# Regularni jezici

1. **Definirati:**
   **a) formalni automat,**
   **b) formalnu gramatiku.**

a) Formalni automat je model diskretnog matematičkog sustava koji čitanjem znak po znak odlučuje je li pročitani niz element zadanog jezika. *(str. 12 u knjizi)*

b) Formalna gramatika je model matematičkog sustava koji primjenom skupa produkcija gradi, odnosno generira nizove znakova. *(str. 12)*

2. Navesti i objasniti načine programskog ostvarenja funkcije prijelaza.

Ostvaruje se na dva načina: vektorski i listom. Vektorski se definira za svako stanje konačnog automata jedan vektor. U vektoru onoliko elemenata kolko je različitih ulaznih znakova. Učinkovito korištenje memorije postize se listom, ali duze traje računanje novog stanja.

3. Formalno definirati DKA i pripadnu funkciju $\hat{\delta}$.

Uređena petorka DKA=(Q, ∑, δ, q0, F)

Q- konacan skup stanja, ∑- konacan skup ul znakova, δ- funkcija prijelaza Qx∑->Q , q0- pocetno stanje, F- skup prihvatljivih stanja

(1) $\hat{\delta}(q, \epsilon) = q$ , gdje je $\epsilon$ prazni niz

(2) za bilo koji niz ulaznih znakova $w$ i za bilo koji ulazni znak $a$ vrijedi:
$\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$ gdje je $w \in \eta^*$ i $a \in \eta$

4. Opisati jednostavan algoritam pronalaženja istovjetnih stanja (1. algoritam).

Provjeravanje dva uvjeta. Uvijet podudarnosti: stanja p i q moraju btii oba prihvatljiva ili neprihvatljiva. Uvijet napredovanja: za bilo koji ulazni zank a vrijdi da su stanja δ(p,a) i δ(q,a) istovjetna.

5. Navesti definiciju nedohvatljivog stanja i napisati algoritam za pronalaženje nedohvatljivih stanja

Stanje p je nedohvatljivo ako ne postoji niti jedan niz w € ∑*, za koje vrijedi da je p= δ(q0, w).Njihovim odbacivanjem dobije se istovjetni atuomat sa manjim brojem stanja.

6. **Opisati postupak dobivanja DKA iz zadanog NKA.**

DKA $M'=(Q', \Sigma', \delta', q_0', F')$ prihvaća isti jezik kao i NKA $M=(Q, \Sigma, \delta, q_0, F)$ ako je zadovoljeno:

1) Skup stanja DKA je $Q'=2^Q$, tj. skup svih podskupova skupa stanja NKA $Q$. Stanja su označena uglatim zagradama $[p_0, p_1, \ldots, p_j] \in Q'$, gdje su $p_k \in Q$.

2) Skup prihvatljivih stanja DKA $F'$ jest skup svih stanja $[p_0, p_1, \ldots, p_j]$ gdje je barem jedan $p_k \in F$.

3) Početno stanje DKA jest $q0'=[q0]$.

4) Funkcija prijelaza DKA jest $\delta'([p_0, p_1, \ldots, p_m], a) = [r_0, r_1, \ldots, r_j]$ ako i samo ako je $\delta(\{p_0, p_1, \ldots, p_m\}, a) = \{r_0, r_1, \ldots, r_j\}$, gdje je $a \in \Sigma$. *(str. 31)*

7. Opisati postupak pretvorbe ε-NKA u NKA.

Q'=Q, q0=q0', F'=F U {q0} ako e-okruzenje(q0) sadrzi barem jedno stanje skupa F, inače F'=F

δ' (q,a)=δ (q,a), Va∑ i Vq€Q.

8. Dokazati ekvivalenciju ε-NKA i DKA koji je dobiven pretvorbom ε-NKA→NKA→DKA.

**Formalno definirati:**

**a) istovjetnost stanja DKA,**

**b) istovjetnost DKA.**

a) Stanje $p$ DKA $M=(Q, \Sigma, \delta, q_0, F)$ je istovjetno stanju $p'$ DKA $M'=(Q', \Sigma', \delta', q_0', F')$ ako i samo ako DKA $M$ u stanju $p$ prihvaća isti skup nizova kao i DKA $M'$ u stanju $p'$. Za bilo koji niz $w$ skupa $\Sigma^*$ vrijedi $(\delta(p,w)\in F) \wedge (\delta(p',w)\in F')$ ili $(\delta(p,w)\notin F) \wedge (\delta(p',w)\notin F')$. *(str. 21)*

9. b) DKA $M$ i $N$ su istovjetni ako i samo ako su istovjetna njihova početna stanja. *(str. 22)*

10. Definirati ekvivalentnost stanja determinističkih konačnih automata. Objasniti postupak minimizacije determinističkog konačnog automata pronalaženjem neekvivalentnih stanja (Algoritam 3).

Stanje $p$ DKA $M=(Q, \Sigma, \delta, q_0, F)$ je istovjetno stanju $p'$ DKA $M'=(Q', \Sigma, \delta', q_0', F')$ ako i samo ako DKA $M$ u stanju $p$ prihvaća isti skup nizova kao i DKA $M'$ u stanju $p'$. Za bilo koji niz $w$ skupa $\Sigma^*$ vrijedi $\delta(p, w)\in F \wedge \delta'(p', w)\in F'$ ili $\delta(p, w)\notin F \wedge \delta'(p', w)\notin F'$.

```
Označi sve parove (p,q) za koje vrijedi da je p∈F i q∉F;

za (Bilo koji par različitih stanja (p,q)∈(FxF) ili (p,q)∈(FxF))
{
      ako (Za neki znak a par (δ(p,a),δ(q,a)) jest označen)
      {
         Označi (p,q);

         Rekurzivno označi sve neoznačene parove u listi koja je pridružena paru
         (p,q) i sve ostale parove u listama koje su pridružene parovima
         označenim u ovom koraku;
      }
      inače
      {
          za (Svi znakovi a)
          {
               ako (δ(p,a)!= δ(q,a))
               Stavi (p,q) u listu koja je pridružena paru (δ(p,a),δ(q,a))
          }
      }
}
```

11. Opisati postupak konstrukcije Mealyevog iz Mooreovog automata.

Definira se kao bTM' (w)=TM (w), w ulazni niz, b izlaz Moorea za prazni niz b= λ(q0)
λ' (q,a)= λ (δ(q,a)), za sve q iz Q i za sve a iz ∑
oba automata imaju istu funkciju prijelaza δ

12. (3 boda) Formalno definirajte postupak konstrukcije Mooreovog automata M' iz zadanog Mealyevog automata M.

Q'=QxΔ gdje je [q,b]€Q', q€Q i b€Δ
q0'=[q0, b0], b0 proizvoljni element iz Δ
δ'([q,b], a)=[ δ(q,a), λ(q,a)], q€Q i b€Δ, a€∑
λ'([q,b])=b , q€Q i b€Δ

13. Dokazati da su regularni jezici zatvoreni s obzirom na operaciju komplementiranj

Neka je DKA M=(Q, ∑, δ, q0, F) prihvaca L(M). Za komplement jezika L(M)C moguce je izgraditi DKA M'=(Q, ∑, δ, q0, Q\F) koji prihvaca L(M')={w | δ(q0, w)€(Q\F) }={w | δ(q0, w)nije €F}= ∑*\{w | δ(q0, w)€ F)}= ∑*\L(M) = L(M)C .

14. Opisati postupak ispitivanja nepraznosti regularnih jezika.

Reg jezik jest neprazan ako i samo ako DKA M prihvaca niz z duljine manje od n, tj. |z|<n
Utvrdjivanjenepraznosti se prosirujealgor. Nedohvatljivih stanja. Ako je u skupu dohvatljivih barem jedno prihvatljivo stanje, onda je reg jezik neprazan.

15. Navesti svojstva ε-NKA dobivenog postupkom konstrukcije iz regularnog izraza.

Broj stanja e-NKA nikad nije veci od 2|r|, gjde je |r| broj znakova u reg izrazu r.
E_NKA ima samo jedno zavrsno stanje f. I vrijedi da je δ(f, a)= prazan skup
Skup δ(q,a) sadrzinajvise jedno stanje za ulazni znak a iz skupa ∑. Dok skup δ(q,e) sadrzinajvise dva stanja.

16. Navesti postupak konstrukcije NKA iz pojednostavljene gramatike.

Neka su produkcije gramatike $G=(V, T, P, S)$ oblika $A \to aB$ ili $A \to \varepsilon$, gdje su $A$ i $B$ nezavršni znakovi gramatike, a znak $a$ jest završni znak. Za zadanu gramatiku moguće je izgraditi NKA $M=(Q, \Sigma, \delta, q_0, F)$ za koji vrijedi $L(M)=L(G)$. NKA $M$ konstruira se na sljedeći način:

Skup ulaznih znakova NKA $\Sigma$ jednak je skupu završnih znakova gramatike $T$, tj. $\Sigma = T$;

Skup stanja NKA $Q$ jednak je skupu nezavršnih znakova gramatike $V$, tj. $Q = V$;

Početno stanje NKA $q_0$ jednako je početnom nezavršnom znaku gramatike $S$, tj. $q_0 = S$;

Na temelju produkcije $A \to aB$ gradi se sljedeći prijelaz:

$\delta(A, a) = \delta(A, a) \cup \{B\}$.

Skup prijelaza $\delta(A, a)$ proširuje se novim stanjem $B$. Početno su svi skupovi $\delta(A, a)$ prazni, tj. $\delta(A, a) = \varnothing$. Moguće je da više produkcija ima isti nezavršni znak na lijevoj strani i isti *završni znak* na desnoj strani, ali različite *nezavršne znakove* na desnoj strani. Na primjer, ako su zadane produkcije $A \to a\underline{B}$ i $A \to a\underline{C}$, onda je skup prijelaza iz stanja $A$ za ulazni znak $a$ jednak $\delta(A, a) = \{B, C\}$. Izgrađeni konačni automat nije nužno DKA, već može biti i NKA.

Ako je u gramatici produkcija $A \to \varepsilon$, onda je $A$ prihvatljivo stanje NKA, tj. $A \in F$.

16.1.

Opisati postupak pretvorbe desno-linearne gramatike u jednostavnu gramatiku pogodnu za konstrukciju NKA.

Za sve produkcije oblika:

$A \to w$, gdje je $w$ neprazni niz završnih znakova,

doda se jedan novi nezavršni znak, na primjer $[\varepsilon]$, i gradi se nova produkcija:

$[\varepsilon] \to \varepsilon$.

Sve produkcije oblika $A \to w$ zamijene se novim produkcijama oblika:

$A \to w[\varepsilon]$

Sve produkcije oblika:

$A \to a_1 \dots a_n B$, za $n>1$,

zamijene se produkcijama oblika:

$A \to a_1 [a_2 \dots a_n B]$
$[a_2 \dots a_n B] \to a_2 [a_3 \dots a_n B]$
$[a_3 \dots a_n B] \to a_3 [a_4 \dots a_n B]$
- - -
$[a_i \dots a_n B] \to a_i [a_{i+1} \dots a_n B]$, za $1 < i < n$
- - -
$[a_{n-1} a_n B] \to a_{n-1} [a_n B]$
$[a_n B] \to a_n B$

gdje su $[a_i \dots a_n B]$ novi nezavršni znakovi, $1 < i \leq n$.

Ako je nezavršni znak $B$ jedini znak desne strane produkcije:

$A \to B$,

onda se izuzmu sve produkcije koje imaju istu lijevu i desnu stranu:

$B \to B$.

Ostanu li produkcije koje imaju različite lijeve i desne strane $A \to B$, one se zamijene produkcijama:

$A \to y$

za sve kombinacije nezavršnih znakova $A$ i desnih strana produkcija $y$ za koje vrijedi:

$A \to B$ i $B \to y$.

