Zadaci za vježbu 3. ciklus – tutorial by Diablo

Skidanjem ovog dokumenta pristali ste donirati autoru bubreg, jetru ili neki drugi organ u slučaju nužde ☺

27. Konstruirati Turingov stroj u osnovnom obliku koji oduzima dva binarna broja zapisana na traci. Najznačajnija znamenka je lijevo, a brojevi su odvojeni znakom –. Drugi broj se oduzima od prvog pri čemu prvi broj sigurno nije manji od drugog. Glava se nalazi na početku ulaznog niza, a s obje strane ulaznog niza nalaze se praznine.

Da bi uopće mogli početi rješavati zadatak, moramo dobro shvatiti kako izgleda početno stanje trake i što se točno od nas traži. Naše početno stanje trake izgleda recimo ovako :

B	1	0	0	1	1	0	-	1	0	1	0	B

Legenda : promjene na traci su označene crvenom, a novi prijelazi plavom bojom.
Dakle, imamo dva broja zapisana u binarnom obliku odvojena znakom -, a prije i poslije se nalaze praznine označene oznakom B.
Kako uopće funkcionira oduzimanje brojeva? Pa oduzimaju se s desna ulijevo, i to drugi broj od prvog broja. Prvi zadatak nam je u početnom stanju (glava se nalazi na početku ulaznog niza i stanje glave je q0) se prebaciti na krajnji desni znak ulaznog niza kako bi uopće mogli početi oduzimati.

B	1	0	0	1	1	0	-	1	0	1	0	B

Znači početno stanje je : (q0, 1) te nam trebaju sljedeći prijelazi :
(q0, 1)=(q0, 1, R), (q0, 0)=(q0, 0, R), (q0, -)=(q0, -, R) i (q0, B)=(q1, B, L).
Ovim zadnjim prijelazom smo, kad smo došli do desne praznine (desnog graničnika), znali da smo došli do kraja, prešli u sljedeće stanje (q1) i pomaknuli se ulijevo na krajnje desni znak ulaznog niza. Sad možemo početi s oduzimanjem ☺
Za početak, kad pročitamo ulazni znak moramo ga nekako zapamtiti, to bila 0 ili 1 da bi znali kako oduzimati. To ćemo ostvariti s dva nova stanja : q2 ako je bila 0, i q3 ako je bilo 1.
Kad zapamtimo ulazni znak prebrišemo ga sa prazninom, te moramo prijeći preko svih ostalih znamenaka drugog broja do krajnje desne prvog broja i onda ih oduzeti. Kad dođemo do drugog broja to ćemo znati po graničniku '-'.

B	1	0	0	1	1	0	-	1	0	1	B	B

Za to nam trebaju sljedeći prijelazi :
(q1, 0)=(q2, B, L), (q2, 1)=(q2, 1, L), (q2, 0)=(q2, 0, L) i (q2, -)=(q4, -, L).
Ovo stanje q4 nam služi kako bi i nakon graničnika '-' znali da smo zapamtili 0, a stanje q5 će nam služiti da smo zapamtili znak 1. Pošto je zapamćen znak 0 i opet je ulazni znak 0, rezultat oduzimanje je 0-0=0. E sada moramo ovaj rezultat označiti nekako drugačije kako ga ne bi miješali sa ostalim znamenkama. Ako će rezultat bio 0 označit ćemo ga znakom N, a ako je bio 1 onda znakom J. Sada će se ovo vrtiti u krug pa prelazimo opet u početno stanje q0 koje će nas dovesti do krajnje desnog znaka drugog broja (ulaznog niza) kako bi mogli započeti novu iteraciju oduzimanja.

B	1	0	0	1	1	N	-	1	0	1	B	B

Za ovo nam trebaju sljedeći prijelazi :
(q4, 0)=(q0, N, R), (q0, -)=(q0, -, R), (q0, 1)=(q0, 1, R), (q0, 0)=(q0, 0, R), i
(q0, B)=(q1, B, L).

E sad imamo novu situaciju. Ulazni znak nam je 1 (koju prebrišemo s prazninom) i prelazimo u stanje q3 koje nam služi za pamćenje jedinice. Prelazimo opet preko svih znamenaka drugog broja do graničnika '-' gdje prelazimo u stanje q5 koje služi za pamćenje jedinice nakon graničnika. Naravno ne smijemo se zaboraviti pomaknuti i preko slova jer su ona dio rješenja, sve do prve desne znamenke prvog broja.

B	1	0	0	1	1	N	-	1	0	B	B	B

Za to trebamo sljedeće prijelaze :
(q1, 1)=(q3, B, L), (q3, 0)=(q3, 0, L), (q3, 1)=(q3, 1, L), (q3, -)=(q5, -, L) i (q5, N)=(q5, N, L)

Sada imamo zapamćen znak 1, ulazni znak je 1, pa je rezultat oduzimanja 1-1=0.
Zapisujemo rezultat kao znak N i opet idemo na početak.

B	1	0	0	1	N	N	-	1	0	B	B	B

Prijelazi :
(q5, 1)=(q0, N, R), (q0, N)=(q0, N, R), (q0, -)=(q0, -, R), (q0, 1)=(q0, 1, R),
(q0, 0)=(q0, 0, R), i (q0, B)=(q1, B, L).

Ok već ste shvatili princip pretpostavljam ☺

B	1	0	0	1	N	N	-	1	B	B	B	B

Prijelazi :
(q1, 0)=(q2, B, L), (q2, 1)=(q2, 1, L), (q2, -)=(q4, -, L), (q4, N)=(q4, N, L).

I evo nove situacije. Sad imamo zapamćen znak 0, a ulazni znak nam je 1. Rezultati oduzimanja je 1-0=1. Ovaj put ćemo morati zapisati kao rješenje znak J.

B	1	0	0	J	N	N	-	1	B	B	B	B

Prijelazi :
(q4, 1)=(q0, J, R), (q0, N)=(q0, N, R), (q0, -)=(q0, -, R), (q0, 1)=(q0, 1, R) i
(q0, B)=(q1, B, L).

B	1	0	0	J	N	N	-	B	B	B	B	B

Prijelazi :
(q1, 1)=(q3, B, L), (q3, -)=(q5, -, L), (q5, N)=(q5, N, L) i (q5, J)=(q5, J, L).

I evo posljednjeg četvrtog mogućeg slučaja koji je ujedno i najkompliciraniji. Imamo zapamćen znak 1, a ulazni znak nam je 0, rezultat oduzimanja 0-1=1, ali zbog toga samo izvršili posudbu s više znamenke, što znači da naše oduzimanje tu ne staje, već ide do sljedeće lijeve znamenke. Ako je i sljedeća znamenka 0 opet imamo oduzimanje 0-1=1 pa idemo dalje. Tek kad dođemo do znaka 1 onda ćemo imati oduzimanje 1-1=0 te smo napokon završili s oduzimanjem u ovom krugu. Ukratko za ovaj slučaj, ako je zapamćen znak 1, sve ulazne znakove 0 pretvaramo u znakove 1, a tek dok dođemo do znaka 1 pretvorimo ga u 0 i završavamo s oduzimanjem. Posudbu će biti uvijek moguće izvršiti jer je uvjet zadatka da prvi broj nije sigurno manji od drugog što i lijepo piše u tekstu zadatka ☺

B	1	0	J	J	N	N	-	B	B	B	B	B

Prijelazi :
(q5, 0)=(q6, J, L).

Tu sad imamo novo stanje q6 koje će nam napraviti to specijalno oduzimanje do kraja. Kad je oduzimanje gotovo, pogađate, idemo u q0 i natrag na novu iteraciju.

B	0	1	J	J	N	N	-	B	B	B	B	B

Prijelazi :
(q6, 0)=(q6, 1, L), (q6, 1)=(q0, 0, R), (q0, 1)=(q0, 1, R), (q0, J)=(q0, J, R), (q0, N)=(q0, N, R), (q0, -)=(q0, -, R) i (q0, B)=(q1, B, L).

Sad imamo u stanju q1 ulazni znak '-'. To nam govori da smo cijeli drugi broj oduzeli od prvog broja. Iako ne piše u zadatku, podrazumijeva se da rezultat oduzimanja zapišemo na traci kako spada, odnosno maknemo znak '-', sva slova pretvorimo natrag u znamenke, i to bez beznačajnih nula s krajnje lijeve strane.

B	0	1	1	1	0	0	B	B	B	B	B	B

Prijelazi :
(q1, -)=(q7, B, L), (q7, N)=(q7, 0, L), (q7, J)=(q7, 1, L), (q7, 1)=(q7, 1, L), (q7, 0)=(q7, 0, L) i (q7, B)=(q8, B, R).

Stanje q8 nam je potrebno da još maknemo beznačajne nule s krajnje lijeve strane. Nakon što ih maknemo i dođemo do prve jedinice prelazimo u prihvatljivo stanje q9, Turingov stroj završava s radom jer nema više definiranih prijelaza, a na traci ostaje konačno rješenje.

