**Teorijska pitanja iz teorije algoritama, jezika i automata – II deo**

**1. Definisati proširenu funkciju prelaza ˆδ za nederministički konačni automat sa ε–prelazima (ε–NKA). Definisati jezik ε–NKA.**

Proširena funkcija prelaza za ε–NKA A=(Q, Σ, δ, q0, F) je funkcija ˆδ(q,w), gde je qϵQ, wϵΣ\*. Vrednost funkcije ˆδ za argumente q i w je skup stanja do kojih se dospeva ako se počne iz stanja q i procesira se reč w. Komentar: Eclose(q) je skup svih stanja automata A u koja se može dospeti iz stanja q samo preko praznih (spontanih) prelaza.

1. ˆδ(q,w) = Eclose(q), ∀qϵQ
2. Pretpostavimo da je moguće utvrditi ˆδ(q,w) za wϵΣ\*, |w|=n
3. |w|=n+1, w=xa, |x|=n, aϵΣ, ˆδ(q,w) = {p1, ... ,pm}

ˆδ(q,w) = ˆδ(q,xa) = Eclose(S) gde je S = U δ(qi,w)

Jezik ε–NKA: L(A) = {wϵΣ\*| ˆδ(q0,w) ∩ F≠Ø}

**2. Neka je E = (QE, Σ, δE, q0, FE) dati ε–NKA. Pretpostavimo da je njemu ekvivalentan DKA (tj. DKA koji definiše isti jezik) automat D = (QD, Σ, δD, qD, FD). Definisati komponente QD, δD, qD i FD automata D.**

1. QD = P(QE), |QD| = 2n
2. Alfabet Σ je isti
3. δD: QD X Σ -> QD tj P(QE) X Σ -> P(QE)

S ⊆ QE, δD(S,a) = U δE(p,a)

1. Startno stanje {qo}
2. Ako S ⊆ QE, onda SϵFD ako S∩FE ≠Ø tj FD = { S ϵ P(QE)| S∩FE≠Ø}

**3. Važi teorema: Ako je L jezik i L = L(A) za neki DKA A, tada postoji regularan izraz R takav da L = L(R). Teorema se dokazuje konstruisanjem skupa regularnih izraza Rkij koji odgovaraju putanjama u dijagramu prelaza datog DKA.**

**(a) Sta tačno predstavlja regularan izraz Rkij?**

Rkij označava regularan izraz čiji jezik se sastoji od svih stringova koji se formiraju putanjama od čvora i do čvora j u automatu A takvih da ne sadrže čvorove sa većom oznakom od k.

(b) Konstruisati induktivno kolekciju Rkij za sve k ∈ {0, 1, . . . , n}. (Dakle, treba definisati skup R0ij, ovo je baza indukcije, i Rkij preko Rlij za l < k.)

(c) Na osnovu konstruisane kolekcije izraza Rkij definisati regularan izraz koji odgovara jeziku automata.

4. Važi teorema: Ako je L jezik i L = L(R) za neki regularan izraz R, tada postoji ε–NKA E takav da L = L(E). Dokazati teoremu konstruisanjem traženog ε–NKA E za dati regularan izraz R. Napomena: ε–NKA E konstruiše se induktivno, prema složenosti datog regularnog izraza.

**5. Dokazati pumping–lemu za regularne jezike.**

Neka je L regularan jezik. Budući da je regularan, postoji DKA, takav da L=L(A). Neka je p broj stanja ovog automata. Neka je wϵL, |w|≥p; w=a1a2,...,am za a1a2,...,am ϵΣ, m≥p.

ˆδ(q,w) – rezultat je jedno stanje jer je u pitanju DKA.

ˆδ(q0, ε) = q0 = p0 , ˆδ(q0, a1) = p1 , ˆδ(q0, a1a2) = p2 .... ˆδ(q0, a1a2,...,am) = pm

Poređajmo dobijena stanja u niz: p0, p1,...,pm -> m+1 stanje automata A. Pošto m≥p, sigurno važi m+1≥p (neka stanja su jednaka). Pretpostavimo da pi = pj, za 0 ≤ i < j ≤ m.

