \subseteq Q$ podskup prihvatljivih stanja
  - funkcija $\delta(q, v) = (p, z, W)$:
    - na osnovu stanja q i pročitanog znaka v
    - određuje slijedeće stanje p, znak za upis na traku z i pomak W

Osnovni model Turingovog stroja

- TURINGOV STROJ PRIMJER
  - Zadan je TS M
    $M = (\{q_0,q_1,q_2,q_3,q_4\}, \{0,1\}, \{0,1,X,Y,B\}, \delta, q_0, B, \{q_4\})$

	0	1	X	Y	B
q_0	(q_1,X,R)			(q_3,Y,R)	
q_1	$(q_1,0,R)$	(q_2,Y,L)		(q_1,Y,R)	
q_2	$(q_1,0,L)$		(q_0,X,R)	(q_2,Y,L)	
q_3				(q_3,Y,R)	(q_4,B,R)

  - M prihvaća jezik $L(M) = \{0^n1^n \mid n \geq 1\}$
    - u q_0 zamijeni krajnje lijevu 0 s X i u q_1 traži prvu 1 desno
    - u q_1 zamijeni krajnje lijevu 1 s Y i u q_2 traži krajnji X lijevo
    - u q_2 ostavi krajnje desni X i nastavi od q_0
    - ako u q_0 pročita Y, nema više 0, provjeri u q_3 ima li koja 1

Osnovni model Turingovog stroja

- PRIHVAĆANJE JEZIKA TSom
  - rad TS opisujemo konfiguracijom $\alpha_1 q \alpha_2$ sa značenjem "lijevi dio trake - stanje - desni dio trake"
  - TS čita krajnje lijevi znak niza α_2
  - oznake za q moraju se razlikovati od oznaka za α
  - pozicija glave se numerira:
    $X_1\ X_2 \ldots X_{i-1}\ q\ X_i\ X_{i+1} \ldots X_{n-1}\ X_n$
    - lijevo od X_1 nema trake, nema lijevog pomaka
    - desno od X_n je sve prazno, samo B
  - za i>1 i $\delta(q,X_i) = (p,Y,L)$ TS promijeni konfiguraciju:

$$X_1 X_2 \cdots X_{i-1} q X_i X_{i+1} \cdots X_n \underset{M}{\succ} X_1 X_2 \cdots X_{i-2} p X_{i-1} Y X_{i+1} \cdots X_n$$

Osnovni model Turingovog stroja

- PRIHVAĆANJE JEZIKA TSom
  - za $\delta(q, X_i) = (p, Y, R)$ TS promijeni konfiguraciju:

$$X_1 X_2 \cdots X_{i-1} q X_i X_{i+1} \cdots X_n \underset{M}{\succ} X_1 X_2 \cdots X_{i-1} Y p X_{i+1} \cdots X_n$$

  - slijedom prijelaza: $q_0 0011 \overset{*}{\underset{M}{\succ}} XXYYBq_4$

  automat M prihvati niz 0011:

$$q_0 0011 \succ Xq_1 011 \succ X0q_1 11 \succ Xq_2 0Y1 \succ q_2 X0Y1 \succ$$
$$Xq_0 0Y1 \succ XXq_1 Y1 \succ XXYq_1 1 \succ XXq_2 YY \succ Xq_2 XYY \succ$$
$$XXq_0 YY \succ XXYq_3 Y \succ XXYYq_3 \succ XXYYBq_4$$

Osnovni model Turingovog stroja

- PRIHVAĆANJE JEZIKA TSom
  - izraz: $J \overset{m}{\underset{M}{\succ}} K$

    znači da TS iz konfiguracije J prelazi u K u m koraka
  - za niz 0011 treba 13 prijelaza:

$$q_0 0011_n \overset{13}{\underset{M}{\succ}} XXYYBq_4$$

  - automat M prihvaća jezik koji ga dovodi u prihvatljivo stanje:

$$L(M) = \left\{ w \middle| w \in \Sigma^* \ \text{i} \ q_0 w \overset{*}{\underset{M}{\succ}} \alpha_1 p \alpha_2 \ ; \ p \in F \right\}$$