B	B	1	1	1	0	0	B	B	B	B	B	B

Prijelazi :
(q8, 0)=(q8, B, R), (q8, 1)=(q9, 1, L).
THE END

Konačnu definiciju automata i tablicu prijelaza imate u zadacima za vježbu.

NAPOMENA : ovo je konstrukcija Turingovog stroja na nekom primjeru. Međutim to je jako sklisko jer je u primjeru jako teško obuhvatiti sve moguće slučajeve pa dobro razmišljajte kod popunjavanja tablice prijelaza što se sve može desiti. U ovom konkretnom primjeru fali slučaj kad su oba broja ista, onda bi rješenje bile same nule, i u stanju q8 bi maknuli sve te beznačajne nule i došli do praznine B, pa fali prijelaz (q8, B)=(q9, 0, L), kojim bi otišli u prihvatljivo stanje q9 i zapisali na traku rješenje 0. Također u primjeru fali i prijelaz (q4, J)=(q4, J, L).

28. Konstruirati Turingov stroj koji redom generira sve potencije broja 2. Vrijednost jednog broja na traci je zapisana odgovarajućim brojem jedinica. Na ulaznoj traci Turingovog stroja na početku je zapisan niz $1. Brojevi su međusobno odvojeni graničnikom $. S obje strane ulaznog niza nalaze se praznine.

Znači od nas se traži da na traci zapišemo potencije broja dva prikazane brojem jedinica i odvojene graničnikom $, gdje nam početno stanje trake izgleda ovako :

$	1	B	B	B	B	B	B	B	B	B	B	B

I ovaj zadatak se da ostvariti pomoću osnovnog modela Turingovog stroja, ali rješenje bi bilo dosta komplicirano, s puno stanja i u 'onoj muškoj stvari', pa ćemo se poslužiti nekim drugim modelom. Za ovaj slučaj korist ćemo Turingov stroj s jednom trakom i pomoćnim tragom koji je istovjetan osnovnom modelu. Vidjet ćete da to nije ništa komplicirano, a uvelike nam pojednostavljuje problem ☺

Dakle novo početno stanje će nam izgledati ovako :

$	1	B	B	B	B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	B	B

Potencije broja 2 jednake su množenju prošle potencije s 2, odnosno u ovom slučaju udvostručenju broja jedinica prošle potencije čije je rješenje prikazano nizom jedinica. Početno stanje q0 će nam služiti da postavimo zastavicu koju smo jedinicu već iskoristili, tj koju jedinicu udvostručujemo. Na traci već imamo zapisan broj 2^0=1, te nam je prvi korak označiti prvu jedinicu posebnom zastavicom. U ovom modelu čita se uređeni par, a ne samo jedan znak, tako da će nam početno stanje biti (q0,[$,B]).

$	1	B	B	B	B	B	B	B	B	B	B	B
B	*	B	B	B	B	B	B	B	B	B	B	B

Prijelazi :
(q0,[$,B])=(q0,[$,B],R), (q0,[1,B])=(q1,[1,*],R).

Sad kad smo označili jedinicu koju treba udvostručiti, slijedi korak gdje provjeravamo da li se piše nova potencija broja, pa se treba postaviti graničnik '$' koji odvaja te dvije različite potencije. Upravo tome će nam služiti stanje q1, za postavljanje graničnika '$'.
Pošto nam je ulazni znak praznina [B,B], moramo postaviti novi graničnik jer zapisujemo novu potenciju broja 2.

$	1	$	B	B	B	B	B	B	B	B	B	B
B	*	B	B	B	B	B	B	B	B	B	B	B

Prijelazi :
(q1,[B,B])=(q2,[$,B],R).

Sada smo prešli u novo stanje q2 kojim ćemo udvostručiti jedinice. Idemo do prve praznine u koju ćemo zapisati novu znamenku 1.

$	1	$	1	B	B	B	B	B	B	B	B	B
B	*	B	B	B	B	B	B	B	B	B	B	B

Prijelazi :
(q2,[B,B])=(q3,[1,B],R).

Sad smo prešli u novo stanje q3 u kojem ćemo zapisati još jednu jedinicu, i time smo udvostručili prije označenu jedinicu.

$	1	$	1	1	B	B	B	B	B	B	B	B
B	*	B	B	B	B	B	B	B	B	B	B	B

Prijelazi :
(q3,[B,B])=(q4,[1,B],L), (q4,[1,B])=(q4,[1,B],L), (q4,[$,B])=(q4,[$,B],L).

Stanje q4 će nam služiti za šetanje ulijevo do prve označene jedinice kako bi ponavljali ovaj postupak. Kad dođemo do prve označene jedinice prelazimo ponovo u početno stanje q0, u kojem ponovo tražimo prvu neoznačenu jedinicu.

$	1	$	1	1	B	B	B	B	B	B	B	B
B	*	B	B	B	B	B	B	B	B	B	B	B

Prijelazi :
(q4,[1,*])=(q0,[1,*],R), (q0,[$,B])=(q0,[$,B],R).

I sad se postupak ponovo ponavlja. Označi se jedinica, prijeđe u stanje q1 gdje se postavlja novi graničnik, ide u q2 gdje se zapisuje prva jedinica, zatim u q3 gdje se zapisuje druga jedinica, te napokon u q4 gdje se opet vraćamo do prve označene jedinice sa zastavicom.

$	1	$	1	1	$	1	1	B	B	B	B	B
B	*	B	*	B	B	B	B	B	B	B	B	B

Prijelazi :
(q0,[1,B])=(q1,[1,*],R), (q1,[1.B])=(q1,[1,B],R), (q1,[B.B])=(q2,[$,B],R), (q2,[B,B])=(q3,[1,B],R), (q3,[B,B])=(q4,[1,B],L), (q4,[1,B])=(q4,[1,B],L), (q4,[$,B])=(q4,[$,B],L).

Sad kad u q0 označimo sljedeću jedinicu, q1 neće postavljati novi graničnik jer nismo udvostručili sve jedinice ove potencije, a to zna po tome što neće doći do praznine nego do već postavljenog graničnika, gdje će prijeći u q2 koji će tražit sljedeće praznine da zapiše nove jedinice.

$	1	$	1	1	$	1	1	1	B	B	B	B
B	*	B	*	*	B	B	B	B	B	B	B	B

Prijelazi :
(q4,[1,*])=(q0,[1,*],R), (q0,[1,B])=(q1,[1,*],R), (q1,[$,B])=(q2,[$,B],R), (q2,[1,B])=(q2,[1,B],R), (q2,[B,B])=(q3,[1,B],R).

I sad tako u nedogled ...

$	1	$	1	1	$	1	1	1	1	B	B	B
B	*	B	*	*	B	B	B	B	B	B	B	B

Prijelazi :
(q3,[B,B])=(q4,[1,B],L), (q4,[1,B])=(q4,[1,B],L), (q4,[$,B])=(q4,[$,B],L).

Konačnu definiciju automata i tablicu prijelaza imate u zadacima za vježbu.

NAPOMENA : u toj tablici ima par grešaka, stanje q5 je jednako stanju q0, a q6=q1.

29. Konstruirati Turingov stroj koji prihvaća nizove iz jezika L={w € (a+b+c)* | $n_a=n_b=n_c$}. Nakon što Turingov stroj završi s radom stanje na traci mora biti isto kao početno. S obje strane ulaznog niza nalaze se praznine.

Od nas se traži konstruirati Turingov stroj koji će provjeriti ispravnost bilo kojeg niza znakova a, b i c, gdje je jedini uvjet da broj tih triju znakova mora biti jednak.

Jedan od mogućih nizova znakova a, b i c može biti sljedeći : **PabbccaP** , gdje je znak P oznaka za prazninu ispred i iza ulaznog niza kao što piše u zadatku.

Mi se u početku nalazimo u početnom stanju q0, a glava je pozicionirana iznad prvog znaka ulaznog niza, znaka **a**.
Međutim to je samo za ovaj slučaj, na prvom mjestu je mogao također biti ili znak b ili c.
Da nam bude lakše, za bilo koji početni pročitani znak, q0 prelazi u novo stanje s indexom toga znaka. Tako će za ovaj slučaj i pročitani znak **a** preći u novo stanje qa, a pročitani ulazni znak ćemo označiti velikim slovima da ih ne miješamo s nepročitanima. Tako za početno stanje imamo ove moguće prijelaze :
(q0,a)=(qa,A,R), (q0,b)=(qb,B,R), (q0,c)=(qc,C,R), te pošto može biti i prazni niz (Kleenov operator *) imamo prijelaz (q0,P)=(qpc,P,L).
Stanje qpc je stanje koje nije prihvatljivo ali označuje da je niz prihvatljiv i kad završimo s ispitivanjem niza vraćamo sva velika slova u mala slova, jer je u zadatku navedeno da na kraju niz mora izgledati kao početni. Tu su sad još navedeni i prijelazi (q0,A)=(q0,A,R), (q0,B)=(q0,B,R), (q0,C)=(q0,C,R), koji su potrebni za šetnju preko velikih slova.