ˆδ(q0, a1...ai) = pi

ˆδ(q0, a1...ai...aj) = pj = pi

w = a1a2,...,am , w = a1...aiai+1...ajaj+1...am dakle w=xykz ϵ L, k≥0.

**6. Neka je L jezik palindroma nad {0, 1}, odnosno L = {w ∈ {0, 1} ∗ | w = w R }. Primenom pumping–leme za regularne jezike pokazati da ovaj jezik nije regularan.**

Pretpostavimo suprotno, L jeste regularan jezik. Na osnovu pumping leme, postoji konstanta p koja zadovoljava uslove leme. Posmatramo sledeću reč: w = 0p10p, w je palindrom, wϵL, |w| = 2p+1≥p. Na osnovu leme, w se može zapisati u formi w=xyz, gde y≠ ε, |xy|≤p, xykz ϵ L, ∀k≥0. Zaključujemo da se xy sastoji samo od nula. Dalje, y≠ ε, zaključujemo da y sadrži bar jednu nulu. Posmtramo reč xz. Ova reč neće biti palindrom, odnosno ne važi pumping lema. Jezik nije regularan.

**7. Neka je L jezik palindroma nad {0, 1}, odnosno L = {w ∈ {0, 1} ∗ | w = w R }. Pokazati da je L kontekst–slobodan jezik konstrusanjem odgovarajuće kontekst–slobodne gramatike.**

Kontekst slobodne gramatike: G=(N,Σ,P,S), X->α, αϵ(N∪Σ)\*, XϵN.

1. Reči ε, 0, 1 su palindromi
2. Ako je reč w palindrom, onda su i reči 0w0 i 1w1 takođe palindromi
3. Svi palindromi nad alfabetom {0, 1} dobijaju se konačnom primenom pravila 1 i 2

Dakle, zaključujemo da su pravila izvođenja ove gramatike: S-> ε|0|1|0S0|1S1, odnosno odgovarajuća kontekst-slobodna gramatika je: G=({S}, {0,1}, {S-> ε, S-> 0, S-> 1, S-> 0S0, S->1S1}, S).

**8. Definisati potisni automat (formalno, kao uredjenu sedmorku). Detaljno objasniti funkciju prelaza, tj. šta predstavlja δ(q, a, X).**

Potisni automat je ε-NKA koji je dopunjen stekom. Akcije vezane za stek su: push, pop i zamena. Formalna definicija: Potsni automat je uređena sedmorka P=(Q, Σ, Γ, δ, q0, z0, F), pri čemu:

1. Q je konačan skup stanja
2. Σ je konačan alfabet
3. Γ je alfabet steka
4. δ je funkcija prelaza, δ: Q X (Σ∪{ε}) X Γ -> P(Q X Γ\*)
5. q0 je startno stanje
6. z0 je dno steka, z0ϵΓ (z0 nije ϵΣ)
7. F je skup završnih stanja

Funkcija prelaza ima oblik: δ(q, a, X) = {(p1,y1),...,(pk,yk)}, pri čemu je q trenutno stanje, a simbol na ulazu, X simbol na steku, pi stanje u koje se prelazi, a yi string koji se gura na stek umesto X.

**9. Neka je L jezik palindroma parne dužine, L = {wwR | w ∈ {0, 1} ∗}. Definisati potisni automat koji prihvata ovaj jezik. Obavezno prikazati i dijagram prelaza.**

Potisni automat P=(Q, Σ, Γ, δ, q0, z0, F):

1. Q = {q0, q1, q2}
2. Σ = {0, 1}
3. Γ = {z0, 0, 1}
4. δ: Q X (Σ∪{ε}) X Γ -> P(Q X Γ\*)
5. q0 je startno stanje
6. z0 je dno steka
7. F = {q2}

|  |  |  |
| --- | --- | --- |
| δ(q0, 0, z0) = {(q0, 0z0)} | δ(q0, 1, 0) = {(q0, 10)} | δ(q0, ε, 1) = {(q1, 1)} |
| δ(q0, 1, z0) = {(q0, 1z0)} | δ(q0, 1, 1) = {(q0, 11)} | δ(q1, 1, 1) = {(q1, ε)} |
| δ(q0, 0, 0) = {(q0, 00)} | δ(q0, ε, z0) = {(q1, z0)} | δ(q1, 0, 0) = {(q1, ε)} |
| δ(q0, 0, 1) = {(q0, 01)} | δ(q0, ε, 0) = {(q1, 0)} | δ(q1, ε, z0) = {(q2, z0)} |

