Dora Maček
Vlatko Maček

PRIMJENA DIGITALNOG RAČUNALA PRI ANALIZI TEKSTA

1. UVOD

Iako su kod nas elektronska računala tek odnedavno u upotrebi u privredi i administraciji, u mnogim zemljama, a osobito u SAD, ona su već našla vrlo široku primjenu kako u spomenutim strukama tako i u znanostima, pa čak i u humanističkim. Željeli bismo ovdje navesti samo nekoliko primjera te primjene, koji su nam poznati iz literature.

U SAD je izrađena sintaktička konkordancija za srednjovisokonjemački (za koju je program sličan našem programu u zadatku 2), koja se može upotrijebiti za iscrpnu sintaktičku analizu kao i u leksikografske svrhe, budući da bi se mogao napisati program koji bi izbacio sve riječi po abecednom redu, sa oznakama za vrstu riječi i oblik, a sam tekst konkordancije, bez gramatičke analize, može se upotrijebiti u bilo koju drugu svrhu, bez obzira na upotrebu računala. Nadalje su upotrebljavani programi za izradu lingvističkog atlasa Kanade, te drugi za utvrđivanje leksičke sličnosti među dijalektima otoka Fidži. U fonologiji su se upotrebljavali programi pomoću kojih se mogu ortografski znakovi pretvarati u fonetičke, radi utvrđivanja faktora koji upravljaju konverzijom engleskih ortografskih znakova u glasove pri čitanju, a trebali bi poslužiti usavršavanju poduke u čitanju, što kod engleske kompliciranе ortografije predstavlja ne mali problem. Osim programa za fonološke, morfološke i sintaktičke analize (koji se mogu koristiti u programiranoj nastavi, strojnom prevođenju i u proučavanju lingvističkog ili stilističkog karaktera) pripremljen je i program za ekstrakciju smislom povezanih rečenica nekog teksta radi strojnog sastavljanja tzv. sažetaka naučne literature.

Kod elektronske analize teksta možemo analizirati bilo needitirani, bilo editirani tekst. Editirati tekst za elektronsku obradu znači ručno ga analizirati i unijeti sve znakove potrebne za vrstu analize koju smo upotrijebili. Npr., ako se za razna proučavanja potrebno služiti vrstama riječi nekog teksta, može se nad svaku riječ ručno unijeti oznaka

za vrstu riječi, pa će se strojno moći izdvajati ne samo pojedine leksičke jedinice nego čitave vrste riječi, odnosno kombinacije tih vrsta riječi, ili što je već potrebno na toj razini analize.

Kod needitiranog teksta preskočen je napor editiranja, no programi za analizu veoma su složeni i često ne vode brigu o izuzecima.

Program 1 u ovom tekstu je primjer takve analize (tzv. automatske analize) i može se primijeniti pri analiziranju teksta na pojedine zadane riječi ili nizove riječi (tzv. selektivna konkordancija), npr. pri katalogiziranju knjiga u biblioteci, gdje je nemoguće predvidjeti položaj ključne riječi u tekstu naslova knjige.

Editirani tekst (program 2) je prikladan, uslijed toga što je pripremljen, za upotrebu u nizu različitih, relativno jednostavnih programa za statističku analizu sintaktičkih modela. Kodiran tekst uopće je vrlo pogodan za analizu jezika koji danas postoje samo u pisanoj formi.

Fonologija i morfologija mrtvih, odnosno zastarjelih formi živih jezika veoma su mnogo proučavane, prva na problematičnim osnovama uslijed nepostojanja zvučnih dokumenata, dok je sintaksa privukla manje pažnje zbog potrebe analize golemog korpusa, što kod elektronske obrade nije problem. Njome se mogu vrlo uspješno istraživati distribucije sintaktičkih uzoraka na neizmjerno velikom tekstu.