U našem primjeru sad bi stanje trake bilo sljedeće : **PAbbccaP** , stanje glave **qa** i pozicija iznad prvog znaka **b**.

Da bi izbrojali da li ima isti broj znakova, mi ćemo za svaki pročitani znak **a** tražiti po jedan **b** i jedan **c** (i obrnuto za svaki slučaj prvog znaka) te tako se vrtjeti u krug do kraja. Za naše stanje qa tražit ćemo bilo koji znak od znakova b ili c, te prijeći u novo stanje čiji je indeks sastavljen od kombinacija ta dva pročitana slova, a preko ostalih znakova ćemo samo prijeći.
Tako ćemo imati sljedeće prijelaze :
(qa,b)=(qab,B,R), (qa,c)=(qac,C,R), (qa,a)=(qa,a,R), (qa,A)=(qa,A,R), (qa,B)=(qa,B,R), (qa,C)=(qa,C,R) i na kraju kad više nema znakova **b** ili **c** onda dođe do prazne ćelije P gdje imamo prijelaz (qa,P)=(qoc,P,L) u kojem prelazimo u stanje qoc koje nam govori da je niz neprihvatljiv i u kojem vraćamo sva velika slova u mala slova kako bi niz izgledao kao početni.
Tako na isti način izgledaju prijelazi i za stanja qb i qc.
Evo već smo riješili prva četiri retka tablice ☺

Sada bi naš niz izgledao ovako : **PABbccaP** , stanje glave **qab** i pozicija iznad znaka **b**.

Sad sva stanja s dva indexa (kao naše qab) prelaze preko svih slova koja nisu ono treće slovo koje se traži (u našem slučaju c). Kad ga nađu znači da su prepisani po jedan a, b i c sa velikim slovima, te prelazimo u novo stanje qv koje će nas vratiti na početak niza do znaka P, prijeći u početno stanje q0, i krenuti sve ispočetka, odnosno tražit će se prvo novo malo slovo i ponavljati postupak.
Tako recimo za naše stanje qab imat ćemo sljedeće prijelaze :
(qab,a)=(qab,a,R), (qab,b)=(qab,b,R), (qab,c)=(qv,C,L), (qab,A)=(qab,A,R), (qab,B)=(qab,B,R), (qab,C)=(qab,C,R), i naravno ako ne nađe traženi treći znak i dođe do prazne ćelije P opet ide u neprihvatljivo stanje qoc prijelazom (qab,P)=(qoc,P,L).
Na isti način glase prijelazi za ostala stanja s dva indeksa.

Sada bi naš niz izgledao ovako : **PABbCcaP** , stanje glave **qv** i pozicija iznad znaka **b**.
Stanje qv ide do početka niza, te za pročitanu praznu ćeliju P ide u stanje q0 gdje se sve ponavlja.
Nakon drugog kruga niz izgleda ovako : **PABBCCAP** , stanje glave qv.
Pošto su sva slova velika, q0 će proći preko cijelog niza sve do prazne ćelije P. Tako će znati da je niz prihvatljiv i prijeći u stanje qpc. To stanje vraća sva velika slova u mala prijelazima :

(qpc,A)=(qpc,a,L), (qpc,B)=(qpc,b,L), (qpc,C)=(qpc,c,L), i kad dođe do lijeve prazne ćelije prelazi u prihvatljivo stanje prijelazom (qpc,P)=(qf,P,R), gdje Turingov stroj staje i prihvaća niz.
Da niz nije bio dobro zadan, nekim prijelazom bi se prešlo u stanje qoc koje označuje da je niz neprihvatljiv, i u tom stanju bi vratili također sva velika slova u mala slova prijelazima :
(qoc,A)=(qoc,a,L), (qoc,B)=(qoc,b,L), (qoc,C)=(qoc,c,L), (qoc,a)=(qoc,a,L), (qoc,b)=(qoc,b,L), (qoc,c)=(qoc,c,L).
Dok dođe do prazne ćelije P stroj nema definiran prijelaz te staje s radom, a pošto je qoc neprihvatljivo stanje, stroj ne prihvaća niz.

Konačnu definiciju automata i tablicu prijelaza imate u zadacima za vježbu.

NAPOMENA : u toj tablici ima par tiskarskih grešaka, kao npr (qc,B)=(qc,A,R) i (qc,C)=(qc,A,R).
Također detaljnijim proučavanjem mislim da ima viška prijelaza, kao npr prijelazi : (qa,A)=(qa,A,R), (qb,B)=(qb,B,R) i (qc,C)=(qc,C,R), jer mi nije jasno kako ako se uvijek traži prvo malo slovo, može naletjet na već iskorišteno malo slovo istog tipa. A mislim da ih ima i još, samo mi se ne da više proučavat ☺
Na sreću pa nitko ne traži minimalnu tablicu prijelaza, pa se ne zabrinjavajte oko toga.

30. Konstruirati gramatiku koja generira nizove iz jezika
L={w € (a+b+c)* | $n_a \neq n_b$, $n_a \neq n_c$, $n_b \neq n_c$}.

Sada trebamo konstruirati gramatiku koja generira bilo koje nizove znakova a, b i c gdje je broj pojedinih znakova različit od broja ostalih znakova.

Eh ovo će biti dosta teško za objasnit, ali pokušat ćemo ☺
Za početak, ne možemo generirati prazni niz jer će onda broj svih znakova biti jednak i to jednak 0. Pa onda prvo stavimo početnu produkciju sa svim nezavršnim znakovima kojima možemo generirati znakove a, b, c neograničen broj puta :

S -> ABCS

No u nizu se ne mora pojaviti jedan znak uopće (ne dva jer bi onda njihov broj bio jednak), i osiguramo to produkcijama :

S -> ABT
S -> ACU
S -> BCV

Tim produkcijama smo automatski osigurali manji broj pojavljivanja jednog znaka, jer da bi se produkcija S prestala izvoditi mora se izvesti jedna od njih triju čime je osigurano manje pojavljivanje jednog znaka. Kojeg, ovisno o produkciji.

Time smo pokrili broj pojavljivanja u početnom slučaju. Nezavršni znakovi T, U i V imaju dvije funkcije, prva je neograničeno pojavljivanje pripadnih nizova produkcijama :

T -> ABT
U -> ACU
V -> BCV

Druga je osiguravanje manji broj pojavljivanja drugog znaka od trećeg produkcijama :

T -> AX	U -> AX	V -> BY
T -> BY	U -> CZ	V -> CZ

Opet, da bi se produkcije T, U ili V prestale izvoditi, mora se izvesti jedna od dviju gornjih pripadnih produkcija, a time je osigurano da će se i drugi znak pojavit različit puta od preostalog zadnjeg.

Sad produkcije X, Y i Z imaju slične dvije funkcije. Prva je neograničeno pojavljivanje pripadnih produkcija, a druga epsilon prijelaz kad ne želimo više generirati te znakove :

X -> AX	X -> ε
Y -> BY	Y -> ε
Z -> CZ	Z -> ε

Naravno nemojmo zaboraviti iz nezavršnih znakova A, B i C generirati pripadne završne :

A -> a
B -> b
C -> c

No još nije gotovo. Mi smo sada samo osigurali različit broj pojavljivanja znakova a, b i c. No raspored i kombinacije znakova mogu biti bilo kakve. Npr mi sad ne bi mogli imati znak c na prvom mjestu. Da bi to osigurali moramo napraviti još sve permutacije nezavršnih znakova :

AB→BA
AC→CA
BC→CB
BA→AB
CA→AC
CB→BC

I to je to, nadam se da ste nešto shvatili ☺

31. Konstruirati gramatiku koja generira nizove oblika $a^i b^j c^k d^i e^j$ pri čemu su *i*,*j*,*k*≥1.

S ovakvim zadacima gdje se traži konstruiranje gramatike (ne kontekstno ovisne) gdje su nizovi ovakvog tipa, znači neki znak i eksponent, su jako jednostavni ako slijedite 'Diablovu kuharicu' ☺

Kuharica :
- ako niz može biti prazan niz odmah se stavlja produkcija S -> ε
- drugi korak je da se stvara početna produkcija gdje se desna strana sastoji od po jednog različitog završnog znaka, i toliko nezavršnih znakova koliko ima potencija različitih stupnjeva
- svi ti nezavršni znakovi idu u **ε** (u slučaju minimalnog niza jel)
- za svaki taj nezavršni znak stvoriti produkciju koja se sastoji od pripadnog završnog znaka, tog nezavršnog, i onoliko dodatnih nezavršnih koliko ima potencija istog stupnja
- za te novonastale nezavršne znakove napravi produkciju koja će se šetati po svim znakovima do svojeg pripadnog završnog
- kad dođe do završnog, treba produkcija koja će ga pretvoriti u taj završni

Pa krenimo ☺

- *ako niz može biti prazan niz odmah se stavlja produkcija S -> ε*

Mi imamo zadano da su ***i*, *j*, *k*≥1,** što znači da niz ne može biti prazan.