**10. Definisati pojam trenutne konfiguracije potisnog automata (ovo je ”instantaneous description” ili ID potisnog automata). Na osnovu ovoga definisati jezik datog potisnog automata P.**

Trenutna konfiguracija potisnog automata je uređena trojka (q, w, y), gde je qϵQ trenutno stanje automata, wϵΣ\* deo reči koji je ostao nepročitan na ulazu i yϵΓ\* sadržaj steka.

Ako je potisni automat P=(Q, Σ, Γ, δ, q0, z0, F), jezik ovog automata je: L(P) = {wϵΣ\*|(q0, w, z0) Ⴡ\*P (q, ε, α), gde je qϵF, αϵΓ\*}.

**11. Ako je P = (Q, Σ, Γ, δ, q0, Z0, F) potisni automat, definisati pokret ovog automata (tj. relaciju Ⴡ). Definisati i zatvorenje ove relacije.**

Pokret automata (Ⴡ): Neka δ(q, a, X) sadrži par (p,α). Tada za sve stringove wϵΣ\* i βϵΓ\* važi sledeće: (q, aw, Xβ) -> (p, w, αβ).

Zatvorenje Ⴡ\*: Ako je I trenutna konfiguracija nekog PDA, onda:

1. I Ⴡ\* I (0 koraka)
2. Ako su I i J dve trenutne konfiguracije ovog PDA, onda I Ⴡ\*J ako postoji konačan niz trenutnih konfiguracija I = I1, I2, I3, ..., In = J, tako da I = I1 Ⴡ I2 Ⴡ...Ⴡ In = J.

**12. Opisati razliku izmedju prihvatanja pomoću završnog stanja i prihvatanja pomoću praznog steka kod potisnih automata. Ako posmatramo potisni autmat P, definisati jezike L(P), jezik prihvaćen pomoću završnog stanja, i N(P), jezik prihvaćen pomoću praznog steka. Da li su, u opštem slučaju, ovi jezici jednaki?**

Kod prihvatanja pomoću završnog stanja prihvata se ulaz nakon čije obrade će se automat naći u nekom od završnih stanja. Kod prihvatanja pomoću praznog steka prihvataju se oni ulazi čijom obradom se dolazi do pražnjenja steka.

Ako je P potisni automat P=(Q, Σ, Γ, δ, q0, z0, F), definišemo jezik ovog automata:

* L(P) = {wϵΣ\*|(q0, w, z0) Ⴡ\*P (q, ε, α), gde je qϵF, αϵΓ\*}, potisni automat P prihvata jezik L(P) završnim stanjem
* N(P) = {wϵΣ\*|(q0, w, z0) Ⴡ\*P (q, ε, ε), gde je qϵQ}, potisni automat P prihvata jezik N(P) praznim stekom

U opštem slučaju L(P) ≠ N(P).

**13. Važi teorema: Ako je PN = (Q, Σ, Γ, δN, q0, Z0) potisni automat, i L = N(PN), tada postoji potisni automat PF takav da L = L(PF). Ako pretpostavimo da je automat PN dat, definisati (formalno, kao uredjenu sedmorku) automat PF . Predstaviti PF apstraktnim dijagramom prelaza.**

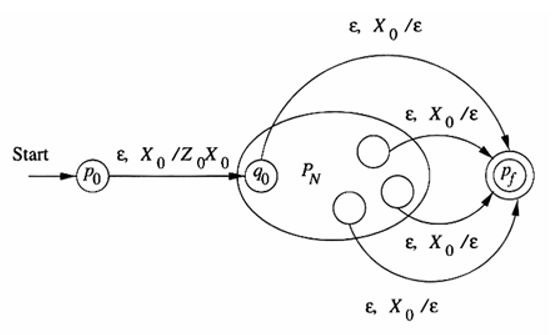
PN = (Q, Σ, Γ, δN, q0, Z0)

