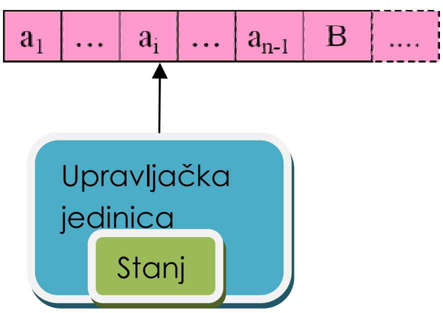
21.TURINGOV STROJ

21.1.Formalna definicija Turingovog stroja

— Rekurzivno prebrojivi jezici i Turingov stroj
— Osnovni model Turingovog stroja
— Donošenje odluka Turingovog stroja
— Formalna definicija Turingovog stroja

Jezik je **rekurzivno prebrojiv** ako i samo ako postoji Turingov stroj koji ga prihvaća. Time je definirana istovjetnost Turingovog stroja i rekurzivno prebrojivih jezika.



Na temelju pročitanog znaka i stanja, Turingov stroj odlučuje:

- Na koju stranu se miče glava za čitanje i pisanje
- Koji znak se zapisuje na traku umjesto pročitanog znaka
- O prijelazu u novo stanje

Turingov stroj se formalno zadaje sedmorkom TS = (Q, Σ, Γ, δ, q_0, B, F). Q → konačan broj stanja; Σ→konačan skup ulaznih znakova trake; Γ→ konačan skup znakova trake; δ → funkcija prijelaza: $Q \times \Gamma \to Q \times \Gamma \times \{L,R\}$; q_0→ početno stanje; B → znak kojim se označava prazna ćelija; F→ skup prihvatljivih stanja.

21.2. Prihvaćanje jezika Turingovim strojem

— Konfiguracija Turingovog stroja
— Prihvaćanje niza (?)
— Prihvaćanje jezika
— Rekurzivni i rekurzivno prebrojivi jezici

Konfiguracija Turingovog stroja se zadaje sadržajem ćelija lijevo od glave za čitanje i pisanje, stanjem TS-a i sadržajem ćelija desno od glave za čitanje i pisanje.

TS prihvaća jezik koji ga dovodi u prihvatljivo stanje: $\alpha_1 q \alpha_2$

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

TS prihvaća klasu rekurzivno prebrojivih jezika. Postoje rekurzivno prebrojivi nizovi za koje nije moguće izgraditi TS koji uvijek stane za bilo koji ulazni niz. Klasa jezika za koju je moguće izgraditi TS koji uvijek stane za bilo koji ulazni niz se naziva klasa **rekurzivnih jezika.**

21.3. Cjelobrojna aritmetika Turingovim strojem

- osnovna sintaksa zapisa
- rekurzivne funkcije i jezici
- primjer

TS se koristi za računanje vrijednosti cjelobrojnih funkcija. Uobičajeno je da se cijeli brojevi zapišu nizom znakova 0. Broj znakova 0 označava vrijednost cijelog broja. (Broj $i \geq 0$ se zapiše: 0^i). Ako cjelobrojna funkcija ima k argumenata tada se oni na traci odvoje sa znakom 1.

Funkcije koje je moguće izračunati koristeći TS se nazivaju **parcijalno rekurzivne funkcije**. Rekurzivno prebrojivi jezici i parcijalno rekurzivne funkcije su analogni u smislu da se prihvaćaju, odnosno da se računaju sa TS koji ne staje uvijek. Ako je funkcija $f(i_1,i_2,\ldots,i_k)$ definirana za sve argumente tada je f **potpuno rekurzivna funkcija.** Rekurzivni jezici i potpuno rekurzivne funkcije su analogni u smislu da se prihvaćaju, odnosno da se računaju sa TS koji uvijek stane.

Primjer za cjelobrojnu aritmetiku je TS koji računa razliku dvaju brojeva. (m-n) . Za niz 0^m10^n TS skida 0 lijevog broja i 0 desnog broja (upisuje 1). Ako je m>n lijevo ostane 0^{m-n}; a ako je m=n ne ostane niti jedna nula.

22. SVOJSTVA TURINGOVOG STROJA

22.1. Višekomponentna stanja i znakovi trake

- višekomponentne oznake stanja
- višekomponentni znakovi trake
- primjeri

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,..). Stanje je **upravljačka komponenta**, dok je informacija **radna komponenta**.

Moguće je da znakovi trake i ulazni znakovi imaju više komponenata: $A_j=[a_1,a_2,\ldots, a_n]$, gdje je A_j složeni znak s komponentama a_i. Rad TS je lakše pratiti ako se pojedine komponente složenih znakova trake zapišu u zasebne tragove ulazne trake. Za konačan skup komponenti rad je ekvivalentan TS.