17. Opisati postupak konstrukcije NKA iz desno linearne gramatike.

Jezik $L$ jest regularan ako i samo ako postoji desno-linearna gramatika $G_D$ koja generira jezik $L$, tj. $L=L(G_D)$, odnosno jezik $L$ jest regularan ako i samo ako postoji lijevo-linearna gramatika $G_L$ koja generira jezik $L$, tj. $L=L(G_L)$. Za bilo koju desno-linearnu ili lijevo-linearnu gramatiku $G$ moguće je izgraditi konačni automat $M$ koji prihvaća jezik koji generira zadana gramatika, tj. $L(M)=L(G)$. Lijevo-linearna ili desno-linearna gramatika preurede se tako da su sve produkcije oblika $A \to aB$ i $A \to \varepsilon$, a nakon toga primijeni se algoritam za gradnju NKA koji je prethodno opisan u ovom odjeljku.

18. Detaljno navesti algoritam za konstrukciju NKA iz lijevo linearne gramatike.

Neka je $G=(V, T, P, S)$ lijevo-linearna gramatika. $\varepsilon$-NKA $M'$ koji prihvaća jezik $L(M')=L(G)$ konstruira se na sljedeći način:

1) Izgradi se desno-linearna gramatika $G'=(V, T, P', S)$. Skup produkcija $P$ gramatike $G$ preuredi se tako da se desne strane produkcija napišu obrnutim redoslijedom:

$P'=\{ A \to \alpha^R \mid A \to \alpha \text{ jest u skupu } P \}$.

Gramatika $G'$ generira nizove završnih znakova koji su obrnuto napisani nizovi znakova koje generira gramatika $G$, tj. $L(G')=L(G)^R$.

2) Na temelju desno-linearne gramatike $G'$ konstruira se NKA $M$ koji prihvaća jezik $L(M)=L(G')=L(G)^R$.

3) Na temelju NKA $M$ izgradi se $\varepsilon$-NKA $M'$ koji prihvaća jezik $L(M')=L(M)^R=L(G')^R=L(G)$, tj. $\varepsilon$-NKA $M'$ prihvaća jezik $L(G)$ koji generira lijevo-linearna gramatika $G$.

$\varepsilon$-NKA $M'$ gradi se na sljedeći način:

NKA $M$ preuredi se tako da ima samo jedno prihvatljivo stanje. Ima li NKA više prihvatljivih stanja, definira se novo jedinstveno prihvatljivo stanje i definiraju se $\varepsilon$-prijelazi iz svih prijašnjih prihvatljivih stanja u novo prihvatljivo stanje. U skupu prihvatljivih stanja ostaje samo novo prihvatljivo stanje. Za početno stanje $\varepsilon$-NKA $M'$ uzima se prihvatljivo stanje NKA $M$, a za prihvatljivo stanje $\varepsilon$-NKA $M'$ uzima se početno stanje NKA $M$. Funkcije prijelaza NKA $M'$ grade se zamjenom smjera usmjerenih grana u dijagramu stanja. Izgrađeni $\varepsilon$-NKA $M'$ prihvaća jezik $L(M')$. U jeziku $L(M')$ su nizovi koji su napisani obrnutim redoslijedom od nizova jezika $L(M)$ koji prihvaća NKA $M$.

19. Formalno definirati nedeterministički konačni automat s $\varepsilon$ prijelazima ($\varepsilon$–NKA).

ε-NKA = ( Q, Σ, δ , q0 , F )
Q –konačan skup stanja
Σ - konačan skup ulaznih znakova
δ - funkcija prijelaza Q x ( Σ U {ε} ) --> 2^Q
q0 – početno stanje
F podskup Q – skup prihvatljivih stanja
Takoder se definira ε-OKRUŽENJE:
ε-OKRUŽENJE(q) = {p| a stanje p jest stanje q ili ε-NKA prijelazi iz stanja q u stanje p primjenom isključivo ε-prijelaza

20. Navesti algoritme koji se koriste u postupku odbacivanja beskorisnih znakova gramatike te navesti i objasniti redoslijed kojim se algoritmi izvode.

Algoritmi koji se koriste u postupku odbacivanja beskorisnih znakova su: algoritam odbacivanja beskorisnih znakova A ZATIM algoritam odbacivanja nedohvatljivih znakova. Primjena algoritama obrnutim redoslijedom ne će nužno rezultirati i odbacivanjem svih beskorisnih znakova...

21. Dokazati da su regularni jezici zatvoreni s obzirom na nadovezivanje

Regularni jezici zatvoreni su s obzirom na operaciju presjeka. Svojstvo zatvorenosti s obzirom na presjek slijedi izravno iz svojstva zatvorenosti s obzirom na uniju i komplement, što se dokazuje De Morganovim pravilom: $L \cap N = ((L \cap N)^c)^c = (L^c \cup N^c)^c$.

Ako su zadani DKA $M_1=(Q_1, \Sigma, \delta_1, q_1, F_1)$ i DKA $M_2=(Q_2, \Sigma, \delta_2, p_1, F_2)$, onda se DKA $M=(Q, \Sigma, \delta, q_0, F)$ koji prihvaća regularni jezik $L(M) = L(M_1) \cap L(M_2)$ gradi na sljedeći način:

1) $Q = Q_1 \times Q_2$, stanje $[q, p] \in Q$, gdje je $q \in Q_1$ i $p \in Q_2$,

2) $q_0 = [q_1, p_1]$,

3) $F = F_1 \times F_2$, stanje $[q, p] \in F$, gdje je $q \in F_1$ i $p \in F_2$,

4) $\delta([q, p], a) = [\delta_1(q, a), \delta_2(p, a)]$, $\forall q \in Q_1$, $\forall p \in Q_2$ i $\forall a \in \Sigma$.

22. Opisati postupak odbacivanja mrtvih znakova gramatike.

Ako nije moguće iz znaka X generirati niz završnih znakova, tj. Ako ne postoji postupak generiranja X =*=>wx, gdje je wxϵT* , onda je znak mrtav. Mrtvi znakovi se traže pomoću algoritna traženja živih znakova:
1. U listu živih znakova stave se lijeve strane produkcija, koje na desnoj strani nemaju nezavršnih znakova
2. Ako su na desnoj strani produkcije isključivo živi znakovi, onda se nezavršni znak lijeve strane produkcije doda u listu živih znakova
3. Ako nije moguće proširiti listu živih znakova, onda se algoritam zaustavlja. Svi znakovi koji nisu u listi živih znakova su mrtvi znakovi.

23. Opisati generator konačnog automata.

Ovisno o vrsti konačnog automata koji se konstruira, izgradi se odgovarajući program simulator. Simulator konačnog automata jest program koji slijedno čita znakove ulaznog spremnika . Nakon pročitanog znaka, simulator računa prijelaz u novo stanje na temelju podataka iz tablice prijelaza. Stanje nakon posljednje pročitanog znaka odlučuje da li se ulazni niz prihvaća. Želi li se konstruirati konačni automat za neki drugi jezik , potrebno je izgraditi novu tablicu prijelaza. Generator konačnog automata gradi tablicu na temelju zadanih regularnih izraza. Tablica prijelaza se ugradi u simulator.
( u pdf / word staviti slike sa stranica 50 i 51 !!!)

24. Dokažite svojstvo zatvorenosti regularnih izraza s obzirom na presjek.

SVOJSTVO ZATVORENOSTI
Slijedi izravno iz zatvorenosti s obzirom na uniju i komplement, što se dokazuje DeMorganovim pravilom: L∩N=(( L∩N)c)c=( Lc∩Nc)c
Ako su zadani DKA M1 = ( Q1, Σ, δ1 , q1 , F1 ) i DKA M1 = ( Q2, Σ, δ2 , p1 , F2 ) onda se gradi
DKA M = ( Q, Σ, δ , q0 , F ) koji prihvaća L(M) = L(M1) ∩ L(M2) gradi na sljedeći način:
1. Q = Q1 x Q2, stanje [q, p] ϵ Q, gdje je qϵQ1 i p ϵQ2
2. Q0 = [q1, p1]
3. F = F1 x F2, stanje [q, p] ϵ F , gdje je qϵF1 i p ϵF2
4. δ ([q, p], a ) = [ δ1(q,a ) , δ2(p,a)] svaki q ϵ Q1 , svaki p ϵ Q2, svaki a ϵ Σ

25. Opisati algoritam za utvrđivanje beskonačnosti jezika *L(M)* kojega prihvaća deterministički konačni automat *M*.

*Nepraznost i beskonačnost regularnog jezika*. Neka DKA $M$ prihvaća jezik $L(M)$ i neka DKA $M$ ima $n$ stanja. Svojstvo napuhavanja primjenjuje se za dokazivanje sljedećih tvrdnji:

a) Regularni jezik $L(M)$ jest neprazan ako i samo ako DKA $M$ prihvaća niz $z$ duljine manje od $n$, tj. $|z|<n$.

Za utvrđivanje da li je jezik $L(M)$ koji prihvaća DKA $M$ neprazan, proširuje se algoritam određivanja nedohvatljivih stanja. Ako je u skupu dohvatljivih stanja barem jedno prihvatljivo stanje, onda je regularni jezik $L(M)$ neprazan.

b) Regularni jezik $L(M)$ jest beskonačan ako i samo ako $M$ prihvaća niz duljine $l$, gdje je $n \le l < 2n$.

Za utvrđivanje da li je jezik $L(M)$ beskonačan, promatra se dijagram stanja DKA $M$. Izuzimanjem svih neprihvatljivih stanja za koja ne postoji niti jedan slijed prijelaza u jedno od prihvatljivih stanja, dobije se DKA $M'$ koji prihvaća isti jezik $L(M)=L(M')$. Ako je u dobivenom dijagramu stanja DKA $M'$ barem jedna zatvorena petlja, onda je regularni jezik $L(M)$ beskonačan.

# Konteksno neovisni jezici

1. Navesti formalnu definiciju konteksno neovisne gramatike.

U lingvistici i računarstvu, **kontekstno neovisna gramatika (KNG)** je formalna gramatika u kojoj je svaka produkcija oblika

$V \rightarrow w$

gdje je V nezavršni znak a *w* niz znakova (string) koji se sastoji od završnih i/ili nezavršnih znakova. Termin "kontekstno neovisna" izražava činjenicu da će nezavršni znak V uvijek biti zamijenjen nizom znakova *w* neovisno o kontekstu u kojem se pojavljuje. Formalni jezik je kontekstno neovisni jezik ako postoji kontekstno neovisna gramatika koja ga generira.

2. Formalno opisati algoritam za izbacivanja mrtvih znakova iz konteksno-neovisne gramatike.

*Str 77.*

*a)* Ako su živi svi znakovi $X_1, X_2, ..., X_n$ desne strane produkcije:

$$A \rightarrow X_1 X_2 \dots X_n,$$

onda je živ i nezavršni znak $A$ lijeve strane produkcije.

Budući da nema mrtvih znakova, za bilo koji $X_i$ vrijedi $X_i \overset{*}{\Rightarrow} w_i$, $w_i \in T^*$. Nadalje, vrijedi da je $A \Rightarrow X_1 X_2 \dots X_n \overset{*}{\Rightarrow} w_1 w_2 \dots w_n$, gdje je $w_1 w_2 \dots w_n = w$ niz završnih znakova.

Algoritam traženja živih znakova izvodi se u tri koraka:

1) U listu živih znakova stave se lijeve strane produkcija koje na desnoj strani nemaju nezavršnih znakova.

2) Ako su na desnoj strani produkcije isključivo živi znakovi, onda se nezavršni znak lijeve strane produkcije doda u listu živih znakova.

3) Ako nije moguće proširiti listu živih znakova, onda se algoritam zaustavlja. Svi znakovi koji nisu u listi živih znakova su mrtvi znakovi.

*3.*

Navesti i objasniti algoritam za izbacivanje ε-produkcija iz kontekstno-neovisne gramatike.

*str 82.*

Zadana je gramatika $G$. Želi se izgraditi istovjetna gramatika $G'$ koja nema $\varepsilon$-produkcija. Algoritam odbacivanja $\varepsilon$-produkcija izvodi se u dva osnovna koraka:

1) Pronađu se svi nezavršni znakovi koji generiraju prazni niz. *Prazni znakovi* su oni znakovi za koje vrijedi $A \overset{*}{\Rightarrow} \varepsilon$. Prazni znakovi traže se sljedećim iterativnim postupkom:

   U listu praznih znakova stave se lijeve strane svih $\varepsilon$-produkcija. Ako su svi znakovi desne strane produkcija zapisani u listu praznih znakova, onda se lista praznih znakova nadopuni lijevom stranom te produkcije. Algoritam se nastavlja sve dok je moguće proširiti listu praznih znakova novim nezavršnim znakom.