  - ako niz nije prihvatljiv, TS ne mora stati

Osnovni model Turingovog stroja

- PRIHVAĆANJE JEZIKA TSom
  - TS prihvaća klasu **rekurzivno prebrojivih** jezika
    - ime na osnovu svojstva da TS ispisuje (nabraja) sve nizove
    - termin “rekurzivno” je u današnjem značenju
  - kontekstno neovisni jezici su pravi podskup rekurzivno prebrojivih jezika
  - za niz koji nije član jezika, TS ne mora stati
  - za jezike gdje TS uvijek stane, kaže se da su samo rekurzivni
  - klasa rekurzivnih jezika je podskup klase rekurzivno prebrojivih jezika

Osnovni model Turingovog stroja

- CJELOBROJNA ARITMETIKA TSom
  - TS se koristi za računanje cjelobrojnih funkcija
  - na traci se koristi notacija 0^n10^m1....
    - niz n nula ima vrijednost n, niz m nula vrijednost m
    - nizovi nula odijeljeni su znakom 1
    - to je sustav s bazom s=1 (rimska notacija!)
  - TS izračunava **parcijalno rekurzivne** funkcije
  - ParRF i RPJ su ekvivalentni, se ne mora zaustaviti
  - ako se TS uvijek zaustavi,
    radi se o **potpuno rekurzivnim funkcijama**
  - PotRF i RekJ su ekvivalentni, TS se uvijek zaustavi

Osnovni model Turingovog stroja

- CJELOBROJNA ARITMETIKA TSom PRIMJER
  - TS računa m-n, ako je m<n razlika je 0
  - za niz 0^m10^n
    - TS za skida 0 lijevog broja i 0 desnog broja (upisuje 1)
    - ako je m>n ostane lijevo 0^{m-n}
    - ako je m=n ne ostane niti jedna nula
    - ako je m<n ostanu nule desno, pa ih TS mora ukloniti
  - gradi se TS $M = (\{q_0,\ldots,q_6\}, \{0,1\}, \{0,1,B\}, \delta, q_0, B, \emptyset)$

$\delta(q_0,0)=(q_1,B,R)$ početna zamjena

$\delta(q_1,0)=(q_1,0,R)$ $\delta(q_1,1)=(q_2,1,R)$ traži desni broj

$\delta(q_2,1)=(q_2,1,R)$ $\delta(q_2,0)=(q_3,1,L)$ zamjeni i idi natrag

$\delta(q_3,0)=(q_3,0,L)$ $\delta(q_3,1)=(q_3,1,L)$ $\delta(q_3,B)=(q_0,B,R)$ idi natrag

Osnovni model Turingovog stroja

- CJELOBROJNA ARITMETIKA TSom PRIMJER
  - ako je m>n , briši jedinice i vrati nulu:

    $\delta(q_2,B)=(q_4,B,L)$ okreni natrag!

    $\delta(q_4,1)=(q_4,B,L)$ briši jedinice

    $\delta(q_4,0)=(q_4,0,L)$ $\delta(q_4,B)=(q_6,0,R)$ idi lijevo i vrati nulu, STOP
  - ako je m<n briši:

    $\delta(q_0,1)=(q_5,B,R)$ briši jedinice desno

    $\delta(q_5,1)=(q_5,B,R)$ $\delta(q_5,0)=(q_5,B,R)$ briši nule i jedinice desno

    $\delta(q_5,B)=(q_6,B,R)$ sve obrisano, STOP

Osnovni model Turingovog stroja

- CJELOBROJNA ARITMETIKA TSom PRIMJER
  - za niz 0010 vrijedi 2-1=1:

$$q_0 0010 \succ Bq_1 010 \succ B0q_1 10 \succ B01q_2 0 \succ B0q_3 11 \succ$$
$$Bq_3 011 \succ q_3 B011 \succ Bq_0 011 \succ BBq_1 11 \succ BB1q_2 1 \succ$$
$$BB11q_2 \succ BB1q_4 1 \succ BBq_4 1 \succ Bq_4 \succ B0q_6$$