Primjer: Za TS koji prihvaća jezik $L(M)=\{ wcw \mid w \in (a+b)^+\}$. Složena stanja [q,d] imaju upravljačku komponentu q i radnu komponentu d (poprima vrijednost ulaznih znakova ili oznake prazne ćelije). Ulazna traka ima dva traga: ulazni trag i označni trag. Složeni znakovi trake [X,d] imaju ulaznu komponentu d i označnu komponentu X.

22.2. Proširenja i pojednostavljenja Turingovog stroja

- proširenja trake, neizravni TS
- stogovni stroj i stroj s brojilima
- ograničenja stanja i trake
- univerzalni TS

Šest osnovnih načina proširenja osnovnog modela TS su: TS s dvostranom beskonačnom trakom, TS s višestrukim trakama, nedeterministički TS, TS s višedimenzionalnim ulaznim poljem, TS s više glava za čitanje i pisanje, te neizravni TS. **Neizravni TS** se koristi u istraživanjima prostorne složenosti prihvaćanja jezika i prostorne složenosti računanja cjelobrojnih funkcija. Neizravni TS ima više radnih traka i jednu ulaznu traku.

Stogovni stroj je deterministički TS s jednom ulaznom trakom i više stogova. Stog je posebna radna traka pojednostavljenih funkcija prijelaza. Kod stroja s brojilima umjesto dva stoga se koriste četiri brojila. Skup znakova stoga se ograniči na dva znaka: oznaku dna stoga X i oznaku prazne ćelije B.

Ako se istodobno ograniči broj znakova trake, broj traka i broj stanja, onda je ograničen broj različitih Turingovih strojeva koje je moguće izgraditi- Zato TS s ograničenim brojem znakova trake ne prihvaća isti skup jezika kao i osnovni model TS.

Primjenom univerzalnog TS M_U moguće je simulirati rad bilo kojeg TS M s jednom trakom. Univerzalni TS ima tri trake. Na prvu traku zapišu se funkcije prijelaza TS M i niz w. Sadržaj druge trake simulira sadržaj trake TS M, dok se na treću traku zapisuje stanje TS M.

22.3. Generiranje jezika Turingovim strojem

— struktura TS za generiranje jezika
— prihvaćanje jezika generiranog TS
— generiranje jezika prihvaćenog TS
— jednostavni i složeni TS

Za generiranje jezika koristi se TS s višestrukim trakama. Jedna traka je izlazna. Znak se na izlaznu traku zapiše trajno i poslije ga nije moguće mijenjati. Glava za čitanje i pisanje se miče isključivo desno. Nizovi jezika se ispisuju na izlaznu traku (odvajaju se #) i čine jezik kojeg generira TS M (označava se s G(M)). TS generira klasu rekurzivno prebrojivih jezika.

Za TS M_1 koji generira $G(M_1)$ moguće je izgraditi TS M_2 koji prihvaća $L(M_2) = G(M_1)$. TS M_2 ima jednu traku više od TS M_1 . Dodatna traka je ulazna traka i na nju se zapiše niz koji se ispituje. TS M_2 generira niz i ako je niz istovjetan zadanom prihvaća se. Inače, generira se sljedeći niz.

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 ili **jednostavan TS** koji generira rekurzivni jezik, ili **složeni TS** koji generira rekurzivno prebrojiv jezik. Za rekurzivne jezike jednostavni TS ispisuje sve nizove na radnu traku. Nakon ispisa niza w_i, TS M_1 simulira rad M_2 i provjeri novi niz. Ako je niz prihvatljiv, M_1 ga kopira s radne trake na izlaznu. Ako je jezik rekurzivno prebrojiv postoji mogućnost da za neki niz TS M_2 nikad ne stane. Zbog toga se ne smije dozvoliti neograničen broj prijelaza tijekom simulacije TS M_2. Složeni TS M_1 generira par cijelih brojeva (i,j), a zatim M_1 simulira rad M_2 za i-ti niz primjenjujući najviše j prijelaza.

22.4. Istovjetnost rekurzivnog jezika i kanonskog slijeda

— definicija kanonskog slijeda
— istovjetnost RekJ i kanonskog slijeda

U kanonskom slijedu kraći nizovi su ispred duljih. Redoslijed nizova iste duljine se određuje na temelju njihove numeričke vrijednosti (Baza za računanje se određuje na temelju $|\Sigma|$).

Rekurzivne jezike je moguće generirati kanonskim slijedom. Ako je L(M) rekurzivan jezik tada je za generiranje jezika moguće koristiti jednostavan TS koji na izlaznu traku generira nizove onim redoslijedom kojim se ti nizovi generiraju na radnu traku.

Nije moguće izgraditi TS za opći slučaj konačnog jezika, ali za bilo koji točno određeni konačni jezik je moguće izgraditi zasebni TS koji ga prihvaća i koji uvijek stane za bilo koji ulazni niz.