2) Skup produkcija gramatike $G'$ gradi se na sljedeći način. Ako je:

$$A \rightarrow X_1 X_2 \dots X_n$$

produkcija gramatike $G$, onda se u skup produkcija gramatike $G'$ dodaju produkcije oblika:

$$A \rightarrow \xi_1 \xi_2 \dots \xi_n.$$

Oznake $\xi_i$ poprimaju sljedeće vrijednosti:

*a)* Ako znak $X_i$ nije prazni znak, onda je oznaka $\xi_i$ jednaka $X_i$.

*b)* Ako je znak $X_i$ prazni znak, onda je oznaka $\xi_i$ jednaka $\varepsilon$ ili $X_i$.

Produkcije se grade na temelju svih mogućih kombinacija oznaka $\xi_1 \xi_2 \dots \xi_n$. Ako sve oznake $\xi_i$ poprime vrijednost $\varepsilon$, onda nastaje $\varepsilon$-produkcija i ona se ne dodaje u skup produkcija gramatike $G'$. Time se odbace sve $\varepsilon$-produkcije iz zadane gramatike $G$.

Navesti algoritam za izbacivanje jediničnih produkcija iz kontekstno neovisne gramatike.

*4.* *Str 83.*

Istovjetna gramatika $G'$ koja nema jediničnih produkcija gradi se na sljedeći način:

1) U skup produkcija gramatike $G'$ stave se sve produkcije gramatike $G$ koje nisu jedinične.

2) Neka se postupkom generiranja iz nezavršnog znaka $A$ dobije nezavršni znak $B$, tj. $A \stackrel{*}{\Rightarrow} B$, gdje su $A$ i $B$ nezavršni znakovi gramatike $G$. Za sve produkcije $B \to \alpha$ koje nisu jedinične grade se nove produkcije $A \to \alpha$.

5. Opisati postupak pretvorbe konteksno neovisne gramatike u Chomskyev normalni oblik.

Pretpostavi se da gramatika $G$ nema beskorisnih znakova, $\varepsilon$-produkcija i jediničnih produkcija. Algoritam pretvorbe produkcija u Chomskyjev normalni oblik izvodi se u tri koraka:

1) U skup produkcija $P'$ stave se sve produkcije koje su u Chomskyjevom normalnom obliku, odnosno produkcije oblika $A \to BC$ ili $A \to a$. U skup nezavršnih znakova $V'$ upišu se svi nezavršni znakovi.

2) Neka je produkcija gramatike $G$ oblika:

$$A \to X_1 X_2 \cdots X_m, \qquad A \in V,\ X_i \in T \text{ ili } X_i \in V,\ m \geq 2,\ 1 \leq i \leq m,$$

Ako je $X_i$ završni znak $a$, $a \in T$, onda se skup nezavršnih znakova proširi novim nezavršnim znakom $C_a$, $C_a \in V'$. Skup produkcija $P'$ proširuje se produkcijom:

$$C_a \to a,$$

koja je u Chomskyjevom normalnom obliku. Svi završni znakovi $a$ u produkciji $A \to X_1 X_2 \cdots X_m$ zamijene se nezavršnim znakom $C_a$. Definiranje novih nezavršnih znakova i novih produkcija nastavlja se za sve završne znakove na desnoj strani produkcije $A \to X_1 X_2 \cdots X_m$. Zamjenom svih završnih znakova nezavršnim znakovima, na desnoj su strani produkcije $A \to X_1 X_2 \cdots X_m$ isključivo nezavršni znakovi gramatike $G'$.

Pretvorbe se nastavlja za sve produkcije oblika $A \to X_1 X_2 \cdots X_m$.

3) Nakon što završi korak (2), sve produkcije su oblika $A \to a$ ili $A \to B_1 B_2 \cdots B_m$, gdje je $m \geq 2$. Znakovi $A$ i $B_i$ su nezavršni znakovi skupa $V'$, a znak $a$ je završni znak. Produkcije oblika $A \to a$ i $A \to B_1 B_2$ su u Chomskyjevom normalnom obliku. Produkcije koje imaju tri i više znakova na desnoj strani:

$$A \to B_1 B_2 \cdots B_m, \text{ gdje je } m \geq 3,$$

zamijene se novim produkcijama. Definiraju se novi nazavršni znakovi $D_1$, $D_2$, ..., $D_{m-2}$, a zatim se produkcija $A \to B_1 B_2 \cdots B_m$ zamijeni skupom produkcija:

$$\{ A \to B_1 D_1, D_1 \to B_2 D_2, D_2 \to B_3 D_3, \cdots, D_{m-3} \to B_{m-2} D_{m-2}, D_{m-2} \to B_{m-1} B_m \}$$

6. Navesti definiciju determinističkog potisnog automata, i obrazložiti pojedine uvjete.
*106.str, 110. str*

Potisni automat (PA) formalno se zadaje uređenom sedmorkom:

$$pa = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

gdje je:

| | |
|---|---|
| $Q$ | - konačan skup stanja; |
| $\Sigma$ | - konačan skup ulaznih znakova (abeceda ulaznih znakova); |
| $\Gamma$ | - konačan skup znakova stoga (abeceda znakova stoga); |
| $\delta$ | - funkcija prijelaza $\delta$ pridružuje trojki $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma$ konačan podskup skupa svih mogućih parova $Q \times \Gamma^*$; |
| $q_0 \in Q$ | - početno stanje; |
| $Z_0 \in \Gamma$ | - početni znak stoga; |
| $F \subseteq Q$ | - skup prihvatljivih stanja. |

PA $M=(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ jest deterministički PA (DPA) ako i samo ako su ispunjena oba sljedeća uvjeta:

*i)* Ako je $\delta(q, \varepsilon, Z)$ neprazni skup, onda je $\delta(q, a, Z)$ prazni skup za bilo koji ulazni znak $a$ iz $\Sigma$.

*ii)* U skup $\delta(q, a, Z)$ je najviše jedan element, i to za bilo koje stanje $q$ iz $Q$, za bilo koji znak stoga $Z$ iz $\Gamma$ i za bilo koji ulazni znak $a$ iz $\Sigma \cup \{\varepsilon\}$.

Uvjet (*i*) ne dozvoljava mogućnost izbora između $\varepsilon$-prijelaza i prijelaza koji koristi ulazni zna primjeru 3.17 prijelazi (1) i (9) imaju za posljedicu opisani nedeterminizam. Iz konfiguracije $(q_1, 00110$ prelazi se u konfiguraciju $(q_2, 001100, \varepsilon)$ primjenom $\varepsilon$-prijelaza $\delta(q_1, \varepsilon, K)=\{(q_2, \varepsilon)\}$, ili u konfiguraciju $01100, NK)$ primjenom prijelaza $\delta(q_1, 0, K)=\{(q_1, NK)\}$. Sličan nedeterminizam vrijedi za prijelaze (2) i (

Uvjet (*ii*) ne dopušta da je u skupu $\delta(q, a, Z)$ više od jednog elementa. Prijelazi (3) i (6) su pri nedeterminističkih prijelaza. U skupovima $\delta(q_1, 0, N)$ i $\delta(q_1, 1, J)$ su po dva elementa. Na slici 3.20 prika je više nedeterminističkih prijelaza. Na primjer, iz konfiguracije $(q_1, 01100, NK)$ prelazi se u konfigur $(q_1, 1100, NNK)$ ili u konfiguraciju $(q_2, 1100, K)$.

Za razliku od NKA koji prihvaća istu klasu jezika kao i DKA, nedeterministički PA prihvaća klasu jezika od determinističkog PA. U primjeru 3.17 pokazano je da postoji nedeterministički PA $M_2$ prihvaća jezik $N(M_2)=\{ww^R\}$. Međutim, za jezik $N(M_2)$ nije moguće izgraditi deterministički PA. U dalj tekstu pod pojmom PA podrazumijeva se nedeterministički PA. Govori li se o determinističkom PA, posebno naglasi i označi kao DPA.

7. Navesti postupak konstrukcije potisnog automata koji prihvaća praznim stogom iz potisnog automata koji prihvaća prihvatljivim stanjem.

*110.str*

Konstruira se PA $M_1$:

$$M_1=(Q\cup\{q_0', q_e\}, \Sigma, \Gamma\cup\{X_0\}, \delta', q_0', X_0, \varnothing),$$

Funkcija prijelaza PA $M_1$ gradi se na sljedeći način:

1) $\delta'(q_0', \varepsilon, X_0) = \{(q_0, Z_0 X_0)\}$.

Na početku rada PA $M_1$ prelazi iz svoje početne konfiguracije u početnu konfiguraciju PA $M_2$. PA $M_1$ ostavlja znak $X_0$ na dnu stoga.

2) U skup $\delta'(q, a, Z)$ stave se svi elementi skupa $\delta(q, a, Z)$.

Skup $\delta'(q, a, Z)$ računa se za sva stanja $q$ iz $Q$, za sve znakove $a$ iz $\Sigma\cup\{\varepsilon\}$ i za sve znakove stoga $Z$ iz $\Gamma$. Ovi prijelazi omogućuju simuliranje rada PA $M_2$.

3) U skup $\delta'(q, \varepsilon, Z)$ dodaje se $\varepsilon$-prijelaz $(q_e, \varepsilon)$, $q\in F$.

$\varepsilon$-prijelazi dodaju se za sva stanja $q$ iz skupa prihvatljivih stanja $F$ i za sve znakove stoga $Z$ iz $\Gamma\cup\{X_0\}$. Uđe li PA $M_2$ u jedno od prihvatljivih stanja, skup prijelaza proširuje se $\varepsilon$-prijelazom u stanje $q_e$. Istodobno se s vrha stoga uzme jedan znak.

4) U skup $\delta'(q_e, \varepsilon, Z)$ dodaje se $\varepsilon$-prijelaz $(q_e, \varepsilon)$.

$\varepsilon$-prijelazi dodaju se za sve znakove stoga $Z$ iz $\Gamma\cup\{X_0\}$. $\varepsilon$-prijelazi u stanju $q_e$ prazne čitav stog, uključujući i znak $X_0$.

| | |
|---|---|
| 8. | Opisati postupak konstrukcije potisnog automata $M_2$ koji prihvaća prihvatljivim stanjem iz potisnog automata $M_1$ koji prihvaća praznim stogom. |

*112. str*

Konstruira se PA $M_2$ :

$$M_2=(Q\cup\{q_0', q_f\}, \Sigma, \Gamma\cup\{X_0\}, \delta', q_0', X_0, \{q_f\})$$

Funkcija prijelaza PA $M_2$ gradi se na sljedeći način:

1) $\delta'(q_0', \varepsilon, X_0) = \{(q_0, Z_0 X_0)\}$.

Na početku rada PA $M_2$ prelazi iz svoje početne konfiguracije u početnu konfiguraciju PA $M_1$. Na dno stoga stavlja se znak $X_0$. Pročita li se tijekom simulacije na vrhu stoga znak $X_0$, to je znak da je PA $M_1$ ispraznio svoj stog.

2) U skup $\delta'(q, a, Z)$ stave se svi elementi skupa $\delta(q, a, Z)$.

Skup $\delta'(q, a, Z)$ računa se za sva stanja $q$ iz $Q$, za sve znakove $a$ iz $\Sigma\cup\{\varepsilon\}$ i za sve znakove stoga $Z$ iz $\Gamma$. Ovi prijelazi omogućuju simuliranje rada PA $M_1$.

3) U skup $\delta'(q, \varepsilon, X_0)$ dodaje se $\varepsilon$-prijelaz $(q_f, \varepsilon)$.

$\varepsilon$-prijelazi dodaju se za sva stanja $q$ iz $Q$. Pročita li se na vrhu stoga znak $X_0$, to je znak da je PA $M_1$ ispraznio svoj stog. PA $M_2$ prelazi u prihvatljivo stanje $q_f$. Prihvaća li PA $M_1$ niz praznim stogom, prijelaz u stanje $q_f$ omogućuje prihvaćanje niza prihvatljivim stanjem.