  - za niz 0100 vrijedi 1-2=0:

$$q_0 0100 \succ Bq_1 100 \succ B1q_2 00 \succ Bq_3 110 \succ q_3 B110 \succ$$
$$Bq_0 110 \succ BBq_5 10 \succ BBBq_5 0 \succ BBBBq_5 \succ BBBBBq_6$$

4.1.2. Metode izrade Turingovog stroja

- VIŠEKOMPONENTNE OZNAKE STANJA
  - olakšavaju izradu TS
  - uz osnovno stanje bilježi se informacija koju ono nosi
  - koriste se uglate zagrade: [q,a,b...]
    - znači da u stanju q nosimo parametre a, b itd.
  - stanje je **upravljačka komponenta**
    - upravljačka komponenta može biti samo jedna
  - informacija je **radna komponenta**
    - radnih komponenti može biti više

Metode izrade Turingovog stroja

- VIŠEKOMPONENTNA STANJA PRIMJER
  - izgraditi TS koji prihvaća nizove kod kojih se prvi znak ne ponavlja unutar niza
  - koristimo višekomponentno stanje [q,a]
  - upravljačka komponenta ima dva stanja
    - q_0, prije čitanja prvog znaka
    - q_1, prvi znak je pročitan
  - radna komponenta koristi se za pamćenje prvog znaka
    - može poprimiti vrijednosti 0, 1 i B
  - imamo svega šest stanja:
    - $[q_0,B]$, $[q_0,0]$, $[q_0,1]$, $[q_1,B]$, $[q_1,0]$, $[q_1,1]$

Metode izrade Turingovog stroja

- VIŠEKOMPONENTNA STANJA PRIMJER
  - gradimo TS
    $M = (Q, \{0,1\}, \{0,1,B\}, \delta, [q_0,B], B, \{[q_1,B]\})$
  - funkcija prijelaza je:

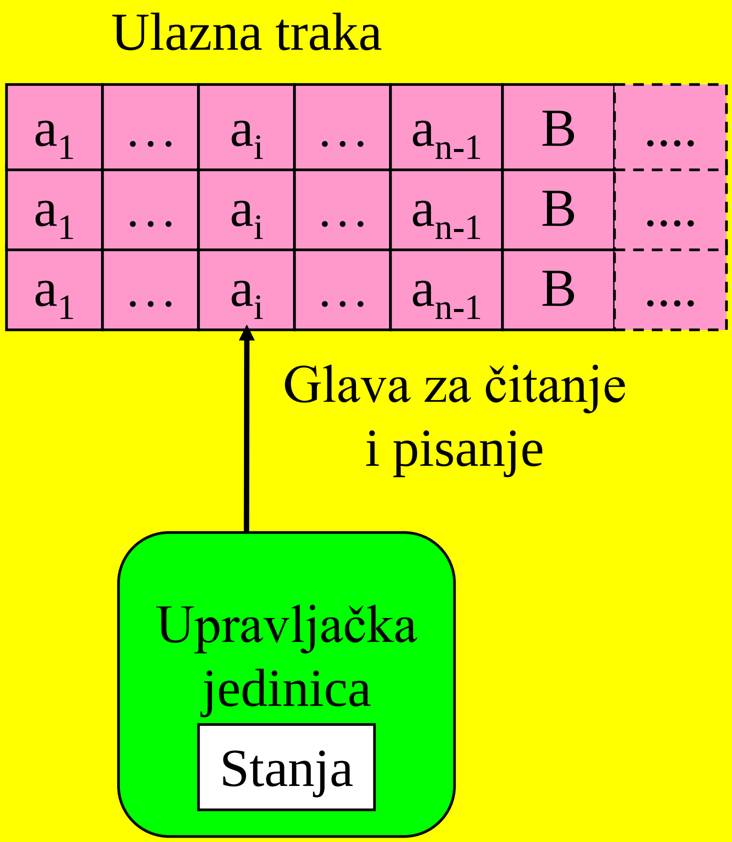
    $\delta([q_0,B], 0) = ([q_1,0],0,R)$ $\delta([q_0,B], 1) = ([q_1,1],1,R)$