23. GRAMATIKA NEOGRANIČENIH PRODUKCIJA

23.1. Formalna specifikacija gramatike neograničenih produkcija

- oblik produkcije i vrsta gramatike
- definiranje jezika generiranog gramatikom

Neograničene produkcije su oblika $\alpha \rightarrow \beta$(α, β su nizovi završnih i nezavršnih znakova; $\alpha \neq \varepsilon$). Naziva se još i **gramatika tipa 0.**

Gramatika neograničenih produkcija je zadana četvorkom G = (V, T, P, S). Ta produkciju $\alpha \rightarrow \beta$ definira se relacija $\Rightarrow$ na slijedeći način: $\gamma\alpha\delta \Rightarrow \gamma\beta\delta$. Refleksivno i tranzitivno okruženje relacije $\Rightarrow$ je $\overset{*}{\Rightarrow}$

Gramatika G generira jezik $L(G) = \left\{ w \middle| w \in T^*;\ S \overset{*}{\Rightarrow} w \right\}$ koji pripada klasi rekurzivno prebrojivih jezika.

23.2. Konstrukcija TS iz GNP

- gramatika, jezik i TS
- postupak rada automata
- simulacija gramatike G

Gramatika neograničenih produkcija generira rekurzivno prebrojiv jezik. Jezik L(G) koji generira gramatika G je rekurzivno prebrojiv ako i samo ako postoji TS koji prihvaća jezik L(M)=L(G).

Gradi se nedeterministički TS (NTS) s dvije trake koji simulira rad gramatike G na način da se na prvu traku zapiše niz w, na drugu traku zapiše se početni nezavršni znak gramatike S. Tijekom simulacije, na drugu traku se ispisuju nizovi α, koje generira gramatika G. Ako je niz a jednak nizu w, tada TS promijeni stanje u prihvatljivo i prihvati niz.

Simulacija gramatike G se izvodi na način da TS nedeterministički izabere mjesto i u nizu α druge trake. Zbog nedeterminističkog pristupa istodobno se izvodi simulacija za sve vrijednosti $1 \le i \le |\alpha|$. Za svaki od $|\alpha|$ izbora TS nedeterministički izabere produkciju $\beta \rightarrow \gamma$ (Zbog nedeterminizma simulacija se istodobno izvodi za sve produkcije gramatike G). Po svim i $\beta \rightarrow \gamma$, ako je β na mjestu i zamijeni se s γ. Niz generiran na drugoj traci uspoređuje se s nizom w koji je zapisan na prvoj traci. Ako su nizovi jednaki, TS se zaustavlja i prihvaća w, ako nisu jednaki, TS nastavlja dalje od početka ove simulacije.

23.3. Konstrukcija GNP iz TS

- pristup definiranju gramatike
- veza TS i gramatike

Ako TS M prihvaća RPJ L(M), tada postoji GNP G koja generira jezik L(G)=L(M). Gradi se gramatika G=(V,T,P,S) koja simulira rad TS M = (Q, Σ, Γ, δ, q_0, B, F) na način da gramatika G generira redom međunizove znakova koji predstavljaju konfiguracije TS M. Nezavršni znak q u generiranom međunizu je oznaka stanja TS M. Početna konfiguracija TS M se simulira međunizom u kojem su nezavršni znakovi oblika $[a_i,a_i]$ ($a_i \in \Sigma$). Prva komponenta čuva znakove niza tijekom simulacije na temelju kojih G generira niz nezavršnih znakova ako i samo ako TS prihvaća taj isti niz. Druga komponenta predstavlja znakove trake.

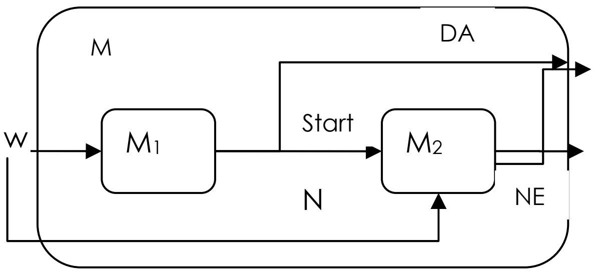
24. SVOJSTVA REKURZIVNIH JEZIKA

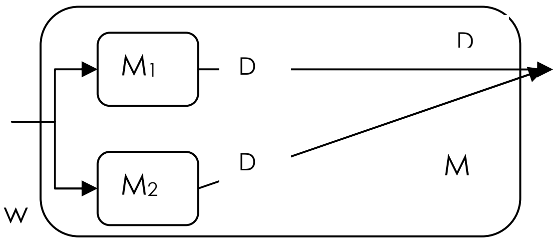
24.1. Zatvorenost RekJ i RPJ obzirom na uniju

- unija rekurzivnih jezika
- unija rekurzivno prebrojivih jezika

Unija rekurzivnih jezika je rekurzivni jezik.

Neka TS M_1 prihvaća RekJ L_1, a TS M_2 prihvaća RekJ L_2. TS uvijek stane za bilo koji niz rekurzivnog jezika. Rad TS M_1 i TS M_2 je moguće slijedno simulirati.
TS M prihvati niz ako i samo ako niz w prihvati TS M_1 ili TS M_2. TS M uvijek stane za bilo koji ulazni niz, jezik $L(M_1) \cup L(M_2)$ je rekurzivan.