Pri editiranju treba nadograđivati na već dosad utvrđenim podacima, te za volju brzine katkada žrtvovati i preciznost. To znači da kategorije treba odrediti dovoljno široko, no paziti da se ne izostave relevantne informacije. Kasnije se mogu ručno ili novim programom obraditi dobiveni rezultati detaljnije. Izuzeci i slično ne smiju naime, neproporcionalno otežati ili čak i onemogućiti kodiranje.

Kodira se prema određenim gramatičkim pravilima, koja treba da omoguće maksimalan broj sintaktičkih uzoraka, ali ne mogu poslužiti za stvaranje novih rečenica, jer dobiveni sljedovi samo teoretski postoje, tj. svi se ne pojavljuju u stvarnom jeziku. Oni se mogu iza prve analize filtrirati restrikcijama na gramatiku, i baš tom analizom može se dobiti baza za pronalaženje elemenata pomoću kojih se gramatika može detaljnije obraditi, pročistiti.

U slučaju da su neke rečenice prekomplicrane i ne mogu se kodirati prema predviđenim kategorijama, bolje ih je ručno analizirati, nego dodavati nove kategorije.

Jednom kodirani tekst prikladan je za desetine analiza bez mijenjanja. Kao što je spomenuto, ovakvo kodiranje omogućava analizu bilo kakvog pisanog teksta, naročito ako se radi o proučavanju stila nekog pisca, frekvencije riječi, odnosno fraza. Moguće su i razne grafemičke analize, no one zahtijevaju drugačije kodiranje teksta nego što je u ovom radu prikazano.

Da bi se odgovorilo na pitanja sintaktičke naravi, iskustvo je pokazalo da kodirani korpus treba sadržavati barem 100.000 redaka poezije, što zahtijeva oko 2.000 sati kodiranja. Ta početna investicija je ipak opravdana kasnijim neograničenim mogućnostima analize na višem nivou.

2. SELEKTIVNA KONKORDANCIJA CHAUCEROVIH KANTEBERISKIH PRIČA

2.1. ZADATAK[1]

Ispitati needitirani tekst na modalne glagole *shall, will, can* i *may*. U tabeli programa su zadani svi oblici (ortografski i morfološki) ovih glagola, verificirani u srednjoengleskim tekstovima. Rezultat analize je selektivna konkordancija, koja će izbaciti sve retke (može se zahtijevati i da izbaci sve rečenice) u kojima se pojavio neki od četiri spomenuta glagola, u bilo kojem obliku. Takva konkordancija treba da posluži ispitivanju glagolskih fraza kojima je sastavni dio neki od zadanih glagola, njihove sintaktičke funkcije i njihova semantičkog polja. Budući da se pretpostavlja da se u ove dvije stavke glagoli *shall, will, can* i *may* razlikuju od istih glagola u suvremenom engleskom jeziku, kasnije se može napraviti konkordancija prijevoda istih tekstova na suvremeni engleski, po potrebi izlučiti sve zanimljive retke originala i prijevoda, te usporediti sintaktičko i semantičko ponašanje zadanih glagola u oba jezična sistema.

Radi dobivanja gore spomenutih konkordancija može se upotrijebiti niže prikazani program pisan strojnim jezikom »Assembler«. Za dalje analize tako dobivenog materijala mogu se upotrijebiti drugi programi na istom, ili drugim jezicima.

2.2. ULAZ

Kao ulazni podaci služi tekst Kanteberiskih priča ubušen na IBM kartice. Prvih 60 kolona svake kartice zauzima redak teksta, a kolone 72–80 oznaku retka unutar teksta. Prije izvršenja programa tekst je prenesen s kartica na disk.

2.3. IZLAZ

Na printeru (štampaču) traži se lista, koja prikazuje sve stihove u kojima se pojavljuju ispitivani glagoli, s oznakom retka u kojem se nalaze. Glagole treba staviti u 60. kolonu na listi i štampati s kontekstom.

Primjer za *wol*

TEKST	REDAK
I *wol* bothe drinke and eten of a cake	37
Som wit and thanne *wol* we gradly heere	41
itd.	