PF = (Q U {p0, pF}, Σ, ΓU{X0}, δF, p0, X0, {pF}), p0 startno stanje, X0 dno steka, pF prihvatajuće stanje

δ: Q X (ΣU{ε}) X Γ -> P(Q X Γ\*)

δF: (QU{p0, pF}) X (ΣU{ε}) X (ΓU{X0}) -> P((QU{p0, pF} X (ΓU{X0})\*)

1. δF(p0, ε, X0) = {(q0, z0X0)}
2. δF(q, a, Y) = δN(q, a, Y), za qϵQ, aϵΣU{ε}, YϵΓ - simulacija
3. δF(q, ε, X0) = {(pF, X0)}, za sve qϵQ

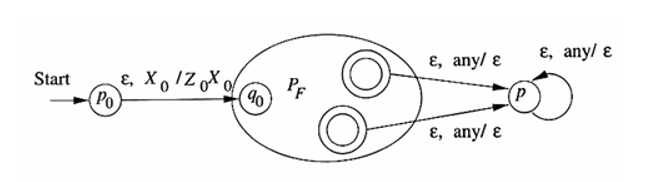


**14. Važi teorema: Ako je PF = (Q, Σ, Γ, δF, q0, Z0, F) potisni automat, i L = L(PF), tada postoji potisni automat PN takav da L = N(PN). Ako pretpostavimo da je automat PF dat, definisati (formalno, kao uredjenu šestorku) automat PN . Predstaviti PN apstraktnim dijagramom prelaza.**

PF = (Q, Σ, Γ, δN, q0, Z0, F)

PN = (Q U {p0, r}, Σ, ΓU{X0}, δN, p0, X0)

1. δN(p0, ε, X0) = {(q0, z0, X0)}
2. δN(q, a, Y) = δF(q, a, Y), za qϵQ, aϵΣU{ε}, YϵΓ - simulacija
3. δN(q, ε, \*) = {(r, ε)}



**15. Definisati deterministički potisni automat.**

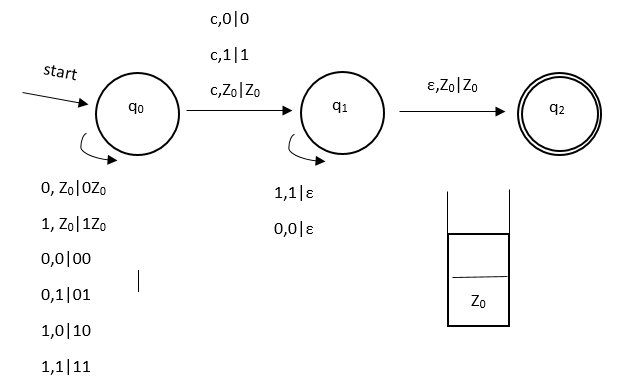
Potisni automat P=(Q, Σ, Γ, δ, q0, z0, F) je deterministički ukoliko važi:

1. δ(q, a, X) sadrži najviše 1 par za svako qϵQ, a ϵ Σ∪{ε}, XϵΓ
2. δ(q, a, X) ≠ Ø, onda δ(q, ε, X) = Ø

**16. Pokazati da sledeći jezik nije regularan: L = {wcwR | w ∈ {0, 1}\*}, gde je c fiksirani simbol koji ne pripada alfabetu {0, 1} (tzv. ”center marker”).**

Pretpostavimo suprotno, da jezik L jeste regularan. Na osnovu pumping leme, postoji konstanta p koja zadovoljava uslove leme. Posmatramo sledeću reč: w = 0pc0p, koja pripada jeziku L. Važi: |w| = 2p+1, w=xyz, |xy|≤p, y≠ ε, xykzϵL. Ceo xy upada u 0p na početku. Zaključujemo da se xy sastoji samo od nula. Dalje, y≠ ε, zaključujemo da y sadrži bar jednu nulu. Posmtramo reč xz. Ova reč će sa leve strane imati više nula nego sa desne, odnosno ne važi pumping lema. Jezik nije regularan.