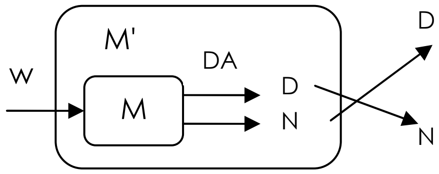


Unija rekurzivno prebrojivih jezika je rekurzivno prebrojiv jezik.
Za rekurzivno prebrojive jezike se koristi paralelna simulacija jer kod rekurzivno prebrojivih jezika postoje neki nizovi w za koje TS neće nikada stati. Gradi se poseban TS koji na zasebnim trakama istodobno simulira rad TS M_1 i TS M_2. Ako bilo TS M_1 ili TS M_2 stane i prihvati niz, tada to napravi i TS M.

24.2. Zatvorenost RekJ i RPJ obzirom na komplement

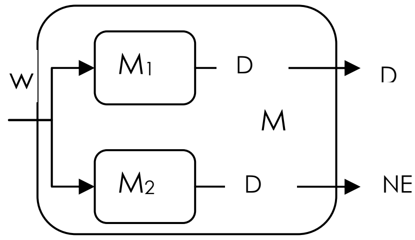
- komplement rekurzivnih jezika
- komplement rekurzivno prebrojivih jezika

Komplement rekurzivnog jezika je rekurzivan jezik.



Ako je L rekurzivan jezik tada postoji TS M koji ga prihvaća i koji uvijek stane za bilo koji w. Za prihvaćanje komplementa rekurzivnog jezika $L'=L^c$ gradi se TS M' koji simulira rad TS M. Ako TS M uđu u prihvatljivo stanje i stane, onda TS M' stane i ne prihvati niz. Ako TS M stane i odbaci niz, tada TS M' stane i prihvati niz. Budući da TS M' stane za bilo koji ulazni niz, komplement rekurzivnog jezika je rekurzivan jezik.

Ako su jezik L_1 i njegov komplement $L_2=L_1^C$ rekurzivno prebrojivi, onda su oba jezika rekurzivna. Neka TS M_1 prihvaća RPJ L_1 i TS M_2 koji prihvaća RPJ L_2 ($L_2=L_1^C$). Za svaki niz w vrijedi ili da je član L_1 ili L_2. TS M uvijek stane i jednoznačno odluči da li se niz prihvaća. Budući da TS uvijek stane L_1 i L_2 su rekurzivni.



25. IZRAČUNLJIVOST I ODLUČIVOST

25.1. Izračunljivost

- Definicija izračunljivosti
- Church-Turingova hipoteza
- Problem kodiranja: RAM, dijagonalni jezik (???)

Problem je **izračunljiv** ako postoji automat koji „mehaničkim“ postupkom korak po korak rješava zadani problem.

Church-Turingova hipoteza određuje da se izračunljive funkcije poistovjećuju s klasom parcijalno rekurzivnih funkcija. Parcijalno rekurzivne funkcije su izračunljive, jer je za njihovo računanje moguće izgraditi TS koji i računa korak po korak primjenom zadanih funkcija prijelaza.

Apstraktno računalo se simulira primjenom TS s višestrukim trakama. Prva traka se koristi za spremanje adresa i sadržaja memorijskih ćelija. Adrese i sadržaji memorijskih ćelija se spremaju na memorijsku traku na način: $\#0{*}v_0\#1{*}v_1\#10{*}v_2\#\ldots\#k{*}v_k$.

Gradnja dijagonalnog jezika L_d započinje kodiranjem funkcija prijelaza proizvoljnog TS. Za kodiranje se koristi model TS: $M=(Q, \{0,1\}, \{0,1,B\}, \delta, q_1, B, \{q_2\})$

25.2. Odlučivost

- Definicija odlučivosti
- Univerzalni TS i jezik

Rekurzivni jezici su **odlučivi** jer ih prihvaćaju TS koji uvijek stanu i odluče o prihvaćanju ili neprihvaćanju niza. Za rekurzivno prebrojive jezike ne postoji TS koji uvijek stane, rekurzivno prebrojivi jezici nisu **odlučivi**.

Univerzalni jezik L_U je primjer rekurzivno prebrojivog jezika (izračunljivog jezika koji **nije** odlučiv). Univerzalni TS ima tri trake. Prva traka je ulazna traka i na nju se zapiše niz <M,w> koji se kodira na način: 111kod$_1$11kod$_2$...111w (kod$_p$ → kod funkcije prijelaza TS M). Sadržaj druge trake simulira sadržaj trake TS M, dok se na treću traku zapisuje stanje q_i TS M nizom 0^i. Univerzalni TS M_U prihvaća **univerzalni jezik** L_U koji se definira: L_U= { <M,w> | TS M prihvaća niz w}.