- *drugi korak je da se stvara početna produkcija gdje se desna strana sastoji od po jednog različitog završnog znaka, i toliko nezavršnih znakova koliko ima potencija različitih stupnjeva*

Imamo 5 različitih završnih znakova i 3 potencije različitog stupnja pa produkcija izgleda :

S -> aAbBcCde

- *svi ti nezavršni znakovi idu u **ε** (u slučaju minimalnog niza jel)*

A -> **ε** **B -> ε** **C -> ε**

- *za svaki taj nezavršni znak stvoriti produkciju koja se sastoji od pripadnog završnog znaka, tog nezavršnog, i onoliko dodatnih nezavršnih koliko ima još* potencija istog *stupnja*

Mi imamo još po jednu potenciju stupnja i pa ide nezavršni znak D, još jednu potenciju stupnja j pa ide još jedan nezavršni znak E, ali nema više duplih potencija stupnja k.

A -> aAD B -> bBE C -> cC

- *za te novonastale nezavršne znakove napravi produkciju koja će se šetati po svim znakovima do svojeg pripadnog završnog*

Db -> bD   Ec -> cE
Dc -> cD   Ed -> dE

- *kad dođe do završnog, treba produkcija koja će ga pretvoriti u taj završni*

Dd -> dd   Ee -> ee

I to je to ☺
P.S. da ne bi mislili da je to slučajnost, provjerite zadatak 33. iz zzv3 i zadatak 8 iz ZI 06/07

32. Pretvoriti zadanu gramatiku s neograničenim produkcijama u konteksno ovisnu gramatiku.

S→aAbBcCde	**B→bBE**	**C→cC**	**Db→bD**	**Ec→cE**
A→aAD	**B→ε**	**C→ε**	**Dc→cD**	**Ed→dE**
A→ε			**Dd→dd**	**Ee→ee**

Kod kontekstno ovisne gramatike jedini je uvjet da desna strana produkcije mora biti jednaka ili veća od lijeve strane. To odmah poteže za sobom da ne smije biti epislon produkcija jer je njoj duljina desne strane jednaka 0.
Ovo je dosta glup primjer kao što je i sam asistent rekao, jer u ovom slučaju treba samo izbaciti epsilon produkcije algoritmom iz drugog ciklusa i to je kraj zadatka ☺

33. Konstruirati gramatiku koja generira nizove oblika $0^n1^n2^n$ pri čemu je $n \geq 0$.

Evo ovaj ćemo isto riješit preko 'Diablove kuharice' ☺

- sad nam n može biti 0 pa prvo radimo produkciju :

S -> ε

- imamo 3 različita završna znaka i samo 1 stupanj potencija pa nam produkcija izgleda ovako

S -> 0A12

- taj nezavršni znak mora ić odma u epsilon za slučaj kad je niz minimalan :

A -> ε

- sad se stvara produkcija od pripadnog završnog, tog nezavršnog i još dodatnih nezavršnih koliko je još potencija istog stupnja (ovdje još dvije) :

A –> 0ABC

- sad treba stvoriti produkcije za šetanje do svojih završnih :

C1 -> 1C
CB -> BC

- i na kraju stvoriti produkcije koje pretvaraju nezavršni u pripadni završni znak :

C2 -> 22
B1 -> 11

Ovo rješenje se malo razlikuje od onog u zadacima za vježbu ali je isto dobro, evo primjer za n=3

S -> 0**A**12 -> 00**A**BC12 -> 000**A**BCBC12 -> 000BCB**C1**2 -> 000B**CB**1C2 -> 000BB**C1**C2 -> 000B**B1**C**C2** -> 000**B1**1**C2**2 -> 000111222

34. Konstruirati konteksno ovisnu gramatiku koja generira nizove iz jezika L=$\{0^n1^n2^n \mid n \geq 1\}$.

E sad pošto se dobro rješenje dobije isto samo odstranjivanjem epsilon produkcija, nećete radit prvo gramatiku neograničenih produkcija i izbacivali epsilon produkcije, nego morate znat drito konstruirat kontekstno ovisnu gramatiku.

Na početku ćemo stvoriti produkciju na čijoj će desnoj strani biti dva nezavršna znaka, jedan za generiranje nezavršnih i završnih znakova, a jedan kao desni graničnik :

S -> AX

Sada nezavršni znak A će generirati po jedan njegov završni znak i jedan nezavršni znak za generiranje jednakog broja drugih završnih znakova, i to u minimalnom slučaju kad je n=1, ili neograničenom slučaju :

A -> 0B
A -> 0AB

Za krajnji B koji je odmah pokraj graničnika X može generirati po jednu jedinicu i dvojku, a kad je više znakova B onda se mora pozicionirati između skupa brojeva 1 i 2

BX -> 12
B1 -> 1B
B2 -> 122

Završni ispit 2006/07 – tutorial by Diablo

1. Opisati postupak pretvorbe ε-NKA u NKA.

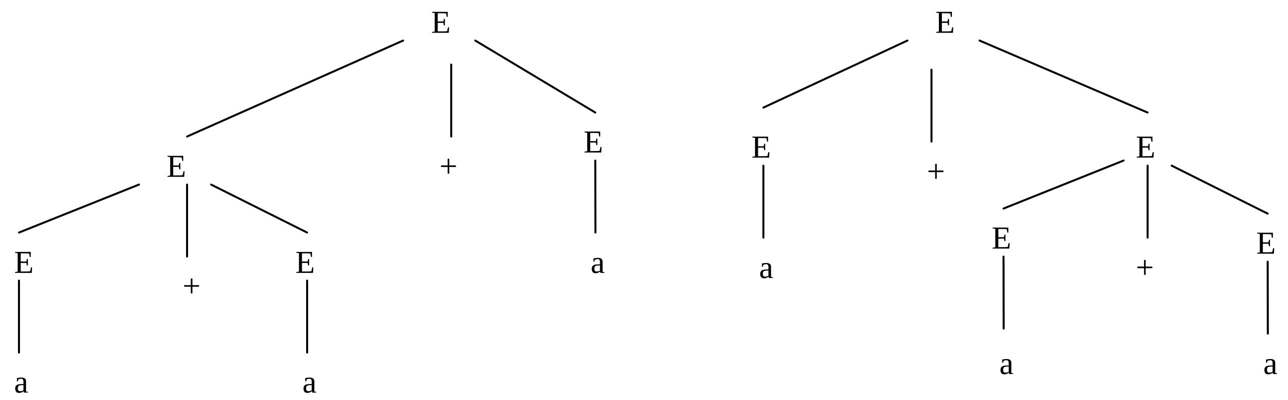
Za bilo koji ε-NKA M=(Q,∑,δ,q0,F) moguće je izgraditi istovjetni NKA M'=(Q',∑,δ',q0',F') na sljedeći način :

Q' = Q,
q0' = q0
F' = F u {q0} ako ε-OKRUŽENJE(q0) sadrži barem jedno stanje skupa F, inače F' = F,
δ'(q,a) = δ(q,a), za svaki a element ∑ i za svaki q element Q