**17. Konstriusati deterministički potisni automat koji prepoznaje jezik L = {wcwR | w ∈ {0, 1}\*} pomoću završnog stanja.**



**18. Kada kažemo da jezik ima prefix–svojstvo (ili prefiksno svojstvo)? Da li jezik {a mb mc k | m, k > 0} ima prefix–svojstvo? Navesti primer jednog regularnog jezika koji ima prefiksno svostvo i jednog koji ovo svojstvo nema.**

Jezik L ima prefiks svojstvo ukoliko u njemu ne postoje dve reči x i y, x≠y, tako da je x prefiks reči y. Na primer, jezik L={0n1n|n≥1} ima prefiks svojstvo.

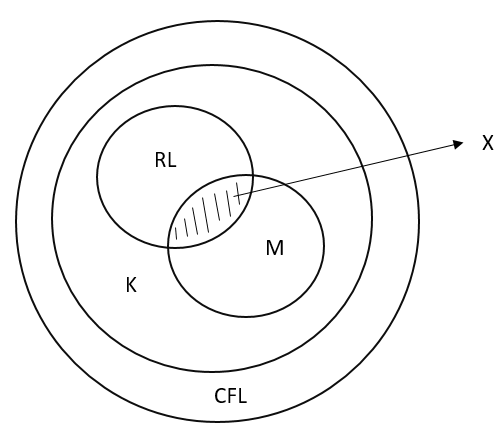
**19. Koji su dovoljni uslovi da neki jezik L bude N(P) za neki deterministički potisni automat P? Da li za jezik 0\* postoji deterministički potisni automat P takav da je 0\* = N(P)? Da li za isti jezik postoji deterministički potisni automat P takav da 0\* = L(P)? Razmotriti ista pitanja i za jezik 10\*1.**

Jezik L je N(P) za neki deterministički potisni konačni automat P ako je L=L(P’) za neki deterministički potisni automat P’ i L ima prefiks svojstvo.

Ne postoji DPDA takav da je 0\*= N(P). Pošto 0\* nema prefiks svojstvo, ne zadovoljava tražene uslove. Nasuprot tome, jezik 10\*1 ima prefiks svojstvo.

Pošto je jezik regularan, on je u klasi K, a to znači da postoji DPDA P takav da važi 0\*= L(P).

**20. U kakvom su odnosu sledeće klase jezika: klasa jezika koji su L(P) za neki deterministički potisni automat P, klasa jezika koji su N(P) za neki deterministički potisni automat P, klasa regularnih jezika i klasa kontekstno–slobodnih jezika? Predstaviti ovo skupovnim dijagramom i objasniti šta su tačno preseci i razlike ovih klasa.**

CFL – klasa kontekst slobodnih jezika

L(P) = K – klasa jezika prihvaćenih od strane DPDA završnim stanjem

RL – klasa regularnih jezika

N(P) = M – klasa jezika prihvaćenih od strane DPDA praznim stekom

X – regularni jezici koji imaju prefiks svojstvo

RL/X – regularni jezici bez prefiks svojstva

**21. Za zadatak 20 navesti primere jezika koji se nalaze u razlikama navedenih klasa jezika. tj. jezike koji pokazuju da se odgovarajuće klase ne poklapaju. (Razlika klasa ima isti smisao kao razlika skupova.)**

{0}\* ϵ RL/X, jeste regularan ali nema prefiks svojstvo

{0\*1} ϵ X, regularan jezik sa prefiks svojstvom

{wcwR} ϵ K– kontekst slobodan jezik za koji postoji DPDA takav da prihvata ovaj jezik završnim stanjem

{wwR} ϵ CFL– kontekst slobodan jezik, ali ne postoji DPDA

22. Pretpostavimo da je L jezik takav da važi L = L(P) za neki deterministički potisni automat P, i L nema prefiksno svojstvo. Pokazati da ne postoji deterministički potisni automat P’ takav da je L = N(P’).

**23. Neka je dat jezik palindroma parne dužine nad alfabetom {0, 1}, L = {wwR | w ∈ {0, 1} ∗}.**

a) Dokazati da je jezik kontekst–slobodan.