    $\delta([q_1,0], 1) = ([q_1,0],1,R)$ $\delta([q_1,1], 0) = ([q_1,1],0,R)$

    $\delta([q_1,0], B) = ([q_1,B],B,L)$ $\delta([q_1,1], B) = ([q_1,B],B, L)$
  - drugi primjer: pomak znakova za n mjesta u desno
    - koristimo n radnih komponenti
    - u svakom koraku prelazimo u novo stanje tako da rotiramo radne komponente

Metode izrade Turingovog stroja

- VIŠEKOMPONENTNI ZNAKOVI TRAKE
  - znakovi trake mogu imati više komponenti
  - za konačni skup komponenti,
    rad je ekvivalentan TS
  - komponente zapišemo u posebne tragove trake
  - ukupni znak trake je kombinacija komponenti



Metode izrade Turingovog stroja

- VIŠEKOMPONENTNI ZNAKOVI TRAKE
  - tragove možemo koristiti za posebne namjene
  - npr. pomoćni trag može se koristiti za označivanje ranije ispitanih znakova
  - korisno za TS koji prihvaćaju jezike
    - gdje se dijelovi niza ponavljaju
    - gdje se dijelovi niza uspoređuju

4.1.3. Prošireni modeli Turingovog stroja

- PROŠIRENJA OSNOVNOG MODELA
  - TS s dvostranom beskonačnom trakom
  - TS s višestrukim trakama
  - TS s višedimenzionalnim ulaznim poljem
  - TS s više glava za čitanje i pisanje
  - Neizravni TS

4.1.4. Pojednostavljeni modeli TS

- POJEDNOSTAVLJENJA OSNOVNOG MODELA
  - stogovni stroj
  - stroj s brojilima
  - TS s ograničenim brojem stanja i znakova trake
  - univerzalni TS M_u

4.1.5. Generiranje jezika Turingovim strojem

- GENERIRANJE JEZIKA
  - koristi se TS s višestrukim trakama
  - jedna traka je izlazna,
    jednom upisani znak ne može se mijenjati
  - glava za čitanje i pisanje izlazne trake miče se u desno
  - nizovi jezika G(M) ispisuju se na izlaznu traku, kraj je #
  - ako TS stane, G je konačan, inače je beskonačan
  - TS generira klasu rekurzivno prebrojivih jezika
  - jezik je rekurzivno prebrojiv
    samo ako postoji TS koji ga generira

Generiranje jezika Turingovim strojem

- PRIHVAĆANJE JEZIKA GENERIRANOG TS
  - za TS M_1 koji generira $G(M_1)$ moguće je izgraditi TS M_2 koji prihvaća $L(M_2) = G(M_1)$
  - gradi se TS M_2
    - ima jednu traku više od TS M_1 i to je ulazna traka
  - TS M_2 generira neki niz
    - ako je niz istovjetan zadanom, niz se prihvaća
    - inače, generira se sljedeći niz

Generiranje jezika Turingovim strojem

- GENERIRANJE JEZIKA PRIHVAĆENOG TS
  - za TS M_2 koji prihvaća $L(M_2)$ moguće je izgraditi TS M_1 koji generira $G(M_1) = L(M_2)$
  - gradi se TS M_1
    - jednostavan, generira rekurzivni jezik, uvijek stane
    - složen, generira rekurzivno prebrojiv jezik, nekad ne stane
  - za rekurzivne jezike jednostavni TS
    - ispisuje sve nizove na radnu traku
    - simulira rad M2 i provjeri novi niz
    - ako je niz prihvatljiv, kopira ga sa radne trake na izlaznu