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

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

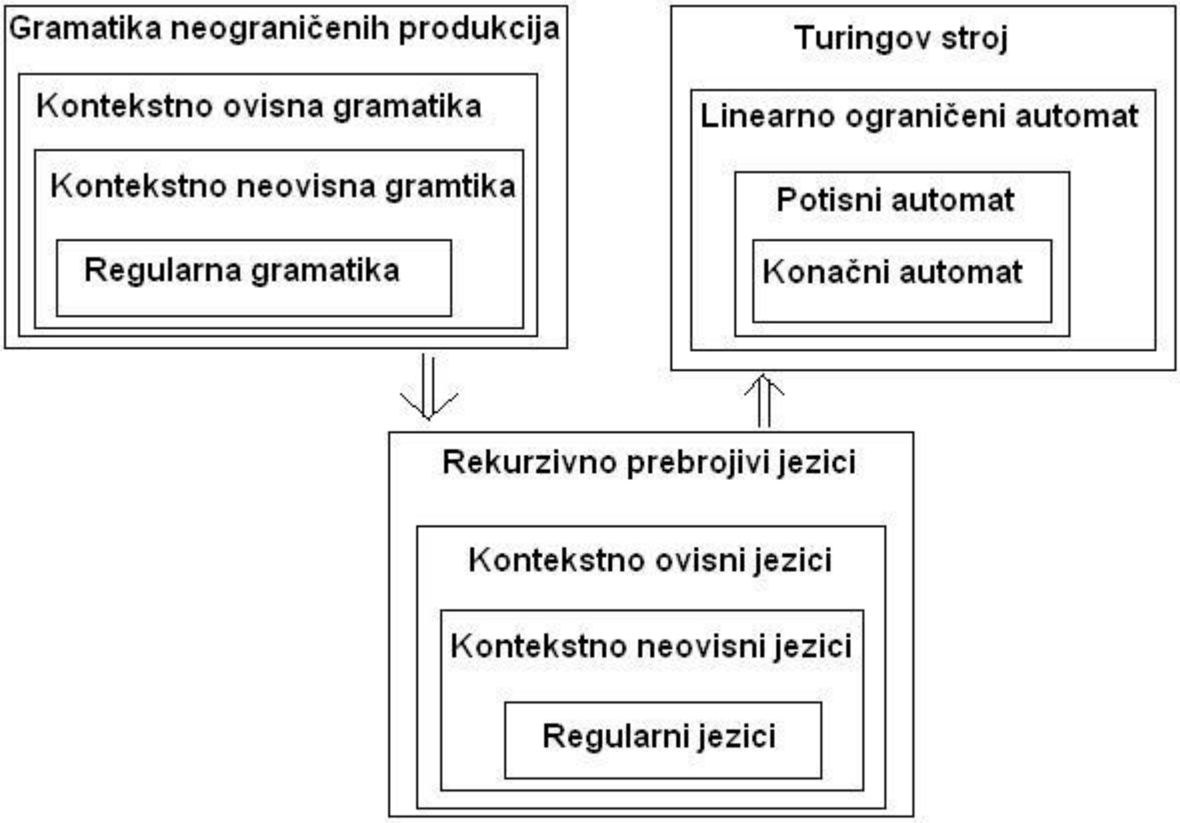
Primjer : a + a + a



3. Opisati algoritam odbacivanja mrtvih znakova za kontekstno-neovisnu gramatiku.

- U listu živih znakova stave se lijeve strane produkcija koje na desnoj strani nemaju nezavršnih znakova,
- Ako su na desnoj strani produkcije isključivo živi znakovi, onda se nezavršni znak lijeve strane produkcije doda u listu živih znakova,
- Ako nije moguće proširiti listu živih znakova, onda se algoritam zaustavlja. Svi znakovi koji nisu u listi živih znakova su mrtvi znakovi.

4. Nabrojati i opisati hijerarhiju jezika, gramatika i automata.



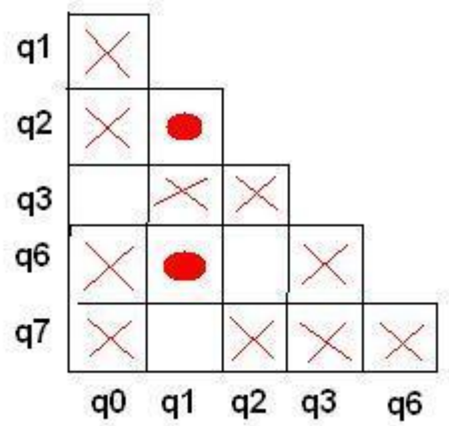
5. Utvrdite da li nadovezivanjem kontekstno-ovisnih jezika nastaje kontekstno-ovisan jezik i objasnite tvrdnju.

Nadovezivanje kontekstno-ovisnih jezika jest kontekstno-ovisan jezik.
Neka gramatika G1 = (V1, T1, P1, S1) generira jezik L(G1), a gramatika
G2 = (V2, T2, P2, S2) neka generira jezik L(G2). Gramatika G4 = (V4, T4, P4, S4) koja generira jezik L(G4) = L(G1)L(G2) konstruira se na sljedeći način :
V4 = V1 u V2 u {S4}, gdje S4 nije element V1 ni V2, te je presjek V1 i V2 prazan skup
T4 = T1 u T2
U skup produkcija P4 = P1 u P2 doda se produkcija : S4 -> S1S2

6. Minimizirajte sljedeći DKA :

	a	b	c	
q0	q0	q7	q2	1
q1	q1	q6	q0	0
q2	q2	q3	q1	0
q3	q3	q1	q2	1
q4	q2	q2	q1	0
q5	q4	q0	q1	1
q6	q6	q3	q1	0
q7	q1	q2	q0	0

Prvo se traže dohvatljiva i nedohvatljiva stanja :
DOHVATLJIVA = {q0, q2, q7, q1, q3, q6}
NEDOHVATLJIVA = {q4, q5}



Lista :
(q1, q7) = {(q0, q3)}
(q2, q6) = {(q1, q7)}

Ekvivalentna stanja :
q0 = q3
q1 = q7
q2 = q6

Rješenje :

	a	b	c	
q0	q0	q1	q2	1
q1	q1	q2	q0	0
q2	q2	q0	q1	0

7. Konstruirati gramatiku nad abecedom {0, 1, 2} koja generira sve nizove abecede u kojima se ne pojavljuju podnizovi 01 i 21.

Za početak imamo početno stanje S iz kojeg možemo generirati bilo koji znak, pa tako i znak 1 :
S -> 1S

Kad generiramo 1 to nam ne povlači nikakve dodatne uvjete pa ostajemo u istom stanju. Možemo generirati i znak 0 :

S -> 1S | 0A

Pošto nakon pojavljivanja znaka 0 se ne smije pojaviti znak 1, odlazimo u novo stanje A u kojem ćemo se pobrinuti da se to ne dogodi. Dalje možemo generirati znak 2 :

S -> 1S | 0A | 2B

Isto kao i u prethodnom slučaju prelazimo u novo stanje da bi osigurali da se ne pojavi podniz 21. Na kraju moguće je generirati i prazni niz :

S -> 1S | 0A | 2B | e

Sada za nova stanja moramo definirati prijelaze za svaki ulazni znak :

S -> 1S | 0A | 2B | e
A -> 0A | 2B | e
B -> 2B | 0A | e

Time smo osigurali da se ne pojave zadani podnizovi. Zadatak se mogao riješiti sa samo jednim novim stanjem.

8. Konstruirati gramatiku koja generira jezik L = $\{a^i b^j c^i d^j \mid i,j >= 1\}$.

- *ako niz može biti prazan niz odmah se stavlja produkcija S -> ε*

Mi imamo zadano da su ***i, j*** **≥ 1,** što znači da niz ne može biti prazan.

- *drugi korak je da se stvara početna produkcija gdje se desna strana sastoji od po jednog različitog završnog znaka, i toliko nezavršnih znakova koliko ima potencija različitih stupnjeva*

Imamo 4 različita završna znaka i 2 potencije različitog stupnja pa produkcija izgleda :

S -> aAbBcd

- *svi ti nezavršni znakovi idu u* **ε** *(u slučaju minimalnog niza jel)*

A -> ε   **B -> ε**

- *za svaki taj nezavršni znak stvoriti produkciju koja se sastoji od pripadnog završnog znaka, tog nezavršnog, i onoliko dodatnih nezavršnih koliko ima još potencija istog stupnja*

Mi imamo još po jednu potenciju stupnja i pa ide nezavršni znak C, još jednu potenciju stupnja j pa ide još jedan nezavršni znak D.

A -> aAC   B -> bBD

- *za te novonastale nezavršne znakove napravi produkciju koja će se šetati po svim znakovima do svojeg pripadnog završnog*

Cb -> bC   Dc -> cD

- *kad dođe do završnog, treba produkcija koja će ga pretvoriti u taj završni*

Dd -> dd   Cc -> cc

9. Konstruirati kontekstno ovisnu gramatiku koja generira nizove oblika $0^n1^{3n}2^{2n}$ pri čemu je n>0. Pažljivo pročitajte koji su eksponenti u zadatku!

Na početku ćemo stvoriti produkciju na čijoj će desnoj strani biti dva nezavršna znaka, jedan za generiranje nezavršnih i završnih znakova, a jedan kao desni graničnik :

S -> AX

Sada nezavršni znak A će generirati po jedan njegov završni znak 0 i jedan nezavršni znak za generiranje trostrukog broja 1, i to u minimalnom slučaju kad je n=1, ili neograničenom slučaju :

A -> 0J
A -> 0AJ

Sada napravimo produkciju iz koje će nezavršni znak J generirati tri jedinice i novi nezavršni znak koji će generirati dvostruko veći broj dvojki od jedinica :

J -> 111D

Za krajnji D koji je odmah pokraj graničnika X ili pokraj znaka 2 može generirati po dvije dvojke, a kad je između skupa jedinica treba mu omogućiti šetanje do desnog kraja :

DX -> 22
D1 -> 1D
D2 -> 222

Provjera : **S** -> **A**X -> 0**A**JX -> 00**JJ**X -> 00111D111**DX** -> 00111**D1**1122 -> 00111111**D2**2 -> 001111112222

10. Konstruirati Turingov stroj s dvostrukom beskonačnom trakom i jednom glavom za čitanje koji traži i provjerava ispravnost niza zapisanog na traci. Niz je zapisan u obliku $w_1\#w_2\#w_3$, gdje su w_1, w_2 i w_3 nizovi nula , a pretpostavka je da su u svim ostalim ćelijama na traci praznine. Dodatno, vrijedi $[w_2] = [w_1] + 2[w_3]$, pri čemu oznaka [x] predstavlja broj nula u nizu x. Na početku rada Turingovog stroja nije poznat položaj glave u odnosu na niz na traci.
Napomena : objesniti ideju i značenje pojedinih stanja i prijelaza koje TS koristi.