**b) Dokazati da ne postoji deterministički potisni automat P takav da važi L = L(P).**

a)

b) Ne postoji deterministički potisni automat takav da odgovara jeziku L = {wwR | w ∈ {0, 1}\*}. Dokaz: pretpostavimo suprotno, da postoji ovakav automat P. Posmatramo reč w = 0n110nϵL. Automat P prihvata ovu reč. Automat P bi morao prvu sekvencu nula da gurne na stek, takođe i simbol 1, a onda da uradi uparivanje. Na steku ostaje samo Z0, tj reč se prihvata. Sada posmatramo reči:

1. w1 = 0n110n|0n110n ϵ L
2. w2 = 0n110n|0m110m – ne pripada L (n≠m)

Automat P bi morao da prihvati prvu reč, a odbaci drugu. Budući da je reč w prefiks obe ove reči i da je automat deterministički, pri obradi reči w1 i w2 automat se ponaša isto kao pri čitanju reči w, To znači da će u trenutku čitanja nakon uspravne crte na steku biti samo Z0. Automat nije zapamtio koliko je bilo nula u prvom delu reči, pa prema tome ne može da napravi razliku prilikom procesiranja reči w1 i w2 do kraja. Sa druge strane, jednu od ove dve reči treba prihvatiti, a drugu odbaciti, što je nemoguće. Zaključujemo da ovakav automat ne postoji.

**24. Pumping lemma za kontekst–slobodne jezike. Iskaz leme. Objasniti za šta se i kako primenjuje ova lema.**

Neka je L kontekst slobodan jezik. Onda postoji konstanta n tako da važi da svaka reč z iz L, |z| >= n može da se predstavi u obliku q = uvwxy i mora da postuje sledeće uslove:

1. |vx| != ε
2. |vwx| <= p
3. uvrwxry ∈ L, za svako r >= 0

Primena leme je da se pokaže da dati jezik nije kontekst-slobodan.

Postupak: Pretpostavimo da posmatrani jezik L jeste kontekst slobodan. Prema pumping lemi to znači da onda postoji const p > 0 za koju važi da se svaka reč iz L |z| >= p može napisti u obliku z=uvwxy, gde je |vwx| <= p i vx ≠ ε. Uzmemo jednu reč z iz jezika i tražimo neko i ≥ 0, takvo da ne važi uviwxiy ϵ L. Ako pronađemo ovkvo i, zanči da imamo kontradikciju, dakle jezik L nije kontekst slobodan.

**25.** **(a) Pokazati da jezik L = {0n1n2n | n ≥ 1} nije kontekst–slobodan.**

Pretpostavimo da jezik L jeste kontekst-slobodan. Tada postoji const p iz pumping leme za CFL, p>0. Biramo reč 0p1p2p ϵ L. Na osnovu pumping leme, w se može zapisati kao w=uvwxy, pri čemu |vwx| <= p i vx ≠ ε i uviwxiy ϵ L za svako i. Pošto je |vwx| <= p zaključujemo da može da obuhvati najviše 2 različita simbola:

1. Pretpostavimo da uvw obuhvata samo jedan simbol npr 1: uvw se sastoji samo od jedinica i važi vx ≠ ε. To znači da bar jedna od reči v,x ima bar jednu jedinicu. Onda pumpanjem, reč uwy ima manji broj jedinica u odnosu na 0 i 2, pa ne pripada jeziku.
2. Pretpostavimo da uvw obuhvata dva simbola npr 1 i 2: uvw se sastoji od simbola 1 i 2 i važi vx ≠ ε. To znači da bar jedna od reči v,x ima bar jednu jedinicu ili dvojku. Onda pumpanjem, reč uwy ima veći broj nula u odnosu na 1 i 2, pa ne pripada jeziku.

Dakle, ne važi pumping lema za CFL, pa jezik nije kontekst-slobodan.

