2a.***Opisati postupak konstrukcije potisnog automata koji prihvaća praznim stogom (M2) na osnovi zadanog potisnog automata koji prihvaća prihvatljivim stanjem (M1)***

Ako je M1 = (Q, ∑, Г, δ, q0, Z0, F), gdje je:

Q – skup stanja  
 ∑ - skup ulaznih znakova  
 Г – skup znakova stoga  
 δ – funkcija prijelaza  
 q0 – pocetno stanje  
 Z0 – pocetni znak stoga  
 F – skup prihvatljivih stanja

Gradi se M2 = (Q U {q0', qe}, ∑, Г U {X0}, δ', q0', X0, {})

Konstrukcija PA M2, zasniva se na simulaciji PA M1. Ude li tokom simulacije PA M1 u jedno od prihvatljivih stanja, PA M2 isprazni svoj stog. Da bi se to omogućilo koristi se dodatni znak stoga X0. Na početku rada PA M2 stavi na dno stoga znak X0, PA M1 svojim prijelazima ne može uzeti sa stoga znak X0. Ako se stog isprazni, a ne ude se u jedno od prihvatljivih stanja na stogu ostaje znak X0 koji onemogucava M2 da prihvati niz.

Rezime: dodaju se stanja q0' i qe u skup stanja, X0 u skup znakova stoga, q0' je novo pocetno stanje, X0 je novi pocetni znak stoga, te se maknu sva stanja iz skupa prihvatljivih stanja

1. Na početku rada PA M2 prelazi iz svoje početne konfiguracije u početnu konfiguraciju PA M1 i ostavlja znak X0 na dnu stoga (onaj prijelaz (q0', eps, X0)=(q0, Z0X0))
2. Svi prijelazi iz skupa prijelaza delta od M1 stave se u skup prijelaza od M2. Ovi prijelazi omogucuju simuliranje M1
3. Dodaju se eps-prijelazi za sva stanja iz skupa prihvatljivih stanja F od M1 i za sve znakove iz stoga od M2. Ovo je kada se ude u neko od prihvatljivih stanja da se prede u stanje qe

(prijelazi tipa (q1, eps, neki znak stoga) = (qe, eps)) (plus skine se jedan znak sa stoga)

1. Dodaju se eps prijelazi za stanje qe i svaki znak stoga, za praznjenje stoga

(prijelazi tipa (qe, eps, neki znak stoga) = (qe, eps))

2b.***Opisati postupak konstrukcije potisnog automata koji prihvaća prihvatljivim stanjem (M2) na osnovi zadanog potisnog automata koji prihvaća praznim stogom (M1)***

Ako je M1 = (Q, ∑, Г, δ, q0, Z0, {}), gdje je:

Q – skup stanja  
 ∑ - skup ulaznih znakova  
 Г – skup znakova stoga  
 δ – funkcija prijelaza  
 q0 – pocetno stanje  
 Z0 – pocetni znak stoga

Gradi se M2 = (Q U {q0', qf}, ∑, Г U {X0}, δ', q0', X0, { qf })

Izgradeni PA M2 simulira rad PA M1, te prelazi u prihvatljivo stanje ako i samo ako PA M1 isprazni svoj stog.

Rezime: dodaju se stanja q0' i qf u skup stanja, X0 u skup znakova stoga, q0' je novo pocetno stanje, X0 je novi pocetni znak stoga, qf postaje prihvatljivo stanje

1. Na početku rada PA M2 prelazi iz svoje početne konfiguracije u početnu konfiguraciju PA M2. Na dno stoga se stavlja X0 (onaj prijelaz (q0', eps, X0)=(q0, Z0X0))
2. Svi prijelazi iz skupa prijelaza delta od M1 stave se u skup prijelaza od M2. Ovi prijelazi omogucuju simuliranje M1
3. U skup prijelaza dodaju se eps prijelazi za svako stanje iz skupa stanja za vrh stoga X0, za prelazak u prihvatljivo stanje qf. To znaci procita li se na vrhu stoga znak X0, to je znak da je PA M1 ispraznio svoj stog i prelazi se u prihvatljivo stanje.

(prijelazi tipa (neko stanje iz Q, eps, X0) = (qf, eps))

**3.a *Opisati postupa konstrukcije gramatike kojom se dokazuje da su konteksno-neovisni jezici zatvoreni s obzirom na proporciju unije***

Gramatika G3=(V3, T3, P3, S3) koja generira jezik L(G1)UL(G2) konstruira se na sljedeci nacin:

1. V3 = V1UV2U{S3} S3 nije element niti V1 niti V2, V1 presjek V2 je prazan niz
2. T3 = T1 U T2
3. u P3 = P1 U P2 dodaju se produkcije:

S3 -> S1|S2

**3.b *Opisati postupa konstrukcije gramatike kojom se dokazuje da su konteksno-neovisni jezici zatvoreni s obzirom na proporciju nadovezivanja***

Gramatika G3=(V3, T3, P3, S3) koja generira jezik L(G1)L(G2) konstruira se na sljedeci nacin:

1. V3 = V1UV2U{S3} S3 nije element niti V1 niti V2, V1 presjek V2 je prazan niz
2. T3 = T1 U T2
3. u P3 = P1 U P2 dodaju se produkcije:

S3 -> S1S2

**3.c *Opisati postupa konstrukcije gramatike kojom se dokazuje da su konteksno-neovisni jezici zatvoreni s obzirom na proporciju kleenovog operatora***