Na temelju koraka (1) do (3) konstrukcije PA $M_2$ dobije se slijed prijelaza:

$$(q_0', x, X_0) \underset{M_2}{\succ} (q_0, x, Z_0X_0) \overset{*}{\underset{M_2}{\succ}} (q, \varepsilon, X_0) \underset{M_2}{\succ} (q_f, \varepsilon, \varepsilon),$$

gdje je prvi prijelaz definiran korakom (1) konstrukcije PA $M_2$, slijed prijelaza $(q_0, x, Z_0X_0) \overset{*}{\underset{M_2}{\succ}} (q, \varepsilon, X_0)$ simulira prijelaze $(q_0, x, Z_0) \overset{*}{\underset{M_1}{\succ}} (q, \varepsilon, \varepsilon)$ PA $M_1$, i posljednji prijelaz definiran je korakom (3). Prikazani slijed prijelaza pokazuje da PA $M_2$ prihvaća niz $x$ prihvatljivim stanjem ako i samo ako PA $M_1$ prihvaća niz $x$ praznim stogom.

| | |
|---|---|
| 9. | Opisati postupak konstrukcije konteksno neovisne gramatike koja generira jezik koji se prihvaća praznim stogom potisnog automata. |

*115.str*

Za zadani PA:

$$M=(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

konstruira se kontekstno neovisna gramatika:

$$G=(V, T, P, S).$$

Nezavršni znakovi gramatike označeni su zagradama $[q, A, p] \in V$. U oznaci nezavršnog znaka, $q$ i $p$ su stanja skupa $Q$, a $A$ je znak stoga iz skupa $\Gamma$. U skup nezavršnih znakova dodaje se početni nezavršni znak $S$. Skup produkcija gradi se na sljedeći način:

1) Za početno stanje $q_0$, početni znak stoga $Z_0$ i sva stanja $q$ iz $Q$ grade se produkcije:

$$S \to [q_0, Z_0, q].$$

Ako je kardinalni broj skupa stanja $Q$ jednak $n$, onda je broj produkcija $S \to [q_0, Z_0, q]$ jednak također $n$.

2) Ako skup $\delta(q, a, A)$ sadrži:

$$(q_1, B_1B_2 \cdots B_m),$$

onda se grade produkcije:

$$[q, A, q_{m+1}] \to a[q_1, B_1, q_2][q_2, B_2, q_3]\,][q_3, B_3, q_4] \cdots [q_{m-1}, B_m, q_m]\,[q_m, B_m, q_{m+1}]$$

gdje su znakovi $q$, $A$, $a$, $q_1$, $B_1$, $B_2$, ... i $B_m$ određeni datom funkcijom prijelaza PA, a za znakove $q_2$, $q_3$, ... i $q_{m+1}$ uzimaju se sve moguće kombinacije svih stanja iz skupa $Q$. Budući da je kardinalni broj skupa stanja $Q$ jednak $n$ i budući da se uzimaju sve moguće kombinacije za niz $q_2$, $q_3$, ... i $q_{m+1}$, za svaku funkciju prijelaza gradi se $n^{m-1}$ produkcija.

Ako $\delta(q, a, A)$ sadrži $(q_1, \varepsilon)$, onda se gradi produkcija $[q, A, q_1] \to a$.

10. Dokazati da su kontekstno neovisni jezici zatvoreni s obzirom na Kleenov operator.

*Str 120.*

Kontekstno neovisni jezici zatvoreni su s obzirom na Kleeneov operator.

Neka gramatika $G_1=(V_1, T_1, P_1, S_1)$ generira jezik $L(G_1)$. Gramatika $G_5=(V_5, T_5, P_5, S_5)$ koja generira jezik $L(G_5)=L(G_1)^*$ konstruira se na sljedeći način:

1) $V_5 = V_1 \cup \{S_5\}$, gdje je $S_5 \notin V_1$.

2) $T_5 = T_1$.

3) U skup produkcija $P_5=P_1$ dodaju se produkcije:

$$S_5 \to S_1\,S_5 \mid \varepsilon.$$

Dokaz se zasniva na sljedeća dva postupka generiranja niza:

1) $S_5 \underset{G_5}{\Rightarrow} S_1S_5 \overset{*}{\underset{G_1}{\Rightarrow}} w_1S_5 \underset{G_5}{\Rightarrow} w_1S_1S_5 \overset{*}{\underset{G_1}{\Rightarrow}} w_1w_1S_5 \underset{G_5}{\Rightarrow} \cdots \overset{*}{\underset{G_1}{\Rightarrow}} w_1^+S_5 \underset{G_5}{\Rightarrow} w_1^+$, gdje su nizovi završnih znakova $w_1^+ \in L(G_5)$ i $w_1 \in L(G_1)$;

2) $S_5 \Rightarrow \varepsilon$.

11. Dokazati zatvorenost konteksno neovisnih jezika s obzirom na supstituciju.

*Str 121.*

Kontekstno neovisni jezici zatvoreni su s obzirom na supstituciju.

Pretpostavimo da gramatika $G=(V, T, P, S)$ generira kontekstno neovisni jezik $L(G)$. Neka s završni znakovi $a_i$ jezika $L(G)$ zamijene nizovima kontekstno neovisnog jezika $L(G_i)$, gdje $1\leq i\leq k$. Broj kardinalni broj skupa završnih znakova $T$. Neka gramatika $G_i=(V_i, T_i, P_i, S_i)$ generira jezik $L(G_i)$. Nastali $L'$ jest kontekstno neovisni jezik za koji je moguće konstruirati gramatiku $G'=(V', T', P', S')$ na sljedeći na

1) U skup nezavršnih znakova $V'$ stave se svi nezavršni znakovi gramatike $G$ i nezavršni zn svih gramatika $G_i$:

$$V' = V\cup V_1\cup V_2 \text{ - - - } V_k,\ V\cap V_i=\emptyset,\ V_i\cap V_j=\emptyset, \text{ za } 1\leq i\leq k,\ 1\leq j\leq k,\ i\neq j$$

2) U skupu završnih znakova $T'$ su isključivo završni znakovi gramatika $G_i$:

$$T' = T_1\cup T_2 \text{ - - - } T_k.$$

3) Početni nezavršni znak $S'$ jest početni nezavršni znak $S$ gramatike $G$:

$$S' = S.$$

4) U skup produkcija $P'=P_1\cup P_2$ - - - $P_k$ dodaju se produkcije gramatike $G$ koje se prethodno preurede na sljedeći način:

U produkcijama gramatike $G$ bilo koji završni znak $a_i$ zamijeni se početnim nezavršnim znakom $S_i$ gramatike $G_i$.

12. Pokazati da konteksno neovisni jezici nisu zatvoreni s obzirom na presjek.

*Str 122.*

Prethodno je pokazano, a u primjeru 3.24 je dodatno objašnjeno, da jezik $L=\{a^ib^ic^i \mid i\geq1\}$ nije kontekstno neovisan. Jezik $L$ je presjek jezika $L_1=\{a^ib^ic^j \mid i\geq1 \text{ i } j\geq1\}$ i jezika $L_2=\{a^jb^ic^i \mid i\geq1 \text{ i } j\geq1\}$. U jeziku $L_1$ su nizovi koji imaju jednak broj znakova $a$ i $b$, te proizvoljan broj znakova $c$. U jeziku $L_2$ su nizovi koji imaju jednak broj znakova $b$ i $c$, te proizvoljan broj znakova $a$. Presjek jezika $L_1$ i $L_2$ jest jezik $L$ koji ima jednak broj znakova $a$, $b$ i $c$.

Jezik $L_1=\{a^ib^ic^j \mid i\geq1 \text{ i } j\geq1\}$ prihvaća PA koji za svaki pročitani ulazni znak $a$ sprema na stog znak stoga $A$, a zatim uspoređuje broj znakova $A$ spremljenih na stog sa brojem pročitanih znakova $b$. Na kraju PA provjera da li je u nizu barem jedan znak $c$. Jezik $L_1$ generira kontekstno neovisna gramatika $G_1=(\{S, A, B\}, \{a, b, c\}, P, S)$ koja ima produkcije:

$$S \to A\ C$$
$$A\to a\ A\ b \mid a\ b$$
$$C\to c\ C \mid c.$$

Konstrukcijom PA i kontekstno neovisne gramatike pokazano je da je jezik $L_1$ kontekstno neovisni jezik.

Na sličan način moguće je izgraditi PA za jezik $L_2=\{a^jb^ic^i \mid i\geq1 \text{ i } j\geq1\}$. PA provjerava ima li jednak broj znakova $b$ i $c$. Gramatika $G_2=(\{S, C, D\}, \{a, b, c\}, P, S)$ koja generira jezik $L_2$ ima produkcije:

$$S \to A\ B$$
$$A\to a\ A \mid a$$
$$B\to b\ B\ c \mid b\ c.$$

Budući da su oba jezika $L_1$ i $L_2$ kontekstno neovisni jezici, a jezik $L$ koji je njihov presjek nije kontekstno neovisni jezik, pokazano je da kontekstno neovisni jezici nisu zatvoreni s obzirom na operaciju *presjeka*.

13. Dokazati da je presjek konteksno neovisnog jezika i regularnog jezika konteksno neovisni jezik.

*Str 123.*

Presjek kontekstno neovisnog jezika i regularnog jezika jest kontekstno neovisni jezik.

Pretpostavimo da kontekstno neovisni jezik $L_1$ prihvaća PA $M_1=(Q_1, \Sigma, \Gamma, \delta_1, q_0, Z_1, F_1)$, a da regularni jezik $L_2$ prihvaća DKA $M_2=(Q_2, \Sigma, \delta_2, p_0, F_2)$. Moguće je izgraditi PA $M'=(Q', \Sigma, \Gamma, \delta', q'_0, Z_0, F')$ koji prihvaća jezik prihvatljivm stanjem $L=L_1 \cap L_2$:

1) $Q' = Q_2 \times Q_1$.

2) $q'_0=[p_0, q_0]$.

3) $F' = F_2 \times F_1$.

4) Skup:

$\delta'([p, q], a, X)$ sadrži $([p', q'], \gamma)$

ako i samo ako za funkciju prijelaza DKA $M_2$ vrijedi:

$\delta_2(p, a)=p'$,

a za funkciju prijelaza PA $M_1$ vrijedi:

$(q', \gamma) \in \delta_1(q, a, X)$.

Ako $a$ jest $\varepsilon$, onda je $p' = p$.

PA $M'$ simulira rad automata $M_1$ i $M_2$. Dokaz da $M'$ prihvaća jezik $L=L_1 \cap L_2$ zasniva se na indukciji s obzirom na broj koraka $i$. Dokazuje se da za PA $M'$ vrijedi:

$$([p_0, q_0], w, Z_0) \overset{i}{\underset{M'}{\succ}} ([p, q], \varepsilon, \gamma)$$

ako i samo ako je:

$$(q_0, w, Z_0) \overset{i}{\underset{M_1}{\succ}} (q, \varepsilon, \gamma) \quad \text{i} \quad \delta_2(p_0, w)=p,$$

gdje je $[p, q] \in F'$ ako i samo ako je $p \in F_2$ i $q \in F_1$.