Da bi uopće mogli početi rješavati zadatak, moramo dobro shvatiti kako izgleda početno stanje trake i što se točno od nas traži. Naše početno stanje trake izgleda recimo ovako :

B	0	0	#	0	0	0	0	0	0	#	0	0	B

Legenda : promjene na traci su označene crvenom, a novi prijelazi plavom bojom.
Dakle, imamo zapisan niz u obliku $w_1\#w_2\#w_3$ gdje vrijedi $[w_2] = [w_1] + 2[w_3]$. Oznakama B smo označili beskonačan broj ćelija u lijevo, odnosno u desno. Glava nam se ne nalazi na početku niza u ovom konkretnom primjeru, te ćemo morati prvo postaviti glavu na početak. To ćemo ostvariti početnim stanjem q0 i prijelazima (q0,0)=(q0,0,L), (q0,#)=(q0,#,L) i (q0,B)=(q1,B,R). Stanje q1 nam označava da se nalazimo na krajnjem lijevom znaku 0 podniza w1.

B	0	0	#	0	0	0	0	0	0	#	0	0	B

E sad, logika provjere da li vrijedi $[w_2] = [w_1] + 2[w_3]$ je sljedeća : za svaki znak 0 prepisan znakom N podniza w1 prepiši jedan znak 0 podniza w2, te za svaki prepisani znak 0 podniza w3 prepiši po dva znaka 0 podniza w2. Ako je sve dobro zadano svi znakovi 0 će biti prepisani znakovima N.
Prema logici slijedi da ćemo za stanje q1 u kojem se trenutno nalazimo i ulazni znak 0 prelaziti u novo stanje q2, te prepisivati ulazni znak 0 sa znakom N. Stanje q2 će nam služiti za šetnju u desno preko cijelog podniza w1 sve do graničnika # nakon kojeg znamo da nam počinje podniz w2, te ćemo zato preći i u novo stanje q3.

B	N	0	#	0	0	0	0	0	0	#	0	0	B

Prijelazi glase : (q1,0)=(q2,N,R), (q2,0)=(q2,0,R) i (q2,#)=(q3,#,R).
Sada kad se nalazimo u stanju q3 ćemo za ulazni znak 0 prijeći u novo stanje q4 i prepisati ulazni znak 0 sa znakom N. Stanje q4 nam služi za šetnju već prepisanog niza w2 do graničnika # (sad na početku smo odmah na graničniku, drugi put će biti izraženije), gdje ćemo prijeći u novo stanje q5 sa kojim ćemo se pozicionirati na krajnji lijevi znak 0 podniza w1 (u primjeru je podniz w1 malen, ali za veći broj 0 trebalo bi se prošetati do krajnje lijeve 0) i završiti ponovno u stanju q1.

B	N	0	#	N	0	0	0	0	0	#	0	0	B

Prijelazi glase : (q3,0)=(q4,N,L), (q4,#)=(q5,#,L), (q5,0)=(q5,0,L) i (q5,N)=(q1,N,R).
Ponavljamo postupak za sve znakove 0 podniza w1, tako dugo kada ne dođemo do graničnika #.

B	N	N	#	N	N	0	0	0	0	#	0	0	B

Kada nam je za stanje q1 ulazni znak # znamo da smo iskoristili cijeli podniz w1, pa ćemo prijeć u stanje q6 kojim ćemo prošetati preko cijelog podniza w2 do novog graničnika #, te prijeći u novo stanje q7.

B	N	N	#	N	N	0	0	0	0	#	0	0	B

Prijelazi glase : (q1,#)=(q6,#,R), (q6,N)=(q6,N,R), (q6,0)=(q6,0,R) i (q6,#)=(q7,#,R).
Stanje q7 u podnizu w3 nam ima jednaku ulogu kao i stanje q1 u podnizu w1.
Ulazni znak 0 ćemo prepisati znakom N te prijeći u novo stanje q8 kojim ćemo se šetati preko svih znakova N podniza w3 sve do graničnika #. Zatim prelazimo u stanje q9 kojim ćemo se pozicionorati na krajnji lijevi znak 0 podniza w2.

B	N	N	#	N	N	0	0	0	0	#	N	0	B

Prijelazi glase : (q7,0)=(q8,N,L), (q8,#)=(q9,#,L), (q9,0)=(q9,0,L) i (q9,N)=(q10,N,R).
Stanje q10 nam služi za prepisivanje prve 0 znakom N, te ćemo prijeć u novo stanje q11 koje će nam služit za prepisivanje drugog znaka 0 znakom N (jer smo prije rekli da za jedan znak 0 podniza w3 prepisujemo dva znaka 0 u podnizu w2). Prelazimo u stanje q6 gdje ćemo prošetati preko cijelog podniza w2, te zatim u q7 gdje se ponovo pozicioniramo na krajnji lijevi znak 0 podniza w3.

B	N	N	#	N	N	N	N	0	0	#	N	0	B

Prijelazi : (q10,0)=(q11,N,R), (q11,0)=(q6,N,R), (q6,0)=(q6,0,R), (q6,#)=(q7,#,R) i (q7,N)=(q7,N,R).
Sada jednostavno ponovimo postupak dok ne iskoristimo cijeli podniz w3.

B	N	N	#	N	N	N	N	N	N	#	N	N	B

Kad nam je u stanju q7 ulazni znak B, znamo da smo iskoristili cijeli podniz w3 i došli na kraj niza. Sada prelazimo u novo stanje q12 kako bi prošetali od kraja niza na početak, pritom pretvarajući sve znakove N u znakove 0.

B	0	0	#	0	0	0	0	0	0	#	0	0	B

Prijelazi : (q7,B)=(q12,B,L), (q12,N)=(q12,0,L), (q12,#)=(q12,#,L) i (q12, B)=(qf,B,R).
Na kraju smo ulaznim znakom B prešli u JEDINO prihvatljivo stanje qf. Niz smo vratili na početni izgled. Da je bilo viška znakova 0 (tj da niz nije dobro zadan) Turingov stroj bi stao zbog nedefiniranih prijelaza, a pošto niti jedno stanje nije prihvatljivo osim konačnog stanja qf, niz bi bio neprihvatljiv.

Konačna tablica prijelaza :

	0	#	B	N
q0	q0,0,L	q0,#,L	q1,B,R	
q1	q2,N,R	q6,#,R		
q2	q2,0,R	q3,#,R		
q3	q4,N,L			q3,N,R
q4		q5,#,L		q4,N,L
q5	q5,0,L			q1,N,R
q6	q6,0,R	q7,#,R		q6,N,R
q7	q8,N,L		q12,B,L	q7,N,R
q8		q9,#,L		q8,N,L
q9	q0,0,L			q10,N,R
q10	q11,N,R			
q11	q6,N,R			
q12		q12,#,L	qf,B,R	q12,0,L

Završni ispit 2007/08

(Izvučeno iz arhive, pisano po sjećanju by Jack_Frost
+ neka rješenja by Diablo)

1. Razlika između rekurzivnih i rekurzivno prebrojivih jezika.

Svaki rekurzivni jezik ujedno je i rekurzivno prebrojiv, obrat ne vrijedi.
Rekurzivni jezici su izračunljivi i odlučni, a rekurzivno prebrojivi jezici su izračunljivi ali ne i odlučni.
Rekurzivni jezici zatvoreni su s obzirom na operaciju unije i komplementa, a rekurzivno prebrojivi jezici su zatvoreni s obzirom na operaciju unije.

2. Opisati hijerarhiju jezika, automata i gramatika.

Identično ZI 06/07 zadatak 4.

3. Izmisliti gramatiku koja generira xml tagove, tipa <aabbcc> i </aabbcc>. Primijeti da su nizovi unutar <> i </> jednaki! (ono što se otvori treba se i zatvoriti).

4. Konstruirati Turingov stroj s jednom trakom koji dijeli dva cijela broja zapisana nizom nula. Znači '00000/00=', glava je na početku bila na znaku '=', rezultat napisati poslije '=' tako da prvo piše rezultat, i ostatak ako ga ima. Za ovaj gore primjer je to 00000/00=00%0, znači 2 i ostatak 1.