26. KONTEKSTNO OVISNI JEZICI

26.1. Definicija kontekstno ovisnog jezika i gramatike

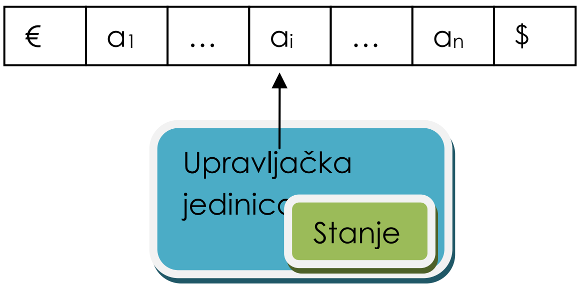
- kontekstno ovisni jezik i gramatika
- oblik produkcija
- primjer ?????

Jezik je kontekstno ovisan ako i samo ako postoji kontekstno ovisna gramatika koja ga generira. Time je definirana istovjetnost kontekstno ovisne gramatike i kontekstno ovisnih jezika.

Kontekstno ovisna gramatika ima produkcije oblika $\alpha \rightarrow \beta$. Nužno je da je broj znakova desne strane produkcije β veći ili jednak broju znakova lijeve strane produkcije α; $|\beta| \geq |\alpha|$; $\alpha \mathrel{!}= \varepsilon$.

26.2. Linearno ovisni automat

- model LOA
- formalna definicija LOA
- naziv LOA



Linearno ograničen automat (LOA) je nedeterministički TS koji koristi samo onaj dio trake na kojem je zapisan ulazni niz w. LOA se formalno zadaje sedmorkom loa= $(Q, \Sigma, \Gamma, \delta, q_0, €, \$, F)$.

Ako je duljina radne trake TS M ograničena za bilo koji niz w linearnom funkcijom f tako da TS M koristi najviše $f(|w|)$ ćelija, onda je moguće izgraditi istovjetni TS M' koji koristi samo onaj dio trake na koji je zapisan ulazni niz w. Zbog tog svojstva se koristi naziv LOA.

26.3. Konstrukcija linearno ovisnog automata iz KOG

- postupak izgradnje
- konstrukcija LOA iz KOG

LOA M koristi dva traga ulazne trake. U gornji trag LOA M zapiše niz €w$, a na početak donjeg traka LOA M zapiše početni nezavršni znak S gramatike G. Za prazni niz ε LOA odmah stane i ne prihvati niz. Za niz $w!=\varepsilon$ koristi se postupak simulacije gramatike tijekom koje se na donji trag ulazne trake generiraju nizovi primjenom produkcija. Nedeterministički izbori generiraju sve nizove jezika L(G), LOA M prihvaća niz završnih znakova w ako i samo ako gramatika generira niz w.

Tijekom rada LOA M ne koristi veći broj ćelija od duljine niza w. Za produkcije KOG vrijedi $\alpha \rightarrow \beta$; $|\beta| \geq |\alpha|$ pa niti jedan međuniz a na donjem tragu ulazne trake nije dulji od niza w.

27. SVOJSTVA KONTEKSTNO OVISNIH JEZIKA

27.1. Unija, nadovezivanje i Kleene

- zatvorenost KOJ po uniji
- zatvorenost KOJ po nadovezivanju
- zatvorenost KOJ po Kleene

Unija KOJ jest KOJ. Neka KOG $G_1=(V_1, T_1, P_1, S_1)$ i $G_2=(V_2, T_2, P_2, S_2)$ generiraju KOJ $L(G_1)$ i $L(G_2)$, $V_1 \cap V_2=\emptyset$. Gradimo $G_3=(V_3, T_3, P_3, S_3)$ koja generira $L(G_3)=L(G_1) \cup L(G_2)$. Sve produkcije sadrže barem jedan nezavršni znak na lijevoj strani i vrijedi $V_1 \cap V_2=\emptyset$. Tijekom generiranja bilo kojeg niza w primjenom gramatike G_3 primjenjuju se produkcije samo jedne gramatike (ili G_1 ili G_2).

Nadovezivanje KOJ jest KOJ. Neka KOG $G_1=(V_1, T_1, P_1, S_1)$ i $G_2=(V_2, T_2, P_2, S_2)$ generiraju KOJ $L(G_1)$ i $L(G_2)$, $V_1 \cap V_2 \neq \emptyset$. Gradimo $G_4=(V_4, T_4, P_4, S_4)$ koja generira $L(G_4)= L(G_1) L(G_2)$. G_4 generira nizove $\gamma\delta$ na način $S_4 \underset{G_4}{\Rightarrow} S_1 S_2 \underset{G_1}{\Rightarrow} * \gamma S_2 \underset{G_2}{\Rightarrow} * \gamma\delta$.