Gramatika G2=(V2, T2, P2, S2) koja generira jezik L(G2)=L(G1)\* konstruira se na sljedeci nacin:

1. V2 = V1U{S2} S2 nije element od V1
2. T2= T1
3. u P3 = P1 U P2 dodaju se produkcije:

S2 -> S1S2|eps

***4.Opisati algoritam za izbacivanje mrtvih znakova iz konteksno-neovisne gramatike***

Ako su zivi svi znakovi desne strane produkcije onda je ziv i nezavrsni znak lijeve strane produkcije

1. u listu zivih znakova stave se lijeve strane produkcija koje s desne strane nemaju nezavrsnih znakova
2. ako su s desne strane produkcije iskljucivo zivi znakovi nezavrsni znak lijeve strane je zivi znak i stavlja se u listu zivih znakova
3. ako nije moguce prosiriti listu zivih znakova algoritam se zaustavlja, svi znakovi koji nisu u listi zivih su mrtvi znakovi

***5.Opisati algoritam za izbacivanje nedohvatljivih stanja iz konteksno-neovisne gramatike***

Ako je dohvatljiv nezavrsni znak lijeve strane produkcije onda su dohvatljivi i svi zavrsni i nezavrsni znakovi desne strane produkcije

1. u listu dohvatljivih stavi se početni nezavršni znak
2. ako je znak lijeve strane produkcije dohvatljiv onda su svi znakovi desne takoder dohvatljivi i dodaju se u listu dohvatljivih
3. ako listu dohvatljivih nije moguce prosiriti algoritam se zaustavlja, svi znakovi koji nisu u listi dohvatljivih su nedohvatljivi

***2) Definirati nejednoznačnost kontekstno-neovisne gramatike***.

Ako je moguce za neki niz w element iz L(G) izgraditi više različitih generativnih stabala, onda je kontekstno neovisna gramatika G nejednoznačna.

Ako je moguce neki niz w generirati primjenom više različitih postupaka generiranja niza zamjenom kranje lijevog nezavršnog znaka ili primjenom više različitih postupaka generiranja niza zamjenom kranje desnog nezavršnog znaka onda je gramatika G nejednoznačna.

***5) Formalno definirati osnovni model Turingovog stroja.***

Turingov stroj formalno se zadaje uređenom sedmorkom

ts = (Q, ∑, Г, δ, q0, B, F)

Г-konačan skup znakova trake  
B element iz Г – znak kojim se označava prazna ćelija

***2.a napisi sto je i kako se razrjesava jedinicna produkcija***

Jedinicna produkcija je oblika : A -> B  
Dakle jedan nezavršni znak s lijeve i desne strane produkcije

1. U skup produkcija gramatike G' stave se sve produkcije gramatike G koje nisu jedinične
2. Za sve jedinične produkcije oblika A -> B , uz produkcije oblika B -> α koje nisu jedinične, grade se produkcije oblika A -> α

***2.b napisi sto je i kako se razrjesava epsilon produkcija***

Epsilon produkcija je oblika A -> eps

1. Pronadu se svi nezavrsni znakovi koji generiraju prazni niz, tzv prazni znakovi.
2. U skup produkcija dodaju se produkcije tipa:

A -> ξ1ξ2…ξn

gdje ξi poprima:

1. Ako znak Xi nije prazni znak, onda je oznaka ξi jednaka Xi
2. Ako znak Xi jest prazan znak, onda je oznaka ξi jednaka eps ili Xi

DODATNA PITANJA (moguca)

1. Što je to kontekstno neovisan jezik

Neki jezik je kontekstno neovisan ako i samo ako postoji kontekstno neovisna gramatika koja ga generira

Za bilo koji kontekstno neovisan jezik moguce je izgraditi kontekstno neovisnu gramatiku koja ga generira i obrnuto

Klasa kontekstno neovisnih jezika sadrži kao podskup klasu regularnih jezika

1. Chomskyjev normalni oblik produkcija

ima oblik : A -> BC ili A -> a

1. U skup produkcija stave se sve produkcije koje vec jesu u Chomskyjevom normalnom obliku
2. ako je Xi završni znak a, a element od T, onda se skup nezavršnih znakova proširi novim nezavršnim znakom Ca, skup produkcija P se proširuje produkcijom:

Ca -> a

Svi završni znakovi a zamijene se nezavršnim znakom Ca

1. Za produkcije koje imaju tri ili više nezavršnih znakova na desnoj strani:

A -> B1B2….Bm m>=3

definiraju se novi nezavršni znakovi D1, D2, … Dm-2

a zatim se produkcija A -> B1B2…Bm zamijeni skupom

{A->B1D1 ; D1->B2D2…..Dm-2->Bm-1Bm}

1. Greibachov normalni oblik produkcija

ima oblik A->aα, gdje je α- niz nezavršnih znakova (moze biti više njih, jedan ili nijedan), a je završni znak

1. Parsiranje od vrha prema dnu ide od početnog nezavršnog znaka (korijena) prema završnim znakovima (listovima). Parsiranje od dna prema vrhu ide obrnutim putem
2. LR(k) parser koristi metodu parsiranja od dna prema vrhu. L znači da se niz w čita s lijeva na desno, znak R označava da se stablo gradi obrnutim postupkom generiranja niza zamjenom kranje DESNOG nezavršnog znaka a broj k određuje da je potrebno pročitati najviše k znakova unaprijed za potrebe donošenja odluke o primjeni redukcije