14. Opisati algoritam kojim se provjerava da li je regularni jezik prazan.

15. Opisati pojam nejednoznačnosti kontekstno-neovisne gramatike i pokazati na primjeru.

Nejednoznačnost kontekstno neovisne gramatike $G$ definira se na sljedeći način:

Ako je moguće za neki niz $w \in L(G)$ izgraditi više različitih generativnih stabala, onda je kontekstno neovisna gramatika $G$ nejednoznačna.

| |
|---|
| 16. Utvrdite da li nadovezivanjem kontekstno-ovisnih jezika nastaje kontekstno-ovisan jezik i objasnite tvrdnju. (2 boda).<br><br>Nadovezivanje kontekstno neovisnih jezika jest kontekstno neovisni jezik.<br><br>Neka gramatika $G_1=(V_1, T_1, P_1, S_1)$ generira jezik $L(G_1)$, a gramatika $G_2=(V_2, T_2, P_2, S_2)$ neka generira jezik $L(G_2)$. Gramatika $G_4=(V_4, T_4, P_4, S_4)$ koja generira jezik $L(G_4)=L(G_1)L(G_2)$ konstruira se na sljedeći način:<br><br>1) $V_4 = V_1 \cup V_2 \cup \{S_4\}$, gdje je $S_4 \notin V_1$ i $S_4 \notin V_2$, $V_1 \cap V_2 = \emptyset$.<br><br>2) $T_4 = T_1 \cup T_2$.<br><br>3) U skup produkcija $P_4=P_1 \cup P_2$ doda se produkcija:<br><br>$S_4 \rightarrow S_1\, S_2$.<br><br>Dokazuje se na sličan način kao i za operaciju unije. Dokaz se zasniva na sljedećem postupku generiranja niza:<br><br>$S_4 \underset{G_4}{\Rightarrow} S_1\, S_2 \overset{*}{\underset{G_1}{\Rightarrow}} w_1\, S_2 \overset{*}{\underset{G_2}{\Rightarrow}} w_1\, w_2$, gdje su nizovi završnih znakova $w_1 w_2 \in L(G_4)$, $w_1 \in L(G_1)$ i $w_2 \in L(G_2)$. |
| 17. Ukratko opisati osnove značajke parsiranja niza tehnikom rekurzivnog spusta (svojstva programskoj jezika, struktura programa, veza programa i gramatike)<br><br>Parsiranje od vrha prema dnu moguće je jednostavno programski ostvariti tehnikom rekurzivnog spusta. Tehnika rekurzivnog spusta ostvaruje se programskim jezikom koji ima svojstvo rekurzivnog poziva potprograma. Nezavršnim znakovima gramatike pridružuju se potprogrami. Potprogram ispituje da li podniz zadanog niza zadovoljava strukturu zadanu desnom stranom one produkcije u kojoj je nezavršni znak pridružen tom potprogramu na lijevoj strani. Završni znakovi na desnoj strani produkcije uspoređuju se sa znakovima u nizu. Za nezavršne znakove na desnoj strani produkcije pozivaju se potprogrami koji provjeravaju strukturu za te nezavršne znakove. Nađe li se jedan te isti nezavršni znak na lijevoj i desnoj strani produkcije, potprogram pridružen tom nezavršnom znaku poziva se rekurzivno. Potprogrami je moguće rekurzivno pozivati i pomoću drugih potprograma. |
| 18. 3. Opisati postupa konstrukcije gramatike kojom se dokazuje da su konteksno-neovisni jezicizatvoreni s obzirom na proporciju unije<br><br>Gramatika $G_3=(V_3, T_3, P_3, S_3)$ koja generira jezik $L(G_1) \cup L(G_2)$ konstruira se na sljedeći način:<br><br>1) $V_3 = V_1 \cup V_2 \cup \{S_3\}$, gdje je $S_3 \notin V_1$ i $S_3 \notin V_2$, $V_1 \cap V_2 = \emptyset$.<br><br>2) $T_3 = T_1 \cup T_2$.<br><br>3) U skup produkcija $P_3=P_1 \cup P_2$ dodaju se produkcije:<br><br>$S_3 \rightarrow S_1 \mid S_2$. |

# Rekurzivno prebrojivi jezici

1. (3 boda) Formalno definirati osnovni model Turingovog stroja. Navesti koje uvjete u odnosu na nedeterministički Turingov stroj zadovoljava linearno ograničeni automat.

Str 127.

Turingov stroj (TS) formalno se zadaje uređenom sedmorkom:

$$ts = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

$Q$ - konačan skup stanja;
$\Gamma$ - konačan skup znakova trake;
$B \in \Gamma$ - znak kojim se označava prazna ćelija;
$\Sigma \subseteq (\Gamma - \{B\})$ - konačan skup ulaznih znakova;
$\delta$ - funkcija prijelaza $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$, gdje $L$ i $R$ označavaju pomak glave u lijevo i desno;
$q_0 \in Q$ - početno stanje;
$F \subseteq Q$ - skup prihvatljivih stanja.

2. **Dokazati zatvorenost regularnih jezika s obzirom na operaciju komplementa.**

Neka DKA $M=(Q, \Sigma, \delta, q_0, F)$ prihvaća regularni jezik $L(M)$. Za komplement jezika $L(M)^c$ moguće je izgraditi DKA $M'=(Q, \Sigma, \delta, q_0, Q \setminus F)$ koji prihvaća jezik $L(M')=\{w \mid \delta(q_0,w) \notin F\}= \Sigma^* \{w \mid \delta(q_0,w) \in F\}=L(M)^c$. (str. 52)

3. Opisati postupak simulacije Turingovog stroja s jednom trakom pomoću stogovnog stroja s dva stog

Jedan stog koristi se za spremanje znakova koji su lijevo od glave za čitanje i pisanje, a drugi stog koristi se za spremanje znakova koji su desno od glave za čitanje i pisanje. Na vrhu stoga su znakovi trake koji su bliže glavi za čitanje i pisanje. Nisu li stogovi prazni, na dnu jednog stoga je sadržaj krajnje lijeve ćelije, a na dnu drugog stoga je sadržaj krajnje desne ćelije do koje se pomaknula glava TS.

4. Navesti rekurzivna pravila za regularne izraze te pravila asocijativnosti i prednosti za osnovne operatore koji se koriste u regularnim izrazima.

1) $\varnothing$ jest regularni izraz i označava jezik $L(\varnothing)=\{\}$.

2) $\varepsilon$ jest regularni izraz i označava jezik $L(\varepsilon)=\{\varepsilon\}$.

3) $\forall a \in \Sigma$, $a$ jest regularni izraz i označava jezik $L(a)=\{a\}$. Treba biti pažljiv, jer je istom oznakom $a$ označen znak abecede $\Sigma$, regularni izraz i niz jedinične duljine koji je element jezika $L(a)$.

4) Ako su $r$ i $s$ regularni izrazi koji označavaju jezike $L(r)$ i $L(s)$, onda:

*a)* $(r)+(s)$ jest regularni izraz koji označava jezik $L((r)+(s))=L(r)\cup L(s)$. Jezik $L((r)+(s))$ nastaje unijom jezika $L(r)$ i $L(s)$. Često se koristi i oznaka $(r)\,|\,(s)$. Oznaka $(r)\,|\,(s)$ koristi se za definiranje aritmetičkih izraza kako bi se ona razlikovala od oznake aritmetičkog operatora zbrajanja +.

*b)* $(r)(s)$ jest regularni izraz koji označava jezik $L((r)(s))=L(r)L(s)$. Jezik $L((r)(s))$ nastaje nadovezivanjem jezika $L(r)$ i $L(s)$.

*c)* $(r)^*$ jest regularni izraz koji označava jezik $L((r)^*)=L(r)^*$. Jezik $L((r)^*)$ nastaje primjenom Kleeneovog operatora nad jezikom $L(r)$.

Pravila asocijativnosti i prednosti operatora definiraju se na sljedeći način:

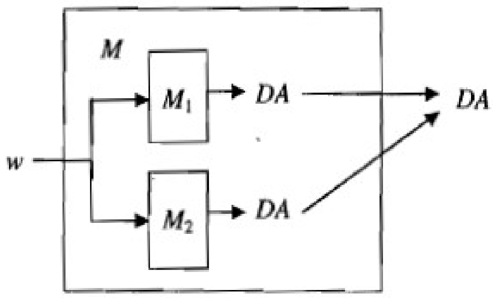
1) Unarni operator * jest lijevo asocijativan i najveće je prednosti.

2) Operator nadovezivanja jest lijevo asocijativan i veće je prednosti od operatora +.

3) Operator + jest lijevo asocijativan i najmanje je prednosti.

5. Dokazati da su rekurzivno prebrojivi jezici zatvoreni s obzirom na uniju.

Unija rekurzivno prebrojivih jezika jest rekurzivno prebrojiv jezik.




**Slika 4.10:** Konstrukcija TS $M$ koji prihvaća uniju dva rekurzivno prebrojiva jezika $L(M_1)$ i $L(M_2)$

Prethodno opisana slijedna simulacija nije moguće primijeniti na rekurzivno prebrojive jezike. Budući da je $L_1$ rekurzivno prebrojiv jezik, moguće je da TS $M_1$ nikad ne stane za neki niz $w$ koji nije u jeziku $L_1$. Ako je niz $w$ u jeziku $L_2$, onda je niz $w$ u uniji jezika $L_1$ i $L_2$. TS $M$ mora stati i prihvatiti niz $w$. Budući da TS $M_1$ nikad ne stane, nije moguće pokrenuti izvođenje TS $M_2$. Ne pokrene li se izvođenje TS $M_2$, TS $M$ nikad ne stane i ne prihvatiti niz $w$ bez obzira što je niz u uniji jezika.

6. Opisati postupak simulacije Turingovog stroja s višestrukim trakama pomoću Turingovog stroja s višestrukim tragovima.

Model TS s višestrukim dvostranim beskonačnim trakama prikazan je na slici 4.5. TS ima $k$ glava za čitanje i pisanje i $k$ dvostrano beskonačnih traka. Upravljačka jedinka TS donosi odluku na temelju dviju grupa parametara:

*a)* stanje upravljačke jedinke;

*b)* $k$ pročitanih znakova sa $k$ traka.

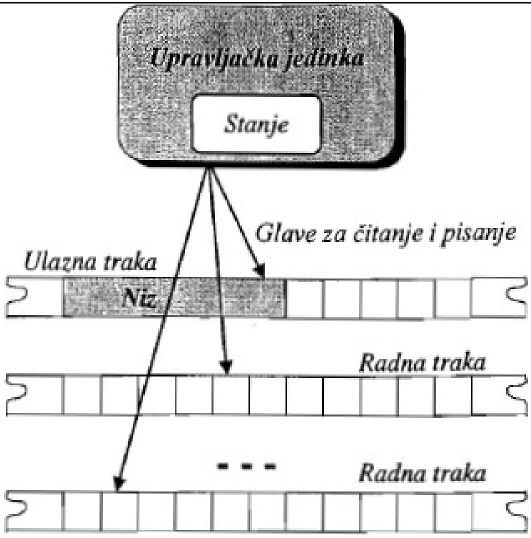
Jednim prijelazom TS:

1) promijeni stanje;

2) zapiše $k$ znakova na $k$ traka;

3) pomakne bilo koju od $k$ glava nezavisno u desno ili lijevo.

Na jednu traku, koja se naziva ulazna traka, zapiše se niz koji se ispituje. Sve ostale trake nazivaju se radnim trakama.




**Slika 4.5:** Model TS s višestrukim trakama

7. Dokazati da su kontekstno-ovisni jezici pravi podskup rekurzivnih jezika.

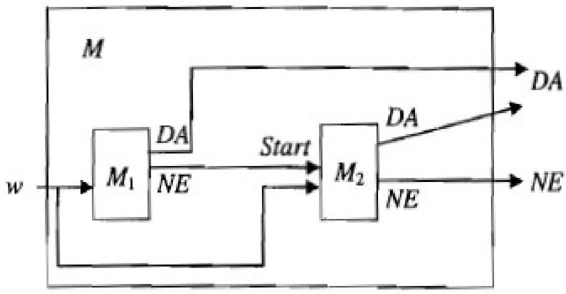
Dokaz da su kontekstno ovisni jezici pravi podskup rekurzivnih jezika zasniva se na sljedećem svojstvu:

*a)* Zadan je skup jezika $K=\{L_{x1}, L_{x2}, L_{x3}, \ldots\}$. Neka za bilo koji jezik $L_{xi}$ u zadanom skupu $K$ postoji barem jedan TS $M_{xi}$ koji prihvaća jezik $L_{xi}$ i koji uvijek stane za bilo koji ulazni niz. Svi TS $M_{xi}$ kodiraju se i poredaju određenim redoslijedom u niz $M_1, M_2, M_3, \ldots$ . Postoji li TS $M_K$ koji redom ispisuje na izlaznu traku kôdove svih TS iz skupa $K$, skup jezika $K$ je pravi podskup skupa rekurzivnih jezika RJ, odnosno $K \subset$ RJ.