Kao i dosad, da bi uopće mogli početi rješavati zadatak, moramo dobro shvatiti kako izgleda početno stanje trake i što se točno od nas traži. Naše početno stanje trake izgleda recimo ovako :

B	0	0	0	0	0	/	0	0	=	B	B	B	B

Legenda : promjene na traci su označene crvenom, a novi prijelazi plavom bojom.
Imamo broj 5 prikazan brojem nula kojeg dijelimo brojem 2. Početna pozicija glave je iznad '='. Lijevo od niza brojeva nalazi se prazna ćelija označena oznakom B, a također se prazne ćelije nalaze i nakon znaka '='.
Logika kojom ćemo riješiti ovaj zadatak glasi : za svaki znak 0 broja 2 kojeg prepišemo znakom X, prepisujemo po jedan znak 0 broja 5 također znakom X. Kad iskoristimo sve znakove broja 2 tada dopisujemo jedan znak 0 nakon '=' (kao znak da broj 2 ulazi u broj 5 jedan puta). To ponavljamo toliko puta koliko je to moguće (znači 2 puta). Treći puta će nam manjkat znakova 0 kod broja 5 pa onda nadopišemo na rezultat znak '%' i toliko nula koliko znakova 0 u broju 2 jest našlo par u broju 5 u trećem krugu kao ostatak.

Na početku se nalazimo u stanju q0 pa ćemo se pomaknuti lijevo na najdesniji znak 0 broja 2 i prijeći u stanje q1.

Prijelaz : (q0,=)=(q1,=,L).

B	0	0	0	0	0	/	0	0	=	B	B	B	B

U stanju q1 ćemo ulazni znak 0 broja 2 prepisati sa znakom X i prijeći u novo stanje q2 kojim ćemo se šetati preko cijelog broja sve do graničnika '/', gdje zatim prelazimo u stanje q3.

B	0	0	0	0	0	/	0	X	=	B	B	B	B

Prijelazi : (q1,0)=(q2,X,L), (q2,0)=(q2,0,L), (q2,/)=(q3,/,L).
Sada ćemo u stanju q3 ulazni znak 0 broja 5 prepisati znakom X, prijeći u stanje q4 kojim ćemo se prošetati preko svih dosad prepisanih znakova broja 5 sve do graničnika '/', gdje prelazimo u stanje q5 kojim se pozicioniramo na krajnji desni znak 0 broja 2.

B	0	0	0	0	X	/	0	X	=	B	B	B	B

Prijelazi : (q3,0)=(q4,X,R), (q4,/)=(q5,/,R), (q5,0)=(q5,0,R) i (q5,X)=(q1,X,L).
Ponovimo još jednom postupak za drugi znak 0 broja 2.

B	0	0	0	X	X	/	X	X	=	B	B	B	B

Prijelazi : (q1,0)=(q2,X,L), (q2,/)=(q3,/,L), (q3, X)=(q3,X,L), (q3,0)=(q4,X,R), (q4,X)=(q4,X,R), (q4,/)=(q5,/,R) i (q5,X)=(q1,X,L).
Sada imamo za stanje q1 ulazni znak '/' što znači da smo iskoristili sve znakove 0 broja dva uspješno, te idemo nakon znaka '=' nadodati jednu nulu kao znak da broj 2 ulazi jednom u broj 5. Za to nam je potrebno stanje q6 kojim ćemo se prošetati po svim znakovima do polja nakon znaka '='. Međutim za vrijeme šetanja preko broja 2 ćemo sve znakove X usput pretvoriti natrag u znakove 0.

B	0	0	0	X	X	/	0	0	=	B	B	B	B

Prijelazi : (q1,/)=(q6,/,R), (q6,X)=(q6,0,R) i (q6,=)=(q6,=,R).
Nalazimo se na prvoj praznoj ćeliji nakon znaka '=' u koju ćemo zapisati znak 0, te prijeći u stanje q7 koje nam služi da prošetamo preko svih dosad već zapisanih znakova 0 ponovo do znaka '=', nakon kojega se vraćamo u stanje q1.

B	0	0	0	X	X	/	0	0	=	0	B	B	B

Prijelazi : (q6,B)=(q7,0,L) i (q7,=)=(q1,=,L).
Ponoviti postupak i drugi put.

B	0	0	X	X	X	/	0	X	=	0	B	B	B

B	0	X	X	X	X	/	X	X	=	0	B	B	B

Ne da mi se gledat, ali mislim da nema novih prijelaza ☺
Sad pošto smo opet uspješno iskoristili sve znakove 0 broja 2 idemo dodati još jednu nulu nakon znaka '=', što znači da nam broj dva u broj pet ulazi dva puta. Ponovo usput prebacujemo sve znakove X broja 2 u znakove 0.

B	0	X	X	X	X	/	0	0	=	0	B	B	B

Prijelazi : (q1,/)=(q6,/,R), (q6,X)=(q6,0,R), (q6,=)=(q6,=,R) i (q6,0)=(q6,0,R).
U praznu ćeliju upisujemo znak 0, prelazimo u stanje q7 kojim prelazimo preko cijelog zasad napisanog broja te nakon znaka '=' prelazimo u stanje q1.

B	0	X	X	X	X	/	0	0	=	0	0	B	B

Prijelazi : (q6,B)=(q7,0,L), (q7,0)=(q7,0,L) i (q7,=)=(q1,=,L).
Sada i treći put radimo jednu te istu stvar.

B	X	X	X	X	X	/	0	X	=	0	0	B	B

B	X	X	X	X	X	/	X	X	=	0	0	B	B

Ovdje je sad nastao problem što posljednji znak 0 broja 2 nema svoj par, što znači da nije treći put djeljivo nego imamo ostatak. Ostatak je upravo onoliko koliko nula je našlo par u ovome trećem krugu. Par je našla samo desna nula broja 2, pa je ostatak jedan. Morat ćemo prijeći u novo stanje q8 kojim ćemo se prošetati preko svih znakova broja 5 sve do znaka '/', te zatim prijeć u novo stanje q9 u kojem ćemo se pozicionirati na najlijeviji znak X broja 2.

B	X	X	X	X	X	/	X	X	=	0	0	B	B

Prijelazi : (q3,B)=(q8,B,R), (q8,X)=(q8,X,R) i (q8,/)=(q9,/,R).
Najprije treba taj prvi znak X pretvoriti u 0 jer to je ta 0 koja nije našla par. Ostali znakovi X broja dva su upravo traženi ostatak, te za svaki znak X treba napisati jedan znak 0 poslije graničnika % koji razdvaja cijeli broj od ostatka.

B	X	X	X	X	X	/	0	0	=	0	0	%	B

Prijelazi : (q9,X)=(q10,0,R), (q10,X)=(q11,0,R), (q11,=)=(q11,=,R), (q11,0)=(q11,0,R) i (q11,B)=(q12,%,R).
U prvu praznu ćeliju nakon znaka % upisujemo jedan znak 0 kao ostatak, te prelazimo u novo stanje q13 kojim ćemo se šetat ulijevo do znaka '=', prijeć u stanje q14 s kojim ćemo se pozicionirati na krajnji lijevi znak X broja 2 ako ga ima.

B	X	X	X	X	X	/	0	0	=	0	0	%	0

Prijelazi : (q12,B)=(q13,0,L), (q13,%)=(q13,%,L), (q13,0)=(q13,0,L), (q13,=)=(q14,=,L) i (q14,0)=(q10,0,R).
Nakon što smo napisali cijeli ostatak, još moramo prijeć u stanje q15 kako bi sve znakove X vratili u znakove 0 i dobili početni zapis. Kad dođemo do prazne ćelije prelazimo u prihvatljivo stanje.

B	0	0	0	0	0	/	0	0	=	0	0	%	0

Prijelazi : (q10,=)=(q15,=,L), (q15,0)=(q15,0,L), (q15,/)=(q15,/,L), (q15,X)=(q15,0,L) i (q15,B)=(qf,B,R).

NAPOMENA : To nisu svi prijelazi, jer za neke druge slučajeve trebaju prijelazi (q14,X)=(q14,X,L), (q11,%)=(q12,%,R), (q12,0)=(q12,0,R). Pokušajte za vježbu riješiti primjer gdje 8 dijelite s 3.

5. Bio je jedan u kojem je trebalo odbaciti sve jedinične i epsilon produkcije.

6. Opisati algoritam odbacivanja mrtvih znakova.

Isti kao ZI 06/07 zadatak 3.

7. Iz zadanog DKA napisati regularni izraz (ne sjećam se automata).

8. Turing sličan kao u zadacima za pripremu, jedna traka, jedan trag, nizovi nula odvojeni sa #. Znači w1#w2#w3. Provjeriti da je (mislim, ne sjećam se točno) [w1]=[w2]+2[w3] (al nisam siguran)

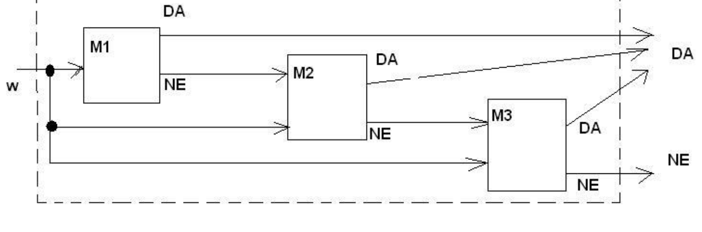
Ovo je možda bio isti zadatak kao ZI 06/07 zadatak 10, međutim ako se inzistiralo na traci i jednom tragu umjesto dvostruko beskonačne trake, onda je rješenje drukčije.