**(b) Pokazati da jezik L = {0j1j2i3j |i ≥ 1, j ≥ 1} nije kontekst–slobodan.**

Pretpostavimo da jezik L jeste kontekst-slobodan. Tada postoji const p iz pumping leme za CFL, p>0. Biramo reč 0l1m2p3l ϵ L, l=m=p. Na osnovu pumping leme, w se može zapisati kao w=uvwxy, pri čemu |vwx| <= p i vx ≠ ε i uviwxiy ϵ L za svako i. Pošto je |vwx| <= p zaključujemo da može da obuhvati najviše 2 različita simbola:

1. Pretpostavimo da uvw obuhvata samo jedan simbol npr 0: uvw se sastoji samo od jedinica i važi vx ≠ ε. To znači da bar jedna od reči v,x ima bar jednu nulu. Onda pumpanjem, reč uwy ima manji broj nula u odnosu na 1 i 3, pa ne pripada jeziku.
2. Pretpostavimo da uvw obuhvata dva simbola npr 0 i 1: uvw se sastoji od simbola 1 i 2 i važi vx ≠ ε. To znači da bar jedna od reči v,x ima bar jednu jedinicu ili dvojku. Onda pumpanjem, reč uwy ima različit broj trojki u odnosu na 0 i 1, pa ne pripida jeziku.

Dakle, ne važi pumping lema za CFL, pa jezik nije kontekst-slobodan.

**26. Definisati pojmove odlučivih i neodlučivih problema (decidable/undecidable). Kakav je odnos ove dve klase problema u odnosu na podelu problema na rešive i nerešive u polinomijalnom vremenu (tj. na tractable/intractable)?**

Odlučivi problemi su problemi koje je moguće rešiti računskim mašinama tj za njih postoji algoritam. Suprotno, neodlučivi problemi su problemi koje nije moguće rešiti računskim mašinama, tj za njih ne postoji algoritam. Tjuringova mašina pravi podelu problema na odlučive i neodlučive u zavisnosti od toga da li za njih postoji nedeterministička Tjuringova mašina ili ne. Dalje, odlučivi problemi se mogu podeliti na tractable i intractable probleme. Tractable problemi su rešivi u polinomijalnom vremenu tj za njih postoje efikasni algoritmi, a intractable problemi nisu rešivi u polinomijalnom vremenu tj za njih postoje algoritmi ali oni nisu efikasni.

**27. Navesti primer neodlučivog problema i objasniti zašto je neodlučiv.**

Posmatramo jezik L koji sadrži beskonačno mnogo reči nad bilo kojim alfabetom, pri čemu ne postoji nikakvo pravilo kako je definisan L. Data je reč w nad alfabetom, da li pripada jeziku L? Ne postoji računsko rešenje za ovaj problem, odnosno, ne postoji Tjuringova mašina => problem nije odlučiv.

**28. Problem sa stanovišta teorije algoritama definišemo kao pitanje utvrdjivanja pripadnosti proizvoljnog stringa datom jeziku L. Pokazati da:**

**(a) ima neprebrojivo mnogo jezika (continuum mnogo) nad datim konačnim alfabetom Σ.**

Svaki jezik nad alfabetom Σ je podskup od Σ\*, dakle pitanje je koliki je broj podskupova od Σ\*? |P(Σ)|= |P(N)| = 2Xo = c = |R| - neprebrojivo mnogo

**(b) ima prebrojivo mnogo programa (bilo kojih, pa specijalno i onih koji rešavaju problem pripadnosti striga datom jeziku).**

Posmatramo neki programski jezik, npr C. Program u programskom jeziku C je konačan niz karaktera. Broj karaktera je takođe konačan. Ako posmatramo program dužine n, nϵN, fiksiran, postoji konačno mnogo ovih programa. Prebrojiva unija konačnih skupova = prebrojiv skup. Dakle, svih programa u C-u ima prebrojivo mnogo.

**29. Tjuringova mašina, definicija. Kakav je značaj ovog formalizma u odnosu na podelu problema na decidable/undecidable?**

Tjuringova mašina je uređena sedmorka T=(Q,Σ,Γ,δ,q0,B,F), gde je:

1. Q konačan skup stanja
2. Σ konačan ulazni alfabet
3. Γ konačan alfabet trake, Σ ⊆Γ
4. δ funkcija prelaza, QXΓ -> QXΓX{L,R}, (q,x) Ⴡ> (p,y,L)
5. q0 startno stanje
6. B simbol praznog polja trake (blank), BϵΓ/Σ
7. F skup završnih stanja

Tjuringova mašina pravi podelu problema na odlučive i neodlučive u zavisnosti od toga da li za njih postoji nedeterministička Tjuringova mašina ili ne (odlučivi imaju Tjuringovu mašinu, a neodlučivi nemaju).