Generiranje jezika Turingovim strojem

- GENERIRANJE JEZIKA PRIHVAĆENOG TS
  - za rekurzivno prebrojive jezike jednostavni TS
    - nekad ne stane
    - stoga ne može pokrenuti ispitivanje slijedećeg niza
  - stoga se koristi složeni TS
  - složeni TS
    - generira par i,j rastućim redoslijedom i+j, i
    - nakon toga generira niz duljine i na radnu traku
    - na kraju se provjerava prihvatljivost niza u j koraka
  - tako se izbjegne situacija da se TS ne zaustavi

Generiranje jezika Turingovim strojem

- ISTOVJETNOST RekJ i KANONSKOG SLIJEDA
  - u kanonskom slijedu kraći nizovi su ispred duljih
  - redoslijed nizova iste duljine
    - određuje se po numeričkoj vrijednosti
    - kao broj s bazom jednakom broju znakova
  - rekurzivne jezike moguće je generirati kanonskim slijedom
    - za generiranje se može koristiti jednostavan TS
    - neka sve nizove generira kanonskim slijedom na radnu traku
    - tada će na izlaznoj traci prihvaćeni nizovi biti isto poredani

Generiranje jezika Turingovim strojem

- ISTOVJETNOST RekJ i KANONSKOG SLIJEDA
  - jezik $G(M_1)$ kojeg je moguće generirati kanonskim slijedom jest rekurzivni jezik
  - Neka
    - TS M_1 generira jezik $G(M_1)$ kanonskim slijedom
    - TS M_2 prihvaća jezik $G(M_2)=G(M_1)$ i koji uvijek stane
  - TS M_2 simulira rad TS M_1
    - M_2 generira niz kanonskim slijedom sve do w_i ili w_j
    - generira li w_i, niz je prihvatljiv i TS stane
    - generira li w_j, $j>i$, w_i sigurno nije prihvatljiv i automat stane

Generiranje jezika Turingovim strojem

- ISTOVJETNOST RekJ i KANONSKOG SLIJEDA
  - ako je jezik $G(M_1)$ beskonačan, TS M_2 uvijek stane
  - ako je jezik $G(M_1)$ konačan
    i ako se ispuituju nizovi iza svih nizova od L (M_2)
    - TS M_2 nikad ne stane
  - nije moguće izgraditi TS za opći slučaj konačnog jezika
  - moguće je izgraditi pojedinačne TS

4.2. Gramatika neograničenih produkcija

- 4.2.1. Konstrukcija TS za jezik zadan GNP
- 4.2.2. Konstrukcija gramatike za jezik zadan TS

4.2. Gramatika neograničenih produkcija

- DOZVOLJENI OBLICI PRODUKCIJA
  - za kontekstno neovisne gramatike
    - na lijevoj strani imaju samo jedan nezavršni znak
  - za regularne gramatike
    - dodatno, još su lijevo linearne i desno linearne
  - za gramatike neograničenih produkcija $\alpha \to \beta$:
    - α i β su nizovi završnih i nezavršnih znakova
    - niz α ne smije biti prazan, $\alpha \neq \varepsilon$
  - gramatika neograničenih produkcija zive se **gramatika tipa 0**

Gramatika neograničenih produkcija

- FORMALNA SPECIFIKACIJA
  - gramatika neograničenih produkcija je četvorka:
    $$G = (V, T, P, S)$$
  - za produkciju $\alpha \to \beta$ definira se relacija $\Rightarrow$
    $$\gamma\alpha\delta \Rightarrow \gamma\beta\delta$$
  - refleksivno i tranzitivno okruženje relacije $\Rightarrow$ je $\overset{*}{\Rightarrow}$
  - G = (V, T, P, S) generira jezik:
    $$L(G) = \left\{ w \,\middle|\, w \in T^*;\ S \overset{*}{\Rightarrow} w \right\}$$
  - L(G) pripada klasi rekurzivno prebrojivih jezika