[1] Zadatak je postavljen radi pripreme materijala za radnju D. Maček, kojoj je svrha da prouči sintaktičke i semantičke funkcije spomenutih glagola u Chaucerovu jeziku, što do sada još nije izvršeno.

2.4. OPIS DIJAGRAMA TOKA PROGRAMA 1

(1) Prvo očistimo izlazno područje,[2] koje smo nazvali RED, a koje ima 130 bajtova.[3] U registar 6 stavimo adresu[4] SLOGA, tj. ulaznog područja,[5] dužine 80 bajtova.
U R7 dolazi adresa radnog područja,[6] tj. PROBE, dužine 16 bajtova. U R8 dolazi adresa TABELE sa ključnim riječima. dužine n $\times$ 16 bajtova. U R9 dolazi adresa lokacije za uskladištenje[7] rečenica, koje sadrže istu ključnu riječ, tj. POPISA, dužine 100 $\times$ 82 bajta.

(2) Unesemo u memoriju prvi bajt izlaznog SLOGA.

(2a) Kraj datoteke? DA, granaj na KRAJ. NE, idi na (3).

(3) Da li je praznina (blenk)? Ako DA, idemo na NVR (nova riječ), jer je 1 praznina dogovorena oznaka za razmak između riječi.

(4) Ako NE, prepišemo to slovo u prvi bajt PROBE. Povećamo adresu SLOGA za 1 i adresu PROBE za 1.

(5) U brojač B1 dodamo 1. B1 će brojiti ukupan iznos bajtova ispitivanog retka. U brojač B2 dodamo 1, on će brojiti samo dužine pojedinih riječi.

(6) Granamo na OPET. Ta petlja se ponavlja sve dok ne unesemo u memoriju čitavu riječ, tj. ne dođemo do praznine. Tada idemo na (7).

(7) NOVA RIJEČ.
Usporedimo riječ u PROBI s prvom ključnom riječi u TABELI. Ako nisu iste, idemo na ISP(itivanje).

(8) Ako su iste, prepišemo čitavih 80 bajtova SLOGA u POPIS. Računamo pomak ključne riječi: oduzmemo iznos u R2 od iznosa u R1 i rezultat R stavimo u POPIS. Dodamo na kraj POPISA 999.

(9) Očistimo B1, B2 i PROBU.

(10) Povećamo adresu POPISA za 82.

[2] Izlazno područje je dio glavne memorije u kojem ćemo pomoću programa sastaviti tekst koji tražimo i zatim ga ispisati na printeru (ovdje RED).

[3] Bajt – dio memorije, dovoljan da se memorira jedno slovo ili posebni znak. Fizički se sastoji od 8 u nizu poredanih magnetskih jezgrica.

[4] Adresa – oznaka nekog mjesta u memoriji, na kojem počinje polje (jedan ili više bajtova) za memoriranje teksta.

[5] Ulazno područje – dio glavne memorije u koji unosimo svaki redak teksta (ovdje SLOG).

[6] Radno područje – dio glavne memorije u kojem ispitujemo svaki redak teksta (bajt po bajt), (ovdje PROBA).

[7] Lokacija za uskladištenje – dio glavne memorije gdje spremamo sve retke u kojima se nalazi zadana glagolska forma, sve dok ne proanaliziramo čitavi korpus (ovdje POPIS).

(11) Granamo na ČIT. To je mjesto gdje u memoriju unesemo novu logičnu snimku (redak) s diska, jer smo u prethodnoj našli ključnu riječ i pohranili s izračunatim pomakom u POPIS.

(12) ISP(itivanje).

(13) Očisti B2 i PROBU.

(14) Da li su prva dva bajta na novoj adresi SLOGA praznine? Ako NE, idi na (15). Ako DA, idi na (17).

(15) Povećaj adresu SLOGA za 1. Dodaj 1 u B1. Prebaci početnu adresu PROBE u REG7.

(16) Granaj na OPET. Tu započinje ciklus prebacivanja slijedeće riječi u PROBU.

(17) Ako imamo dvije praznine, znači da je to kraj logičke snimke, (retka). Očistimo B1 i idemo na (11).