**30. Tjuringova mašina, definicija. Funkcija prelaza.**

Tjuringova mašina je uređena sedmorka T=(Q,Σ,Γ,δ,q0,B,F), gde je:

1. Q konačan skup stanja
2. Σ konačan ulazni alfabet
3. Γ konačan alfabet trake, Σ ⊆Γ
4. δ funkcija prelaza, QXΓ -> QXΓX{L,R}, δ(q,x) = (p,y,L)
5. q0 startno stanje
6. B simbol praznog polja trake (blank), BϵΓ/Σ
7. F skup završnih stanja

**31. Trenutna konfiguracija (instantaneous description, ID) Tjuringove mašine. Pokret Tjuringove mašine, odnosno relacija Ⴡ. Zatvorenje ove relacije.**

Trenutna konfiguracija: npr x1x2...xi-1qxixi+1...xn

1. x1x2...xn – skup ulaznih simbola
2. levo i desno od x1 tj xn su B (blank)
3. stanje u kojem se mašina nalazi piše se kao deo stringa trenutne konfiguracije
4. pozicija gde se upisuje stanje je ispred glave za čitanje

Pokret Tjuringove mašine je promena konfiguracije. Oznaka je ჁT. Pokret je definisan funkcijom prelaza na sledeći način: Pretpostavimo da imamo trenutnu konfiguraciju x1x2...xi-1qxixi+1...xn. Pretpostavimo da je definisana funkcija prelaza δ(q,xi) = (p,y,R). Tada: x1x2...xi-1qxixi+1...xn Ⴡ x1x2...xi-1ypxi+1...xn predstavlja pokret Tjuringove mašine.

Zatvorenje relacije pokreta (Ⴡ\*) – ako su K1 i K2 konfiguracije Tjuringove mašine, tada K1 Ⴡ\*T K2 označava da mašina T od konfiguracije K1 dolazi do konfiguracije K2 u konačno mnogo pokreta (0 ili više).

**32. Specijalni slučajevi pokreta Tjuringove mašine: glava za čitanje se nalazi na prvom non– blank simbolu sleva ili poslednjem non–blank simbolu. (Razmotriti pokrete u oba smera za oba slučaja.)**

1. i = 1 (glava za čitanje se nalazi na poziciji 1)
2. δ(q,x1) = (p,y,R), qx1x2...xn Ⴡ ypx2x3...xn
3. δ(q,x1) = (p,y,L), qx1x2...xn Ⴡ pByx2x3...xn
4. δ(q,x1) = (p,B,L), qx1x2...xn xn Ⴡ pBBx2x3...xn
5. δ(q,x1) = (p,B,R), qx1x2...xn Ⴡ Bpx2x3...xn
6. i = n (glava za čitanje se nalazi na poslednjoj poziciji)
7. δ(q,xn) = (p,y,L), x1x2...xn-1qxn Ⴡ x1x2...xn-2pxn-1y
8. δ(q,xn) = (p,y,R), x1x2...xn-1qxn Ⴡ x1x2...xn-1ypB
9. δ(q,xn) = (p,B,L), x1x2...xn-1qxn Ⴡ x1x2...xn-2pxn-1B
10. δ(q,xn) = (p,B,R), x1x2...xn-1qxn Ⴡ x1x2...xn-1BpB

**33. Dijagram prelaza Tjuringove mašine. Opis.**

Dijagram prelaza Tjuringove mašine sastoji se od skupa čvorova koji odgovaraju stanjima Tjuringove mašine lukova koji označavaju prelaze. Luk od stanja q do stanja p je obeležen s jednim ili više prelza tipa: X|Y,D

* X je šta upravo čitamo sa trake
* Y je šta stavljamo na pročitano mesto
* D znači u koju stranu se pomeramo nakon čitanja i upisivanja (L-levo ili R-desno).