KOJ su zatvoreni obzirom na Kleenee L+. Neka KOG $G_1=(V_1, T_1, P_1, S_1)$ generira KOJ $L(G_1)$. Gradimo $G_5=(V_5, T_5, P_5, S_5)$ koja generira $L(G_5)=L(G_1)+$. Budući da se generiraju jedan do drugog međunizovi primjenom iste gramatike, potrebno je izgraditi dodatnu gramatiku G' koja se gradi na temelju gramatike G tada da se $A \in V$ zamijene sa A' tako da vrijedi $V \cap V'=\emptyset$. Gramatika G_5 se konstruira na način: $V_5=V \cup V' \cup \{S_5, S_5'\}$; $T_5=T$;U skup produkcija $P_5=P \cup P'$ dodaju se produkcije: $S_5 \rightarrow SS_5' | S$; $S_5' \rightarrow S' S_5 | S'$.

27.2. Presjek i komplement

- zatvorenost KOJ po presjeku
- zatvorenost KOJ po komplementu

Presjek KOJ jest KOJ. Neka LOA M_1 prihvaća jezik $L(M_1)$, a LOA M_2 prihvaća jezik L(M2). LOA M_3 prihvaća jezik $L(M_3)=L(M_1) \cap L(M_2)$ i ima dva traga ulazne trake. U oba traga zapisuje se isti niz završnih znakova w. LOA M_3 simulira rad LOA M1 na gornjem tragu ulazne trake, dok se rad LOA M_2 simulira na donjem tragu ulazne trake. Niz se prihvaća ako i samo ako tijekom simulacije LOA M_1 i LOA M_2 prihvate niz.

Komplement determinističkog KOJ jest DKOJ. Neka DLOA M prihvaća DKOJ L(M). Za bilo koji DLOA moguće je izgraditi istovjetni DLOA M' koji uvijek stane za bilo koji ulazni niz. Budući da DLOA koristi isključivo ćelije trake na kojima je zapisan ulazni niz w, moguće je izračunati maksimalan broj različitih konfiguracija DLOA na slijedeći način : $s(n+2)t^n$. $s=|Q|$, $n=|w|$; $t=|\Gamma|$.

27.3. Odlučivost kontekstno ovisnih jezika

- svojstvo odlučivosti KOJ

Bilo koji KOJ jest rekurzivni jezik. TS koji uvijek stane za bilo koji ulazni niz izvodi sljedeći algoritam prihvaćanja KOJa. Algoritam prihvaćanja jezika zasniva se na gradnji USMJERENOG GRAFA.

Čvorovi grafa su nizovi završnih i nezavršnih znakova gramatike a koji su jednako dugački ili kraći od ulaznog niza w |a|≤ |w| . Vrijedi li relacija a => β čvorovi grafa a i β povezuju se usmjerenim grafom. Put od čvora S do čvora w postoji ako i samo ako gramatika G generira w.

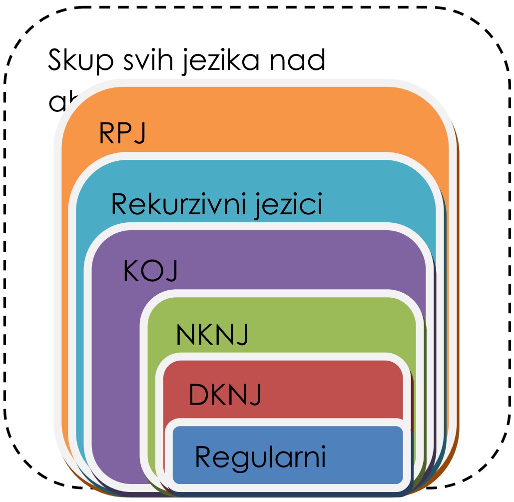
Budući da je konačan broj nizova a za koje vrijedi |a|≤ |w| , TS u konačnom broju koraka izgradi usmjereni graf i pretraži sve putove usmjerenog grafa. TS se uvijek zaustavi za bilo koji ulazni niz i odluči prihvaćanju niza, što dokazuje da su kontekstno ovisni jezici **REKURZIVNI** tj. **ODLUČIVI**.

28. STRUKTURNA SLOŽENOST JEZIKA

28.1. Klase i hijerarhija jezika

- klase jezika
- hijerarhija Chomskog
- definicije istovjetnosti

Neka su A i B dvije različite klase jezika. Ako je klasa A pravi podskup B, onda za te dvije klase vrijedi: automat koji prihvaća jezik iz klase A jednostavnije je strukture od automata koji prihvaća jezike iz klase B. Produkcije gramatike koja generira jezike iz klase A su jednostavnije od produkcija gramatike koja generira jezike iz klase B.



Chomskyjeva hijerarhija jezika

28.2. Hijerarhija gramatika i automata

- jezici i istovjetnosti
- hijerarhija gramatika i automata

— oblici produkcija i definicije automata

Hijerarhija gramatika i automata temelji se na:

- istovjetnost RJ konačnih automata i RG
- istovjetnost KNJ, KNG, i PA
- istovjetnost KOJ, KOG i linearno ograničenog automata
- istovjetnost RPJ, gramatike ograničenih produkcija i Turingovog stroja.