8. Dokazati da je unija dva rekurzivna jezika također rekurzivni jezik.

Unija rekurzivnih jezika jest rekurzivni jezik.

Neka TS $M_1$ prihvaća rekurzivni jezik $L_1$, a TS $M_2$ neka prihvaća rekurzivni jezik $L_2$. Budući da su jezici $L_1$ i $L_2$ rekurzivni i budući da TS $M_1$ i TS $M_2$ uvijek stanu za bilo koji ulazni niz, rad TS $M_1$ i rad TS $M_2$ moguće je slijedno simulirati. Slijedna simulacija prikazana je shematski na slici 4.9. Ako TS $M_1$ stane i prihvati niz, onda stane TS $M$ i prihvati niz. Stane li TS $M_1$ i ne prihvati niz, TS $M$ pokrene simulaciju rada TS $M_2$. Ako TS $M_2$ stane i prihvati niz, onda stane TS $M$ i prihvati niz. Stane li TS $M_2$ i ne prihvati li niz, a budući da prethodno ni TS $M_1$ nije prihvatio niz, TS $M$ stane i ne prihvati niz.



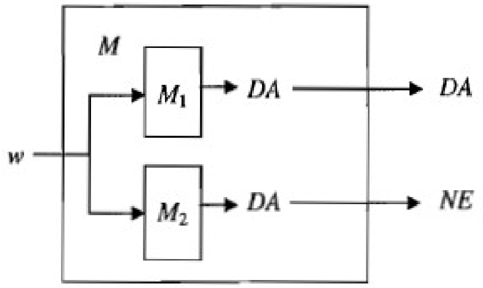

**Slika 4.9:** Konstrukcija TS $M$ koji prihvaća uniju rekurzivnih jezika $L(M_1)$ i $L(M_2)$

9. Jesu li rekurzivni i rekurzivno prebrojivi jezici zatvoreni s obzirom na operaciju unije? Dokažite t

JESU...dokazi 5. i 8. pitanje

10. Dokazati da je jezik L rekurzivan ako su jezici L i njegov komplement $L^C$ oba rekurzivno prebrojivi.

Neka TS $M_1$ prihvaća rekurzivno prebrojiv jezik $L_1$, a TS $M_2$ neka prihvaća rekurzivno prebrojiv jezik $L_2$ koji je komplement rekurzivno prebrojivog jezika $L_1$, odnosno $L_2 = L_1^c$. Gradi se TS $M$ koji prihvaća niz $w$ ako i samo ako TS $M_1$ prihvaća niz $w$, odnosno TS $M$ ne prihvaća niz $w$ ako i samo ako TS $M_2$ prihvaća niz $w$. Shema TS $M$ prikazana je na slici 4.12. Budući da je niz $w$ u jeziku $L_1$ ili u jeziku $L_2 = L_1^c$, jedan i samo jedan TS $M_1$ ili TS $M_2$ prihvaća niz. TS $M$ uvijek stane i jednoznačno odluči da li se niz prihvaća. Ako su jezik $L_1$ i njegov komplement rekurzivno prebrojivi, onda je za jezik $L_1$ moguće izgraditi TS $M$ koji uvijek stane za bilo koji ulazni niz, odnosno jezik $L_1$ jest rekurzivan.




**Slika 4.12:** Shema dokaza da je jezik $L(M)$ rekurzivni jezik ako su $L=L(M_1)$ i njegov komplement $L^c = L(M_2)$ rekurzivno prebrojivi jezici

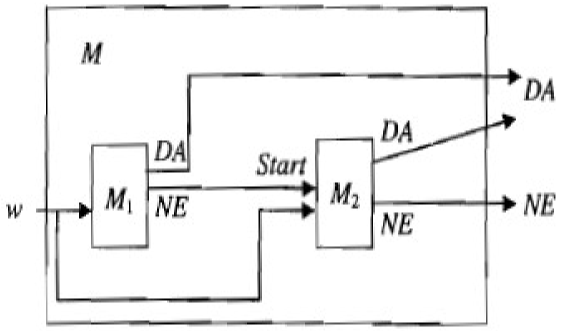
11. Dokazati da konkatenacija dva kontekstno-neovisna jezika daje kontekstno-neovisan jezik.

Konkatenacija je valjda vezivanje pa se misli na nadovezivanje... O.o
Odgovor u 17. u konetekstnoneovisnim jezicima....dokaz za nadovezivanje

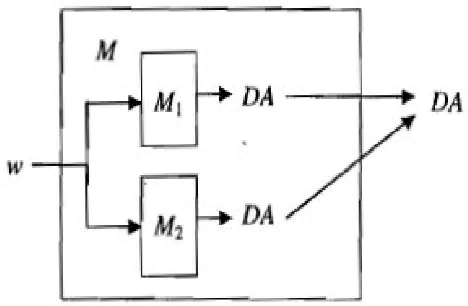
12. Objasniti razliku između rekurzivnih i rekurzivno prebrojivih jezika. (3 boda)

Definicija rekurzivno prebrojivih jezika zasniva se na Turingovom stroju: jezik jest rekurzivno prebrojiv ako i samo ako postoji Turingov stroj koji ga prihvaća. Time je definirana istovjetnost Turingovog stroja i rekurzivno prebrojivih jezika: za bilo koji rekurzivno prebrojiv jezik moguće je izgraditi Turingov stroj koji ga prihvaća, i obrnuto, bilo koji Turingov stroj prihvaća jedan od rekurzivno prebrojivih jezika.

13. Opisati postupak i nacrtati shematski prikaz konstrukcije Turingovog stroja koji prihvaća uniju
a) tri (3) rekurzivna jezika (1,5 bodova)
b) tri (3) rekurzivno prebrojiva jezika (1,5 bodova)




**Slika 4.9:** Konstrukcija TS $M$ koji prihvaća uniju rekurzivnih jezika $L(M_1)$ i $L(M_2)$




**Slika 4.10:** Konstrukcija TS $M$ koji prihvaća uniju dva rekurzivno prebrojiva jezika $L(M_1)$ i $L(M_2)$

# Konteksno ovisni jezici

1. Opisati postupak konstrukcije linearno ograničenog automata za jezik zadan konteksno-ovisnom gramatikom.

*168. str*

Postupak gradnje LOA za jezik zadan kontekstno ovisnom gramatikom sličan je postupku gradnje TS za jezik zadan gramatikom neograničenih produkcija. Postupak je opisan u odjeljku 4.2.1. Umjesto dvije trake, LOA $M$ koristi dva traga ulazne trake. U gornji trag LOA $M$ zapiše niz $¢w\$$, a na početak donjeg traga LOA $M$ zapiše početni nezavršni znak $S$ gramatike $G$.

Za prazni niz $\varepsilon$ LOA odmah stane i ne prihvati niz. Ako niz $w$ nije prazan, onda LOA $M$ primijeni postupak simulacije gramatike $G$ opisan u odjeljku 4.2.1. Budući da se tijekom simulacije na donji trag ulazne trake generiraju nizovi primjenom produkcija gramatike $G$ i budući da nedeterministički izbori generiraju sve nizove jezika $L(G)$, LOA $M$ prihvaća niz završnih znakova $w$ ako i samo ako gramatika $G$ generira niz $w$.

Tijekom rada LOA $M$ ne koristi veći broj ćelija od duljine niza $w$. Budući da su desne strane svih produkcija kontekstno ovisne gramatike jednako dugačke ili dulje od lijevih strana, u postupku generiranja niza:

$$S \overset{*}{\Rightarrow} \alpha \overset{*}{\Rightarrow} w$$

niti jedan međuniz $\alpha$ na donjem tragu ulazne trake nije dulji od niza $w$ koji je zapisan na gornjem tragu. Ako je generirani međuniz $\alpha$ dulji od niza $w$, onda se daljnja simulacija zaustavlja bez ulaska LOA $M$ u prihvatljivo stanje. Budući da nije moguće generirati niz $w$ iz međuniza $\alpha$ u slučaju da je $|w|<|\alpha|$, nije potrebno da LOA $M$ nastavi simulaciju generiranja međunizova niti za jedan niz $\alpha$ koji je dulji od niza $w$.

2. Opisati konstrukciju konteksno ovisne gramatike koja generira jezik koji prihvaća linearno ograničeni automat.

Početna konfiguracija LOA jest $q_0\, ¢\, a_1\, a_2 \cdots a_n\, \$$. Sljedeće produkcije gramatike generiraju međuniz koji predstavlja početnu konfiguraciju:

1) $A_1 \to [a,\ q_0 ¢ a]\, A_2$,

2) $A_1 \to [a,\ q_0 ¢ a \$]$,

3) $A_2 \to [a, a]\, A_2$, za sve $a \in \Sigma$,

4) $A_2 \to [a, a\, \$]$, za sve $a \in (\Sigma \backslash \{¢, \$\})$.

Produkcije koje simuliraju prijelaze LOA slične su produkcijama (6) i (7) u odjeljku 4.2.2. Oznake stanja grupiraju se zajedno sa znakovima trake:

5) $[b, q\, X]\,[a, Z] \to [b, Y]\,[a, p\, Z]$,
$[b, q\, ¢\, X] \to [b, ¢\, p\, X]$,
$[b, ¢\, q\, X]\,[a, Z] \to [b, ¢\, Y]\,[a, p\, Z]$,
$[b, q\, X]\,[a, Z\, \$] \to [b, Y]\,[a, p\, Z\, \$]$,
$[b, q\, X\, \$] \to [b, Y\, p\, \$]$,

6) $[b, Z]\,[a, q\, X] \to [b, p\, Z]\,[a, Y]$,
$[b, Z\, q\, \$] \to [b, p\, Z\, \$]$,
$[b, Z]\,[a, q\, X\, \$] \to [b, p\, Z]\,[a, Y\, \$]$,
$[b, ¢\, Z]\,[a, q\, X] \to [b, ¢\, p\, Z]\,[a, Y]$,
$[b, ¢\, q\, X] \to [b, p\, ¢\, Y]$,

7) $[a,\ q\, ¢\, X\, \$] \to [a, ¢\, p\, X\, \$]$,
$[a, ¢\, q\, X\, \$] \to [a, ¢\, Y\, p\, \$]$,
$[a, ¢\, X\, q\, \$] \to [a, ¢\, p\, X\, \$]$,
$[a, ¢\, q\, X\, \$] \to [a,\ p\, ¢\, Y\, \$]$.

**8)** $[a, \alpha q \beta] \to a$,

za sve znakove $a$ iz skupa $\Sigma \backslash \{¢, \$\}$ i za sve nizove $\alpha$ i $\beta$, gdje su $\alpha$ i $\beta$ u skupu $\Sigma^*$. Nizovi $\alpha$ i $\beta$ su konačni i čine ih najviše jedan znak trake i dva graničnika. Budući da je konačni broj znakova $a$ i konačni broj različitih nizova $\alpha q \beta$, konačni broj je i produkcija oblika (8).

Ako je do nezavršnog znaka završni znak, onda se na temelju nezavršnog znaka generira završni znak sljedećim produkcijama:

**9)** $[a, \alpha]\, b \to a\, b$,
**10)** $b\, [a, \alpha] \to b\, a$,

za sve znakove $a$ i $b$ iz skupa $\Sigma \backslash \{¢, \$\}$ i za sve nizove $\alpha$.

Budući da su desne strane produkcija dulje ili jednako dugačke kao lijeve strane, izgrađene produkcije zadovoljavaju zahtjeve kontekstno ovisne gramatike.

3. Dokazati da su konteksno ovisni jezici zatvoreni s obzirom na nadovezivanje.
Str 170.

Neka kontekstno ovisne gramatike $G_1=(V_1, T_1, P_1, S_1)$ i $G_2=(V_2, T_2, P_2, S_2)$ generiraju kontekstno ovisne jezike $L(G_1)$ i $L(G_2)$. Pretpostavlja se da su produkcije obje gramatike $G_1$ i $G_2$ u normalnom obliku i da je presjek skupova nezavršnih znakova $V_1$ i $V_2$ prazni skup. Za gradnju kontekstno ovisne gramatike $G_4=(V_4, T_4, P_4, S_4)$ koja generira jezik $L(G_4)=L(G_1)L(G_2)$ koristi se postupak opisan u odjeljku 3.3.1.