Gramatika neograničenih produkcija

- PRIMJER

4.2.1. Konstrukcija TS za jezik zadan GNP

- KONSTRUKCIJA TS IZ GNP
  - G = (V, T, P, S)
    - ako je gramatika neograničenih produkcija i generira L(G),
    - tada je L(G) rekurzivno prebrojiv jezik
  - po definiciji
    - L(G) je rekurzivno prebrojiv
    - ako postoji TS koji ga prihvaća: L(M) = L(G)

Konstrukcija TS za jezik zadan GNP

- KONSTRUKCIJA TS IZ GNP
  - gradi se nedeterministički TS (NTS)
    - s dvije trake
    - simulira rad gramatike G
  - postupak:
    - na prvu traku zapiše se niz w
    - na drugu traku zapiše se početni nezavršni znak gramatike S
    - tijekom simulacije, nadrugu traku ispisuju se nizovi α
    - TS uspoređuje nizove α s nizom w i eventualno ga prihvati

Konstrukcija TS za jezik zadan GNP

- KONSTRUKCIJA TS IZ GNP
  - simulacija gramatike G se izvodi:

1. TS nedeterministički izabere mjesto i u nizu α druge trake, dakle za sve vrijednosti od i
2. po svim i, TS nedeterministički izabere produkciju $\beta\rightarrow\gamma$, dakle sve produkcije
3. po svim i, $\beta\rightarrow\gamma$, ako je β na mjestu i zamijeni se s γ
4. niz generiran na drugoj traci uspoređuje se s nizom w
   - ako su nizovi jednaki, TS se zaustavlja i prihvaća w
   - ako nisu jednaki, TS nastavlja od koraka 1.

4.2.2. Konstrukcija GNP za jezik zadan TS

- KONSTRUKCIJA GNP IZ TS
  - ako TS prihvaća RPJ L(M),
    - postoji gramatika neograničenih produkcija GNP
    - generira L(G) = L(M)
  - gradi se G = (V, T, P, S)
    koja simulira rad TS $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$
    - gramatika G generira redom međunizove znakova
    - oni predstavljaju konfiguracije TS M

Konstrukcija GNP za jezik zadan TS

- PRIMJER

4.3. Svojstva rekurzivnih i rekurzivno prebrojivih jezika

- 4.3.1. Svojstva zatvorenosti
- 4.3.2. Izračunljivost
- 4.3.3. Odlučivost

4.3.1. Svojstva zatvorenosti RekJ i RPJ

- SVOJSTVA ZATVORENOSTI
  - koriste se TS
  - na temelju TS koji uvijek stane
    - RekJ su zatvoreni obzirom na uniju i komplement
    - RPJ zatvoreni su obzirom na operaciju unije
  - važno svojstvo:
    - ako je komplement RPJ L rekurzivno prebrojiv
    - tada su L i L komplement rekurzivni!
    - nije li LC rekurzivno prebrojiv, L sigurno nije rekurzivan a vjerojatno nije ni rekurzivno prebrojiv

Svojstva zatvorenosti RekJ i RPJ

- UNIJA REKURZIVNIH JEZIKA
  - unija rekurzivnih jezika jest rekurzivni jezik
  - TS M kao serijski spoj TS M_1 i TS M_2
    - TS M_1 prihvaća RekJ L_1
    - TS M_2 prihvaća RekJ L_2
  - ako M_1 stane i prihvati niz, M stane i prihvati niz
  - ako M_1 stane i ne prihvati niz, starta se M_2
  - ako M_2 stane i prihvati niz, M stane i prihvati niz
  - ako M_2 stane i ne prihvati niz, M ne prihvati niz