9. Opiši i definiraj univerzalni Turingov stroj.

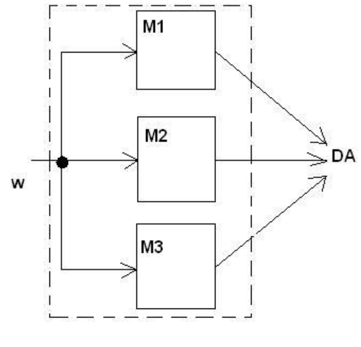
Knjiga 'Uvod u teoriju računarstva' strana 149.
Univerzalni TS ima tri trake. Na prvu traku se zapišu funkcije prijelaza TS M i niz w. Zapis funkcije prijelaza i niza označuje se na sljedeći način : <M, w>. Na treću traku zapisuje se stanje TS M. Sadržaj druge trake univerzalnog TS Mu simulira sadržaj trake proizvoljnog TS M. Univerzalni prepiše niz w s prve trake na drugu traku, simulira rad TS M primjenom prijelaza zapisanih na prvoj traci i zapisuje stanje TS M na treću traku. Nema li za stanje zapisano na trećoj traci i za znak zapisan na drugoj traci daljnjih prijelaza zapisanih na prvoj traci, zaustavlja se rad univerzalnog TS Mu. Ako je stanje zapisano na trećoj traci prihvatljivo, onda univerzalni TS Mu prelazi u prihvatljivo stanje. TS Mu prihvaća niz <M,w> ako i samo ako TS M prihvaća niz w.

10. Skiciraj kako se pomoću Turingovih strojeva konstruiraju automati koji prihvaćaju uniju :
a) 3 rekurzivna jezika,
b) 3 rekurzivno prebrojiva jezika.

a)



b)



Završni ispit 2008/09

1. Opisati algoritam odbacivanja mrtvih znakova iz kontekstno-neovisne gramatike.

Isti kao ZI 06/07 zadatak 3.

2. Definirati univerzalni Turingov stroj i objasniti princip rada.

Isti kao ZI 07/08 zadatak 9.

3. Objasniti razliku između rekurzivnih i rekurzivno prebrojivih jezika.

Isti kao ZI 07/08 zadatak 1.

4. Skicirati hijerarhiju formalnih jezika, automata i gramatika.

Isti kao ZI 06/07 zadatak 4.

5. Opisati postupak i nacrtati shematski prikaz konstrukcije Turingovog stroja koji prihvaćaju uniju :
a) tri rekurzivna jezika
b) tri rekurzivno prebrojiva jezika

Isti kao ZI 07/08 zadatak 10.

6. Dokazati jednakost jezika definiranih sljedećim regularnim izrazima ili jednakost opovrgnuti protuprimjerom :
a) ab + c = (a + c)(b + c)

b) $a^*(a^* + b^*)^* b^* = a(a+b)^* b$

a) ab + c = (a + c)(b + c)
= ab + ac + cb + c
= ab + c (1 + b + c)
= ab + c => jer 1 plus bilošta je 1

7. Za zadanu gramatiku konstruirati automat najjednostavnijeg razreda koji prihvaća isti jezik.

S->abA	**A->aB**	**B->abcC**	**C->acS**
	A->C	**B->cbcC**	**C-> ε**
	A->a		
	A->ε		

8. Konstruirati osnovni model TS koji računa kvadrat pozitivnog broja zapisanog na traci. Broj je na traci zapisan kao niz nula, a vrijednost broja odgovara broju nula u nizu. U krajnje lijevu ćeliju trake zapisan je znak #, nakon čega neposredno slijedi pozitivan broj znakova 0. Glava za čitanje i pisanje se na početku rada nalazi na znaku #. Konačno stanje trake mora odgovarati početnom uz dodatak graničnika # i traženog kvadrata zapisanog nizom nula. Na primjer, za kvadrat broja dva, početno stanje trake je #00, a konačno #00#0000.

Jedna mogućnost rješavanja : svaku znamenku 0 zadanog broja prepisati onoliko puta od koliko se taj broj znamenki 0 sastoji. Recimo broj 3 ima tri nule 000, pa svaki znak 0 prepisujemo tri puta jer se broj 3 sastoji od tri nule, pa dobijemo na kraju broj 9.

9. Na osnovi zadanog Turingovog stroja <u>s obostranom beskonačnom trakom</u> M= ({q0, q1, q2, q3, q4}, {a, b}, {a, b, X, B}, δ, q0, B, {q4}) konstruirati gramatiku koja prihvaća isti jezik. Potrebno je navesti <u>sve</u> produkcije za generiranje početne konfiguracije TS-a, <u>sve</u> produkcije za generiranje praznih ćelija i <u>sve</u> produkcije za prihvaćanje niza, dok je od produkcija koje proizlaze iz funkcije prijelaza dovoljno navesti <u>samo one</u> koje proizlaze iz funkcije prijelaza za stanje q3 (<u>osjenčana polja u tablici</u>).

	a	**b**	**X**	**B**
q0	q1,a,R	q0,b,R	q0,X,R	q1,B,L
q1	-	q2,B,R	q3,X,L	-
q2	q0,X,R	q2,b,L	q2,X,L	q0,X,R
q3	-	-	q3,X,L	q4,B,R
q4	-	-	-	-

10. Konstruirati gramatiku koja generira *for* petlje sljedećeg oblika :

for(int w=0;w<n;++w)x;

pri čemu vrijedi w,x € (a + b)*. (Napomena: u nizu se nalazi samo jedan razmak nakon ključne riječi *int*).

Završni ispit 2009/10

1. Skicirati hijerarhiju formalnih jezika, automata i gramatike.

Isti kao ZI 06/07 zadatak 4.

2. Formalno definirati osnovni model Turingovog stroja

3. U okvir uz svaki zadani jezik upisati klasu jezika iz Chomskyjeve hijerarhije najmanje strukturne složenosti u kojoj je zadani jezik sadržan:

Ld (dijagonalni jezik)
[svi binarni brojevi veći od 128]
[wwRwR|w E (a+b)^+]
Lu (univerzalni jezik)
[wwR|w E (a+b)^+]
[w|w E (a+b+c+d)^+ u kojima je na+nb=nc+nd]

4. Definirati polinomnu vremensku svodivost. Definirati kada je neki jezik L potpun, a kada je težak s obzirom na neku klasu jezika K i polinomnu vremensku svodisvost.

5. Definirati algoritam za izgradnju gramatike neograničenih produkcija za jezik kojeg prihvaća neki zadani Turingov stroj u osnovnom modelu.

6. Iz zadanog Mealyjevog automata konstruirati istovjetni Mooreov automat:

δ	a	b	c
q0	q1	q2	q0
q1	q0	q1	q1
q2	q1	q1	q0

λ	a	b	c
q0	X	X	Y
q1	Y	Y	Z
q2	Z	Y	X

7. Za zadanu kontekstno neovisnu gramatiku izgraditi parser zasnovan na tehnici rekurzivnog spusta. Parser opisati pseudokodom.

S-->aAb	**A-->BaS**	**B-->bA**
S-->bBa		**B-->ε**

8. Konstruirati Turingov stroj u osnovnom modelu koji računa umnožak dvaju cijelih brojeva. Brojevi su na traci zapisani kao nizovi znaka X, ovojeni graničnikom *. Nakon drugog broja na traci je zapisan znak =. Na početku rada glava TS nalazi se na krajnje lijevom znaku ulaznog niza. Rezultat treba zapisati na traku desno od znaka =. Dozvoljeno je da faktori budu jednaki nuli. Početno stanje trake na kraju rada mora biti očuvano. Na primjer, za ulazni niz XX*XXX= na kraju rada na traci treba pisati XX*XXX=XXXXXX

9. Konstruirati kontekstno ovisnu gramatiku koja generira proizvoljne neprazne nizove znakova a,b i c pri čemu vrijedi na!=nb, na!=nc i nb!=nc

10. Konstruirati gramatiku neograničenih produkcija koja generira jezik [(wwR)^n | w E (a+b)^*] pri čemu je n>=1 (Napomena: niz w u svim ponavljanjima označava isti niz znakova). Na primjer, nizovi abba (w=ab, n=1), abbaabba (w=ab, n=2) su u jeziku, a nizovi aba,abab,abbabaab nisu.

Sretno svima na ispitu!