Oblici produkcija:

- Za gramatiku ograničenih produkcija G_0: $\alpha \rightarrow \beta$ $\alpha, \beta \in (T \cup V)^*$, $\alpha \neq \varepsilon$.
- Za KOG G_1: $\alpha \rightarrow \beta$, $|\alpha| \leq |\beta|$, $\alpha, \beta \in (T \cup V)^*$, $\alpha \neq \varepsilon$.
- Za KNG G_2: $A \rightarrow \alpha$, $A \in V$, $\alpha \in (T \cup V)^*$.
- Za RG G_3: $A \rightarrow wB$ i $A \rightarrow w$ ili $A \rightarrow Bw$ i $A \rightarrow w$, $A, B \in V$, $w \in T^*$.

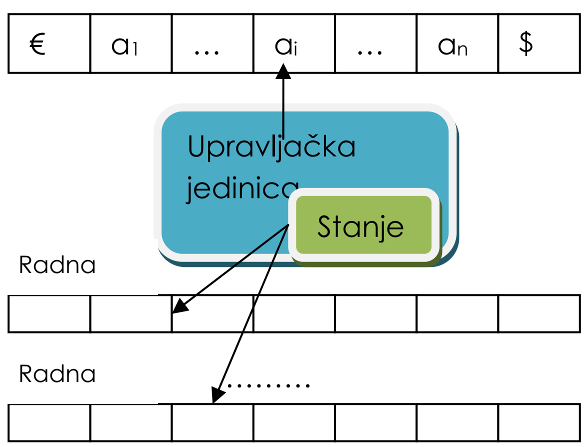
Definicije automata:

- Konačni automat: $M_3=(Q, \Sigma, \delta, q0, F)$
- Potisni automat: $M_2=(Q, \Sigma, \Gamma, \delta, q0, Z0, F)$
- Linearno ograničeni automat: $M_1=(Q, \Sigma, \Gamma, \delta, q0, c, \$, F)$
- Turingov stroj: $M_0=(Q, \Sigma, \Gamma, \delta, q0, B, F)$

29. SLOŽENOST PRIHVAĆANJA JEZIKA

29.1. Prostorna složenost

— korištenje modela TS
— model za prostornu složenost
— definicija prostorne složenosti



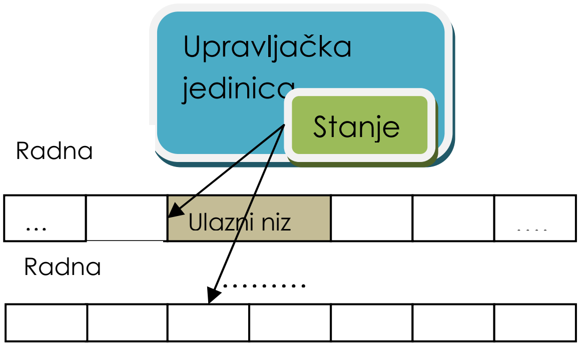
Za ocjenu PROSTORNE složenosti prihvaćanja jezika koristi se NEIZRAVNI deterministički TS sa k polubeskonačnih traka.

Ulazna traka sadrži niz koji se ispituje i moguće ju je samo čitati. PROSTORNA složenost se određuje na temelju samo jedne radne trake, i to one radne trake na kojoj TS koristi najviše ćelija. Uzima u obzir samo broj ćelija radnih traka, a ne broj ćelija ulazne trake.

Jezik L=L(M) koji prihvaća TS M jest prostorne složenosti S(n).

29.2. Vremenska složenost

— model za vremensku složenost
— definicija vremenske složenosti



TS ima k dvostrano beskonačnih traka. Jedna radna traka jest ulazna i na njoj je zapisan ulazni niz w duljine n. Sve trake je moguće čitati i pisati.

VREMENSKA složenost računa se na temelju broja pomaka glave za čitanje i pisanje. Jezik L=L(M) koji prihvaća TS M je vremenske složenosti T(n).
Budući da se čitaju svi znakovi niza, vremenska složenost je max $(n+1, \lceil T(n) \rceil)$, što se kraće zapisuje T(n).

29.3. Broj traka i složenost

— svojstva složenosti
— broj traka i prostorna složenost
— broj traka i vremenska složenost

Broj traka nema utjecaja na prostornu složenost, već samo na vremensku složenost. Vremenska složenost se povećava smanjivanjem broja traka.

Broj traka i prostorna složenost: Ako TS M_1 s k radnih traka prostorne složenosti S(n) prihvaća jezik $L(M_1)$, onda postoji TS M_2 s jednom radnom trakom koji je jednake prostorne složenosti S(n) i koji prihvaća jezik $L(M_2)=L(M_1)$.