Svojstva zatvorenosti RekJ i RPJ

- UNIJA REKURZIVNO PREBROJIVIH JEZIKA
  - unija rekurzivno prebrojivih jezika jest rekurzivno prebrojivi jezik
  - TS M kao paralelni spoj TS M_1 i TS M_2
    - TS M_1 prihvaća RPJ L_1
    - TS M_2 prihvaća RPJ L_2
  - ako M_1 stane i prihvati niz, M stane i prihvati niz
  - ako M_2 stane i prihvati niz, M stane i prihvati niz

Svojstva zatvorenosti RekJ i RPJ

- KOMPLEMENT REKURZIVNOG JEZIKA
  - komplement rekurzivnog jezika jest rekurzivni jezik
  - TS M kao komplement TS M_1 (okrenuti izlazi)
    - TS M_1 prihvaća RPJ L_1
  - ako M_1 stane i prihvati niz, M stane i **ne** prihvati niz
  - ako M_1 stane i **ne** prihvati niz, M stane i prihvati niz

Svojstva zatvorenosti RekJ i RPJ

- KOMPLEMENT REK. PREBROJIVIH JEZIKA
  - ako su jezik L i njegov komplement rekurzivno prebrojivi, tada su oba jezika **rekurzivni**!
  - TS M kao paralelni spoj TS M_1 i TS M_2
    - TS M_1 prihvaća RPJ L_1
    - TS M_2 prihvaća RPJ $L_2 = L_1^C$
  - ako M_1 stane i prihvati niz, M stane i prihvati niz
  - ako M_2 stane i prihvati niz, M stane i **ne** prihvati niz
  - M uvijek stane, dakle L i L^c su rekurzivni!

Svojstva zatvorenosti RekJ i RPJ

- KOMPLEMENT REK. PREBROJIVIH JEZIKA
  - za L i L^c vrijedi jedno od tri svojstava:

1. oba jezika L i L^c su rekurzivna
2. oba jezika L i L^c nisu rekurzivno prebrojivi
3. ako je L rekurzivno prebrojiv, ali ne i rekurzivan, L^c sigurno nije rekurzivno prebrojiv

4.3.2. Izračunljivost

- DEFINICIJA IZRAČUNLJIVOSTI
  - definiramo intuitivno
  - problem jest izračunljiv
    - prihvaćanje jezika
    - računanje cjelobrojnih funkcija
    - generiranje jezika

    ako postoji automat koji korak po korak (mehanički) rješava zadani problem
  - ne nameću se nikakva ograničenja
    - na broj koraka
    - niti na veličinu spremnika
  - ne traži se da postupak ikad stane

Izračunljivost

- CHURCH-TURINGOVA HIPOTEZA
  - na temelju intuitivne definicije izračunljivosti
  - izračunljive funkcije su
    parcijalno rekurzivne funkcije
  - pokazuje se samo prikladnost ove hipoteze
  - parcijalno rekurzivne funkcije su izračunljive,
    jer je za njihovo računanjemoguće izgraditi TS

Izračunljivost

- SIMULACIJA RAM RAČUNALA S TS
  – probem kodiranja

Izračunljivost

- NEIZRAČUNLJIVOST DIJAGONALNOG JEZIKA
  - probem kodiranja

4.3.3. Odlučivost

- DEFINICIJA ODLUČIVOSTI
  - rekurzivni jezici su odlučivi
    - jer ih prihvaćaju TS koji uvijek stanu
  - rekurzivni jezici su izračunljivi i odlučivi
  - rekurzivno prebrojivi nisu odlučivi
    - jer ne postoji TS koji će uvijek stati
    - nije li w u jeziku, moguće je da TS nikad ne stane
    - ako TS ne stane , nema odluke da w ne pripada jeziku
  - rekurzivno prebrojivi jezici su izračunljivi i neodlučivi
  - univerzalni jezik Lu je primjer
    - rekurzivno prebrojivog jezika
    - dakle izračunljivog jezika koji **nije** odlučiv

Odlučivost

- UNIVERZALNI TS I UNIVERZALNI JEZIK

Odlučivost

- IZRAČUNLJIVOST I ODLUČIVOST UNIVERZALNOG JEZIKA