Da se izgradi gramatika $G_4$, nije dovoljno osigurati da je presjek skupova nezavršnih znakova $V_1$ i $V_2$ prazni skup. Gramatika $G_4$ generira međunizove $\gamma\delta$ na sljedeći način: $S_4 \underset{G_4}{\Rightarrow} S_1S_2 \underset{G_1}{\overset{*}{\Rightarrow}} \gamma S_2 \underset{G_2}{\overset{*}{\Rightarrow}} \gamma\delta$. Na primjer, neka je sufiks niza $\gamma$ niz završnih znakova $\alpha_1$, neka je prefiks niza $\delta$ niz $A\alpha_2$ i neka je $\alpha_1 A\alpha_2 \to \alpha_1\beta\alpha_2$ produkcija gramatike $G_2$. Spajanjem susjednih znakova međunizova $\gamma$ i $\delta$ nastaje niz $\alpha_1 A\alpha_2$ koji je desna strana produkcije $\alpha_1 A\alpha_2 \to \alpha_1\beta\alpha_2$. Primjenom produkcije $\alpha_1 A\alpha_2 \to \alpha_1\beta\alpha_2$ na međuniz koji nastaje spajanjem znakova međunizova $\gamma$ i $\delta$ moguće je generirati niz koji nije u jeziku $L(G_1)L(G_2)$.

Budući da nije moguće osigurati da je presjek skupova završnih znakova $T_1$ i $T_2$ prazni skup, definira se sljedeće ograničenje na oblik produkcija kontekstno ovisne gramatike: na lijevoj strani produkcija su isključivo nezavršni znakovi gramatike. Produkcije bilo koje kontekstno ovisne gramatike $G=(V, T, P, S)$ moguće je preurediti tako da zadovolje postavljeni zahtjev. Konstruira se gramatika $G'=(V', T, P', S)$ na sljedeći način. Za sve završne znakove $a \in T$ definiraju se novi nezavršni znakovi $A_a \in V'$. U skupu nezavršnih znakova $V'$ su svi nezavršni znakovi iz skupa $V$ i novi nezavršni znakovi $A_a$. U skup produkcija $P'$ stave se produkcije $A_a \to a$ i produkcije $\alpha' \to \beta'$. Produkcije $\alpha' \to \beta'$ grade se na temelju produkcija $\alpha \to \beta$ iz skupa $P$ tako da se u nizovima $\alpha$ i $\beta$ završni znakovi $a$ zamjene nezavršnim znakovima $A_a$.

Ako su na lijevoj strani produkcije isključivo nezavršni znakovi i ako je presjek skupova nezavršnih znakova $V_1$ i $V_2$ prazni skup, onda nije moguće da se spajanjem znakova bilo koja dva međuniza $\gamma$ i $\delta$ za koje vrijedi $S_1 \underset{G_1}{\overset{*}{\Rightarrow}} \gamma$ i $S_2 \underset{G_2}{\overset{*}{\Rightarrow}} \delta$ dobije desna strana jedne od produkcija iz skupova $P_1$ i $P_2$.

Kontekstno ovisni jezici zatvoreni su s obzirom na Kleeneov operator $L^*$.

# Razredba jezika, automata i gramatika

**1.** Opišite Chomskijevu hijerarhiju jezika, gramatika i automata.

Klasa regularnih jezika jest pravi podskup klase determinističkih kontekstno neovisnih jezika.

Klasa determinističkih kontekstno neovisnih jezika jest pravi podskup klase nedeterminističkih kontekstno neovisnih jezika.

Klasa nedeterminističkih kontekstno neovisnih jezika jest pravi podskup klase kontekstno ovisnih jezika.

Klasa kontekstno ovisnih jezika jest pravi podskup klase rekurzivnih jezika.

Klasa rekurzivnih jezika jest pravi podskup klase rekurzivno prebrojivih jezika.

Klasa rekurzivno prebrojivih jezika jest pravi podskup skupa svih jezika zadanih nad abecedom Σ.

Istovjetnost regularnih jezika, konačnih automata i regularne gramatike.

Istovjetnost kontekstno neovisnog jezika, kontekstno neovisne gramatike i potisnog automata.

Istovjetnost kontekstno ovisnog jezika, kontekstno ovisne gramatike i linearno ograničenog automata.

Istovjetnost rekurzivno prebrojivih jezika, gramatike neograničenih produkcija i Turingovog stroja.

**2.** (3 boda) Navedite primjer po jednog jezika za svaku od zadanih klasa koji **ne** pripada niti jednoj užoj klasi od ovdje navedenih klasa (0.5 bodova za svaki točan odgovor).

REGULARAN JEZIK

1. L(M1) = {} nad abecedom ∑ = {0,1} prihvaća DKA M1:
M1 = ( {q} , {0,1} , {δ(q,0) = q , δ(q,1 ) =q} , q , {})

2. L(M2) = {ε} nad abecedom ∑ = {0,1} prihvaća DKA M2:
M2 = ( {q,p} , {0,1} , {δ(q,0) = p , δ(q,1 ) =p , δ(p,0) = p , δ(p,1 ) =p}, q , {q})

3. L(M3) = ∑* nad abecedom ∑ = {0,1} prihvaća DKA M3:
M3 = ( {q} , {0,1} , {δ(q,0) = q , δ(q,1 ) =q} , q , {q})

DETERMINISTIČKI KONTEKSTNO-NEOVISNI JEZIK

- valjda oni koje prihvaća potisni automat, kaže wikipedija, ako je tako onda
Moze i anbn ( n je exponent )
L(M) = {(na)m | n ≥ 1, m ≥ 0 , n ≤ m} n,m eksponenti
Provjeriti i N (M) = {w2wR | w npr. Niz 0 i 1 , tj ( 0 + 1 )* ,a wR obrnuto, dvojka omogućuje donošenje odluke}

NEDETERMINISTIČKI KONTEKSTNO-NEOVISNI JEZIK

N ( M ) = { wwR | w npr. Niz 0 i 1 , tj ( 0 + 1 )* ,a wR obrnuto }
Općenito palindromi

REKURZIVNI JEZIK

Jezik kojega prepoznaje TS

{ww | w ϵ Σ* }
{ wcy |w ϵΣ* , w različito y }

REKURZIVNO PREBROJIV JEZIK

Jezik koji TS prihvaća,
- Lu -> Univerzalan
- L(M) = { a^ib^jc^k | i≠j ili j≠k }
- L(M) = { 0^n 1^n | n>0 }
- L(M) = { wcy | w,yeΣ*, w≠y }

NEIZRAČUNLJIV

L = {ai : ai∉ Mi } , jezik za koji ne postoji TS koji ga prihvaća, ako netko može pronaći konkretan primjer nek javi , i je exponent

- Ld -> Dijagonalan

**3.** Definirati nejednoznačnost niza, nejednoznačnost gramatike i inherentnu nejednoznačnost jezika.

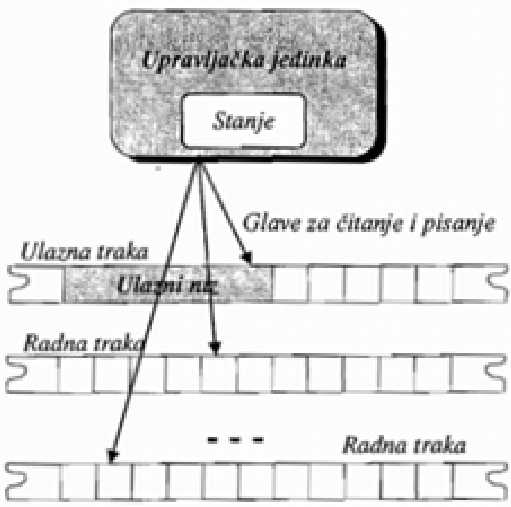
**4.** Opisati model kojim se ocjenjuje vremenska složenost prihvaćanja jezika.

*Str 182.*

Slika 6.4 prikazuje model determinističkog TS koji se koristi za ocjenu vremenske složenosti prihvaćanja jezika. TS ima $k$ dvostrano beskonačnih traka. Jedna radna traka jest ulazna i na njoj je zapisan ulazni niz $w$ duljine $n$. Sve trake, uključujući i ulaznu traku, moguće je čitati i po svim trakama moguće je pisati.

Vremenska složenost računa se na temelju broja pomaka glave za čitanje i pisanje:

Izvede li TS $M$ najviše $T(n)$ pomaka glave tijekom prihvaćanja bilo kojeg niza duljine $n$, TS $M$ jest vremenske složenosti $T(n)$. Jezik $L=L(M)$ koji prihvaća TS $M$ jest vremenske složenosti $T(n)$.




**5.** Dokazati istovjetnost jezika koje je moguće generirati kanonskim slijedom i rekurzivnih jezika.

Rekurzivne jezike moguće je generirati kanonskim slijedom.

Ako je jezik $L(M_2)$ rekurzivan, onda je za generiranje jezika moguće koristiti jednostavni TS. Jednostavni TS generira nizove jezika $L(M_2)$ na izlaznu traku onim redoslijedom kojim se ti nizovi generiraju na radnu traku. Generiraju li se nizovi na radnu traku kanonskim slijedom, jednostavni TS $M_1$ generira nizove jezika $L(M_2)$ kanonskim slijedom na izlaznu traku.

| | |
|---|---|
| **6.** | Opisati utjecaj broja traka Turingovog stroja na prostornu složenost. |

***Str 183.***

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 $L(M_2)=L(M_1)$.

| | |
|---|---|
| **7.** | Opišite utjecaj broja traka Turingovog stroja na vremensku složenost izračunavanja. |

***Str 184.***

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 $L(M_2)=L(M_1)$.

| | |
|---|---|
| **8.** | Opisati postupak ubrzanja Turingovog stroja za konstantni faktor. |

**Str 184. i 185.**

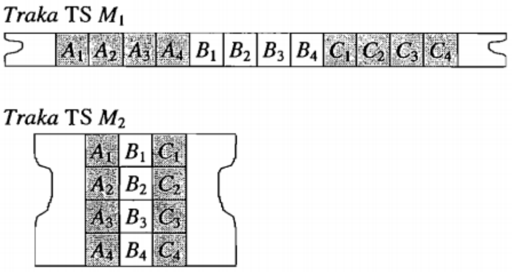
Ako TS $M_1$ s $k$ traka vremenske složenosti $T(n)$ prihvaća jezik $L(M_1)$, onda za bilo koju konstantu $c>0$ postoji TS $M_2$ s $k$ traka vremenske složenosti $cT(n)$ koji prihvaća jezik $L(M_2)=L(M_1)$, gdje je $k>1$ i $\inf_{n\to\infty} T(n)/n=\infty$. Funkcija $\inf_{n\to\infty} f(n)$ je najveća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

TS $M_2$ ima barem dvije trake ($k>1$). Jedna traka jest ulazna, a ostale su radne. TS $M_2$ čita ulazni niz znak po znak s ulazne trake, kodira $m$ susjednih znakova ulaznog niza u jedinstveni znak, te dobiveni jedinstveni znak zapiše na izabranu radnu traku. Prepisivanje i sažimanje se nastavlja sve dok se cijeli niz s ulazne trake ne prepiše u sažetom obliku na radnu traku. Nakon što se niz prepiše i sažme, zamijene se uloge traka: radna traka je ulazna, a ulazna je radna.

| | |
|---|---|
| **9.** | Opisati konstrukciju TS $M_2$ ekvivalentnog TS $M_1$ uz sažimanje prostora na TS $M_2$ za konstantni faktor $c$. |

***Str 184.***

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 $L(M_2)=L(M_1)$.