Broj traka i vremenska složenost: Ako TS M_1 s k traka vremenske složenosti T(n) prihvaća jezik $L(M_1)$, onda postoji TS M_2 s jednom trakom koji je vremenske složenosti $T_2(n)$ i koji prihvaća jezik $L(M_2)=L(M_1)$.

29.4. Sažimanje prostora i ubrzanje vremena

— sažimanje prostora
— ubrzanje vremena

Sažimanje prostora za konstantni faktor: Ako TS M_1 sa k radnih traka prostorne složenosti S(n) prihvaća jezik $L(M_1)$, onda za bilo koju konstantu c>0 postoji TS M_2 prostorne složenosti cS(n) koji prihvaća jezik $L(M_2)=L(M_1)$.

Ubrzanje vremena za konstantni faktor : Ako TS M_1 sa k traka vremenske složenosti T(n) prihvaća jezik $L(M_1)$, onda za bilo koju konstantu c>0 postoji TS M_2 sa k traka vremenske složenosti cT(n) koji prihvaća jezik $L(M_2)=L(M_1)$, gdje je k>1 i $\inf_{n\to\infty} T(n)/n=\infty$. Funkcija $\inf_{n\to\infty} f(n)$ je najveća donja granica funkcije f(n) kada n teži u beskonačnost.

30. KLASE JEZIKA PO SLOŽENOSTI

30.1. Model složenosti i odnosi među klasama

— model složenosti i TS
— osnovne klase složenosti
— odnosi među klasama složenosti

Nedeterministički TS je *prostorne složenosti S(n)* ako za niti jedan niz duljine **n**, niti jedan mogući slijed prijelaza, na niti jednoj od radnih traka ne koristi više od S(n) ćelija.

NTS je *vremenske složenosti T(n)* ako za niti jedan niz duljine **n**, niti jedan mogući slijed prijelaza ne pomakne glavu više od T(n) puta.

4 osnovne klase složenosti: klasa DSPACE(S(n)), klasa NSPACE(S(n)), klasa DTIME(T(n)), klasa NTIME(T(n)).

Odnosi među klasama složenosti:

- ako je jezik L u klasi DTIME(f(n)), onda je jezik L u klasi DSPACE(f(n))

- ako je jezik L u klasi DSPACE(f(n)), i ako za f-ju f(n) vrijedi $f(n) \geq \log_2 n$, onda je jezik L u klasi DTIME($C^{f(n)}$). Vrijednost C ovisi o jeziku L.

- ako je jezik L u klasi NTIME(f(n)) i ako je f-ja $f(n) \geq \log_2 n$ onda je jezik L u klasi DTIME($C^{f(n)}$). Vrijednost C ovisi o jeziku L.

- ako je jezik L u klasi NSPACE(f(n)) i ako je f-ja $f(n) \geq \log_2 n$ potpuno prostorno izgradiva, onda je jezik L u klasi DSPACE($f^2(n)$).

30.2. Vremenska složenost DTS i NTS i prostorna izgradivost

— opis NTS stablom
— vremenska složenost i stablo
— prostorna izgradivost S(n)
— svojstva hijerarhije

(nekakva slika vremenskog stabla).

Čvorovi grafa su konfiguracije TS. Korijen je početna konfiguracija.

Vremenska složenost determinističkog TS proporcionalna je vremenu obilaženja stabla u cilju pronalaženja barem jedne konfiguracije u listovima stabla koja prihvaća zadani niz.

Prostorna izgradivost S(n): Neka je TS M prostorne složenosti S(n). F-ja S(n) je prostorno izgradiva ako za bilo koji **n** postoji barem jedan niz duljine **n** za koji TS M koristi svih S(n) ćelija.

Svojstva hijerarhije: *beskonačnost*, za potpuno prostorno i vremenski izgradive funkcije hijerarhija je *neprekinuta*, suprotno *nisu neprekinute*, moguće je izgraditi jezik za koji *ne postoji* optimalni TS koji ga prihvaća u minimalnom vremenu ili prostoru, za uniju klasa jezika moguće je odrediti funkciju složenosti tako da obuhvati sve jezike iz unije.

30.3. Klasa jezika polinomne složenosti

- odlučivost i jezici polinomne složenosti
- označivanje
- odnosi determinističke i nedeterminističke polinomne složenosti

Značajna klasa jezika je ona DETERMINISTIČKE POLINOMNE SLOŽENOSTI. Klasa jezika polinomne složenosti označava se znakom P.

Klasa jezika P definira se na sljedeći način: $P = \bigcup_{i \geq 1} DTIME(n^i)$

Na temelju svojstva unije klasa jezika, u klasi jezika P su svi oni jezici za koje postoji deterministički TS polinomne vremenske složenosti n, n^2, n^3, n^4,...

Mnogi značajni jezici nisu u klasi P, ali za njih postoji nedeterministički TS polinomne vremenske složenosti. Klasa jezika nedeterminističke polinomne složenosti označava se oznakom NP:

$$NP = \bigcup_{i \geq 1} NTIME(n^i)$$