(18) KRAJ.

(19) Povećamo broj stranice za 1. Stavimo početnu adresu POPISA u REG9.

(20) Stavimo adresu REDA u REG10. Stavimo iznos 60 u POLR. 60 bajtova je dužina promatrane logičke snimke. Oduzmemo od 60 iznos R, koji se nalazi pohranjen u POPISU. Dobiveni iznos je pomak za koji ćemo početi pisati sadržaj te logičke snimke da bi nam ključna riječ ušla u kolonu 60.

(21) Da li je iznos u POLR = 0? Ako NE, idemo na (22). Ako DA, idemo na (24).

(22) Povećamo adresu REDA za 1, a iznos u POLR smanjujemo za 1.

(23) Granamo na JOŠ. To se ponavlja tako dugo dok nova adresa REDA ne dobije pomak za početni iznos u POLR.

(24) DOLJE.

(25) Prepišemo sadržaj iz POPISA u RED, te šifru (zadnjih 8 bajtova iz POPISA) u zadnjih 8 bajtova REDA.

(26) Štampano RED.

(27) Brišemo RED. Povećavamo adresu POPISA za 82.

(28) Da li je nova snimka u POPISU = 999? Tj. da li je to kraj POPISA?

(29) Ako NE, granaj na GORE, tj. izračunaj pomak novog retka, modificiraj adresu i štampaj ga. Ako DA, znači nema više snimaka koje su selektirane po toj ključnoj riječi, idi na novu ključnu riječ, tj. na (30).

(30) Povećaj adresu TABELE za 16.

(31) Da li je nova riječ 999? Ako DA, idi na (32). Ako NE, idi na (33).

(32) STOP.

(33) N(ovi) KLJUČ.

(34) Stavi početnu adresu POPISA u REG9. Stavi iznos 100 u REG10. Naime, pretpostavili smo da će se ključna riječ pojaviti u najviše 100 ulaznih snimaka.

(35) Brišemo POPIS (po 82 bajta).

(36) Iznos u REG10 smanjimo za 1. Da li je iznos u REG10 sada 0? Ako NE, idemo na (35).

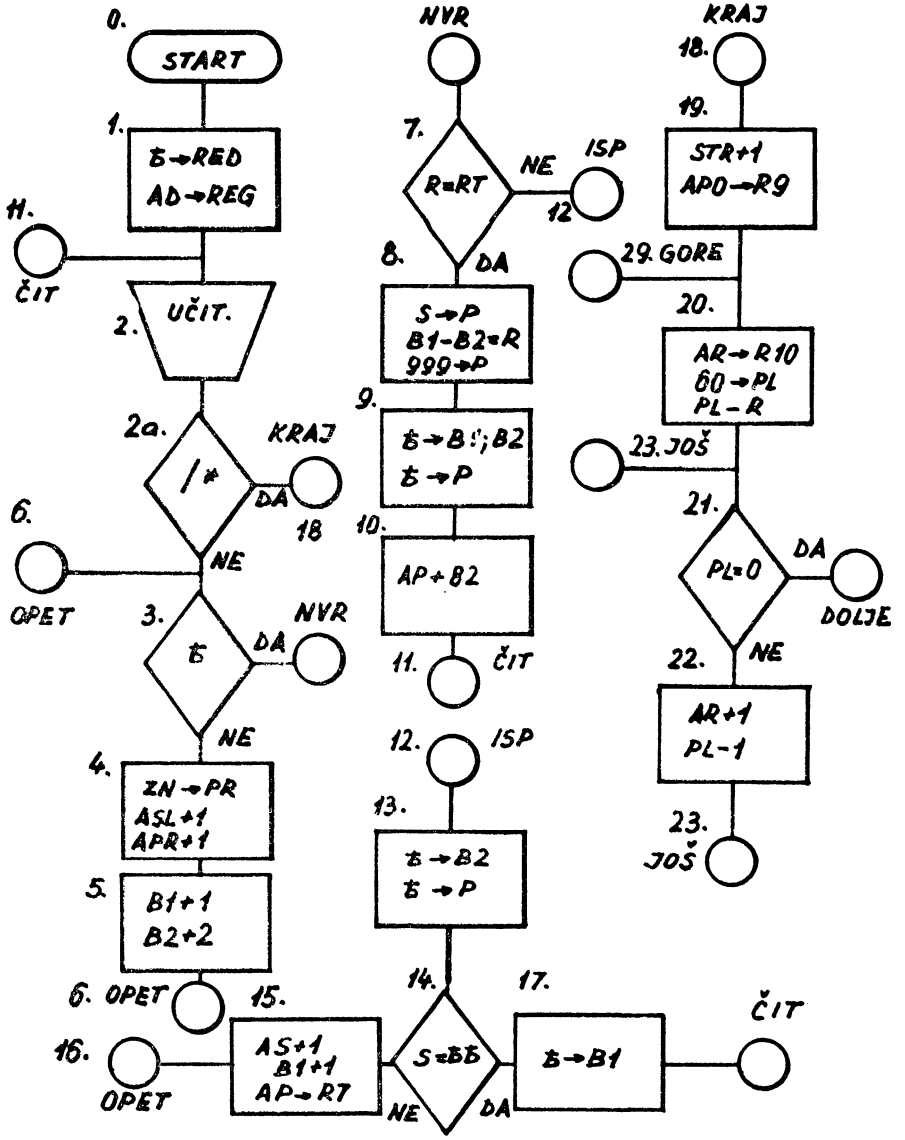
(37) Ako DA, stavimo početnu adresu POPISA u REG9 i idemo na (11), gdje u memoriju unesemo slijedeću logičku snimku. Tu počinje petlja analiziranja teksta po novoj ključnoj riječi.