**Slika 6.7:** Sažimanje prostora za konstantni faktor ($r$=4)

Slika 6.7 prikazuje konstrukciju TS $M_2$. Znakovi u $r$ susjednih ćelija radnih traka TS $M_1$ kodiraju se jedinstvenim znakom TS $M_2$. Jedan znak radne trake TS $M_2$ predstavlja sadržaj $r$ ćelija TS $M_1$. Na slici 6.7 znakovi trake TS $M_2$ su složeni znakovi koji se sastoje od $r$ komponenata. Svaka komponenta ima svoj zaseban trag. Tijekom simulacije TS $M_2$ u jednu komponentu složenog stanja spremi podatak koji trag radne trake se čita. Komponenta stanja poprima vrijednosti od 1 do $r$.

| | |
|---|---|
| **10.** | (3 boda) Kolika je sigurno dovoljna prostorna složenost determinističkog Turingovog stroja $TS_1$ koji prihvaća jezik L, ako je jezik L moguće prihvatiti determinističkim Turingovim strojem $TS_2$ vremenske složenosti $f(n)$ ? Obrazložite odgovor. |

***Str 187.***

Ako je jezik $L$ u klasi DTIME($f(n)$), onda je jezik $L$ u klasi DSPACE($f(n)$).

Budući da je jezik $L$ u klasi DTIME ($f(n)$), postoji deterministički TS $M$ vremenske složenosti $f(n)$. Primjenom $f(n)$ pomaka glave TS $M$ ne može obići više od $f(n)$+1 ćelija. Sažimanjem sadržaja dviju ćelija u jednu ćeliju izgradi se TS koji ne koristi više od $\lceil (f(n)+1)/2 \rceil$ ćelija, odnosno broj ćelija koje se koriste jest manji od $f(n)$. Budući da jezik $L$ prihvaća deterministički TS koji ne koristi više od $f(n)$ ćelija, jezik $L$ je u klasi DSPACE($f(n)$).

**11.** Definirati polinomne klase složenosti jezika.

***Str 194.***

$$P = \bigcup_{i \geq 1} \text{DTIME}(n^i)$$

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

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

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

Klase jezika s obzirom na prostornu polinomnu složenost su:

$$\text{PSPACE} = \bigcup_{i \geq 1} \text{DSPACE}(n^i)$$

$$\text{NSPACE} = \bigcup_{i \geq 1} \text{NSPACE}(n^i)$$

12. Dokazati da ako se jezik L′ polinomski svodi na jezik L tada je jezik L′ u klasi P ako je jezik L u klasi P.

**Str 197.**

*a)* Ako je jezik $L_2$ u klasi $P$, onda je i jezik $L_1$ u klasi $P$

*b)* Ako je jezik $L_2$ u klasi $NP$ , onda je i jezik $L_1$ u klasi $NP$ .

Dokazi za oba slučaja su slični, pa je dan dokaz samo za slučaj (*a*). Neka je TS $M_R$ koji svodi jezik $L_1$ na jezik $L_2$ polinomne vremenske složenosti $p_1(n)$, a TS $M_2$ koji prihvaća jezik $L_2$ neka je polinomne vremenske složenosti $p_2(n)$. Dokazuje se da je TS $M_1$ koji prihvaća jezik $L_1$ polinomne vremenske složenosti $p_1(n)+p_2(p_1(n))$. TS $M_R$ pomakne glavu najviše $p_1(n)$ puta za generiranje izlaznog niza $y \in L_2$, gdje je $y=R(x)$ i $x \in L_1$. Budući da je moguće generirati najviše jedan znak jednim pomakom glave, duljina niza $y$ mora biti kraća od maksimalnog broja pomaka glave $p_1(n)$, odnosno $|y| \leq p_1(n)$. Prihvaćanje niza $y$ iz jezika $L_2$ moguće je obaviti u vremenu $p_2(p_1(n))$, jer je vremenska složenost TS $M_2$ jednaka $p_2(n)$, a duljina niza je kraća od $p_1(n)$. Zaključuje se da je ukupno potrebno vrijeme da se prihvati jezik $L_1$ jednako $p_1(n)+p_2(p_1(n))$, odnosno TS $M_1$ koji prihvaća jezik $L_1$ je vremenske složenosti $p_1(n)+p_2(p_1(n))$, što je polinomna funkcija argumenta $n$.

**13.** Jezik $L_1$ svodi se u logaritamskom prostoru na jezik $L_2$. Dokazati da ako je $L_2$ u klasi $P$ onda je i $L_1$ u klasi $P$.

***Str 197.***

*c)* Ako je jezik $L_2$ u klasi $P$, onda je i jezik $L_1$ u klasi $P$.

Dokaz za slučaj (*c*) izvodi se u dva koraka. Najprije se dokazuje da determinističko logaritamsko prostorno svođenje jezika ne troši više od determinističkog polinomnog vremena. U odjeljku 6.2.3 dokazano je da jezik $L$ koji je u klasi DSPACE ($f(n)$) je u klasi DTIME($c^{f(n)}$). Uvrsti li se za $f(n)$ da je jednako $\log_b n$, dokazuje se da je jezik $L$ polinome vremenske složenosti, odnosno da je jezik u klasi DTIME($n^k$):

$$c^{f(n)} = c^{\log_b n} \leq (b^k)^{\log_b n} = b^{k \log_b n} = (b^{\log_b n})^k = n^k$$

# Zadaci na zaokruživanje

(1 bod) Označite koja od navedenih produkcija je u Chomskyjevom normalnom obliku:

**a)** $A \to aB$ **b)** $A \to a$ **c)** $A \to aBCD$ **d)** $A \to \varepsilon$ **e)** $A \to aaB$

(1 bod) Uvjet podudarnosti za stanja $p$ i $q$ jest::

**a)** $(p \in F \wedge q \in F) \vee (p \notin F \wedge q \notin F)$ **b)** $(p \notin F \wedge q \in F) \vee (p \in F \wedge q \notin F)$ **c)** $(p \in F \wedge q \notin F) \vee (p \notin F \wedge q \in F)$
**d)** $\delta(p, a)$ i $\delta(q, a)$ su istovjetna stanja **e)** $\delta(p, a)$ i $\delta(q, a)$ su prihvatljiva stanja

(1 bod) Za NKA koji ima $p$ stanja, $n$ ulaznih znakova, $m$ prihvatljivih stanja $(m < p)$ gradi se istovjetan DKA koji ima najviše:

**a)** $p!$ stanja **b)** $p \cdot m$ stanja **c)** $2^p$ stanja **d)** $p \cdot m \cdot n$ stanja **e)** $p^n$ stanja

(1 bod) Postupak odbacivanja beskorisnih znakova provodi se odbacivanjem:

**a)** mrtvih znakova pa nedohvatljivih znakova **b)** nedohvatljivih znakova pa mrtvih znakova **c)** $\varepsilon$- produkcija pa jediničnih produkcija **d)** jediničnih produkcija pa $\varepsilon$- produkcija **e)** ništa od navedenog

(1 bod) Najuža klasa jezika u kojoj se uvijek nalazi presjek kontekstno neovisnog i regularnog jezika jest:

**a)** rekurzivno prebrojiv jezik **b)** kontekstno ovisan jezik **c)** regularan jezik **d)** kontekstno neovisan jezik
**e)** nije moguće utvrditi u općem slučaju

(1 bod) Funkcija prijelaza osnovnog modela Turingovog stroja definira se na sljedeći način:

**a)** $\delta : Q \times \Sigma \times \Gamma \to Q \times \Gamma \times \{L, R\}$ **b)** $\delta : Q \times \Sigma \to Q \times \Sigma \times \{L, R\}$ **c)** $\delta : Q \times \Sigma \to Q \times \Sigma \times \{L, N, R\}$
**d)** $\delta : Q \times \Sigma \to Q \times \Gamma \times \{L, R\}$ **e)** $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

8. (1 bod) Prema Chomskyjevoj hijerarhiji jezika kontekstno ovisni jezici su podskup:

   **a)** regularnih jezika **b)** nedeterminističkih kontekstno neovisnih jezika **c)** determinističkih kontekstno neovisnih jezika **d)** rekurzivno prebrojivih jezika **e)** niti jedan odgovor nije točan

9. (1 bod) Odredite koji niz pripada jeziku opisanom regularnim izrazom: $((a+b)^*)^+ c^* d^+ (e^+ + a^*)^+$

   **a)** $babcae$ **b)** $cbbddaea$ **c)** $babbdeea$ **d)** $abbcaa$ **e)** $babbdccaa$

10. (1 bod) Odrediti **razred najjednostavnijeg oblika** formalnog automata kojim je moguće prihvatiti jezik: $ww^R$, ako je $w = (0+1+2)^*3$.

    **a)** deterministički potisni automat **b)** nedeterministički potisni automat **c)** Turingov stroj **d)** LOA
    **e)** konačni automat

11. (1 bod) Odrediti **razred najjednostavnijeg oblika** formalnog automata kojim je moguće prihvatiti jezik: $a^i b^{2(i+k)} c^k$, $0 \le i \le N, 0 \le k \le M$, pri čemu su $N$ i $M$ cjelobrojne konstante.

    **a)** deterministički potisni automat **b)** nedeterministički potisni automat **c)** Turingov stroj **d)** LOA
    **e)** konačni automat

12. (1 bod) Odrediti **razred najjednostavnijeg oblika** formalnog automata kojim je moguće prihvatiti jezik: $ww$ , ako je $w = (0+1+2)^+$.

    **a)** deterministički potisni automat **b)** nedeterministički potisni automat **c)** Turingov stroj **d)** LOA
    **e)** konačni automat

13. (1 bod) Kolika je vremenska složenost prihvaćanja jezika $L = \{wcw^R | w \in (a+b)^*\}$ u ovisnosti o duljini niza $n$.

    **a)** $n+1$ **b)** $n$ **c)** $n^2$ **d)** 2n+1 **e)** $n \cdot (n+1)/2$

(1 bod) Zadan je TS sa 16 radnih traka koji s prostornom složenošću $n^2$ prihvaća neki jezik $L$. Tada postoji TS s 8 radnih traka koji prihvaći isti jezik $L$ s prostornom složenošću (odabrati najsporije rastuću funkciju koja zadovoljava uvjete zadatka):

**a)** $n$ **b)** $n^4$ **c)** $n^8$ **d)** $n^2$ **e)** $n^2 \cdot \log n$

(1 bod) Neka je zadan NP-potpun jezik $L_1$ i neki jezik $L_2$. Ako je jezik $L_1$ moguće u polinomnom vremenu svesti na jezik $L_2$, za jezik $L_2$ možemo zaključiti:

**a)**postoji deterministički TS koji $L_2$ prihvaća u polinomnom vremenu **b)** postoji nedeterministički TS koji $L_2$ prihvaća u polinomnom vremenu **c)**jezik $L_2$ je NP-potpun **d)** jezik $L_2$ je NP-težak **e)** ništa od navedenog

(1 bod) Kontekstno ovisni jezici zatvoreni su s obzirom na (odabrati najveći točan skup):

**a)** unija, nadovezivanje, Kleenov operator $L^+$, presjek **b)**unija, nadovezivanje, presjek
**c)** Kleenov operator $L^+$, unija, nadovezivanje **d)** unija, nadovezivanje **e)** unija i presjek

.

(3 boda) Neka je $T(L)$ skup svih onih Turingovih strojeva (TS) koji prihvaćaju jezik $L$, a $w$ je niz čija prihvatljivost se ispituje Turingovim strojem. Koja od sljedećih tvrdnji (1-5) vrijedi za zadane scenarije prihvaćanja jezika (a-f): **Odgovor upišite na praznu liniju.**

(a) L je regularni jezik, niz w JEST u jeziku L ________

(b) L je regularni jezik, niz w NIJE u jeziku L ________

(c) L je rekurzivni jezik, niz w JEST u jeziku L ________

(d) L je rekurzivni jezik, niz w NIJE u jeziku L ________

(e) L je rekurzivno-prebrojiv jezik, niz w JEST u jeziku L ________

(f) L je rekurzivno-prebrojiv jezik, niz w NIJE u jeziku L ________

(1) svaki TS iz skupa T se zaustavlja

(2) postoji barem jedan TS u skupu T koji se zaustavlja, ali ne nužno svaki

(3) sigurno ne postoji niti jedan TS u skupu T koji se zaustavlja

(4) moguce je da ne postoji niti jedan TS u skupu T koji se zaustavlja

(5) zadanu klasu jezika nije moguće prihvaćati Turingovim strojem