TABELA I.

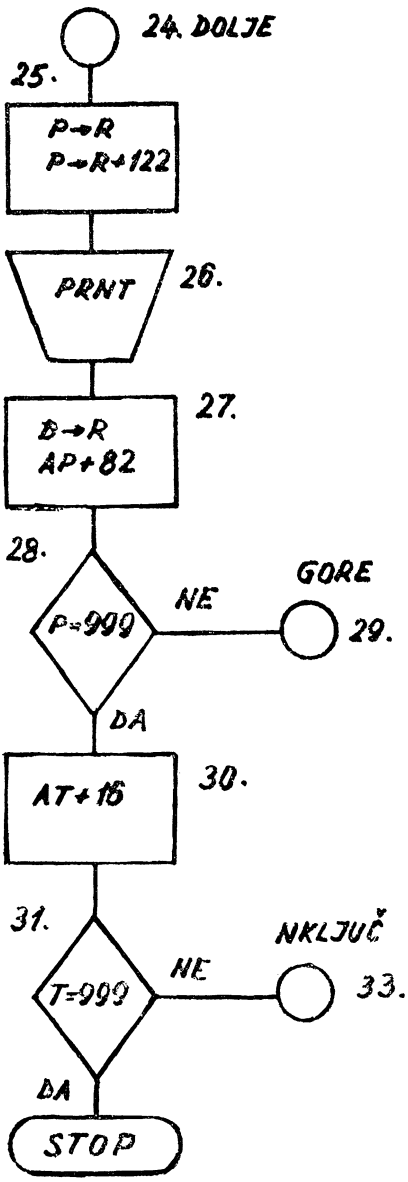
POPIS KLJUČNIH RIJEČI ZA KONKORDANCIJU CHAUCEROVIH TEKSTOVA

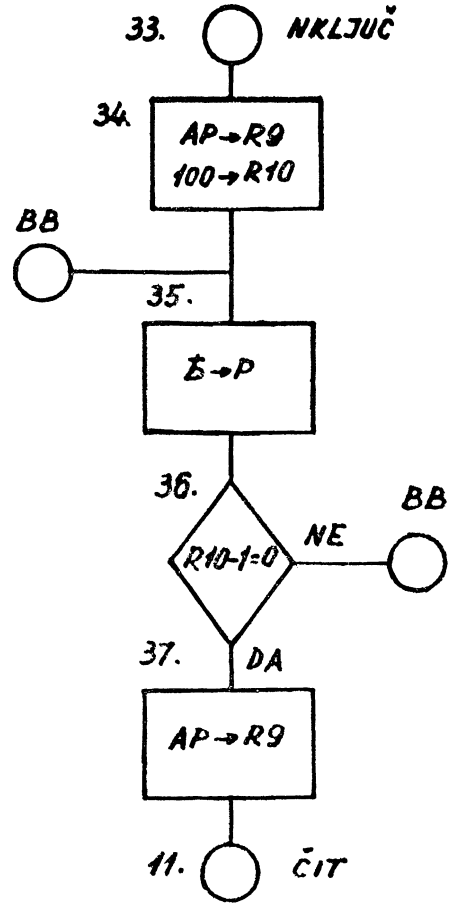
SHAL	WIL	CAN	MAI
SHALL	WILL	CANN	MAY
SHALST	WILST	CANNST	MAIST
SHALT	WILT	CANSTOW	MAYST
SHALLT	WILLT		MIGHT
SHALE	WILE		MIGHTE
SHALLE	WILLE		MIGHTEN
SHOLD	WOLD		MAISTOW
SHOLDE	WOLDE		MAYSTOW
SHALEN	WOLE		
SHALLEN	NIL		
SHALSTOW	NILL		
SHALLSTOW	NILE		
	NILLE		
	NILEN		
	NILLEN		
	NILETH		
	NILLETH		
	NOLE		
	NOLLE		
	NOLEN		
	NOLLEN		
	NOLETH		
	NOLLETH		
	NOL		
	NOLL		
	NOLD		
	WILTOW		

DIJAGRAM TOKA ZA PROGRAM 1.



PROGRAM 1 - NASTAVAK





3. SINTAKTIČKA ANALIZA MATOŠEVA TEKSTA

3.1. ZADATAK

U editiranom Matoševu tekstu, prema priloženoj tabeli, treba izbrojiti rečenice bez subjekta (BSUB) i rečenice sa subjektom (SUB). Kod toga nećemo brojiti rečenice koje se sastoje samo od jedne riječi, rečenice s glagolom u imperativu, te rečenice bez glagola.

3.2. ULAZ

Editirani tekst, kao što to prikazuje tabela, prepisan je sa kartica na disk. Jedan redak tabele je jedna editirana riječ, tj. riječ + kodovi gramatičkih kategorija.

3.3. IZLAZ

Ispisivanje na printeru iznosa BSUB i SUB.

3.4. OPIS DIJAGRAMA TOKA PROGRAMA 2

(1) Očistimo registre u kojima ćemo sumirati BSUB i SUB iznose.

(2) Unesemo u memoriju prvu editiranu riječ korpusa.

(3) Ispitujemo da li je kolona 3 praznina? Ako DA, idemo na IMP(erativ).

(4) Ako NE, dodajemo + 1 u SUB.

(5) Unesemo u memoriju slijedeću riječ iz korpusa.

(6) Ispitujemo da li je to riječ ili smo došli do kraja korpusa. Ako je to riječ, idemo na (8).

(7) Ako nema više riječi, ispisujemo na printeru sume u registrima BSUB i SUB.

(8) Ispitujemo da li je u koloni 2 slovo P. Ako DA, idemo na POČ(etak). Ako NE, idemo na SLR.

(9) STOP, završetak analize.
IMP.

(10) Ispitujemo da li je u koloni 7 oznaka IMP. Ako DA, ignoriramo tu riječ i idemo na (12). Ako NE, idemo na (14).

(12) Unesemo u memoriju iduću riječ.

(13) Ako je to nova riječ, idemo na POČ. Ako nema više riječi u korpusu, idemo na PIŠI, tj. na 7.

(14) Unesemo u memoriju iduću riječ.

(15) Ispitamo da li je to riječ iz korpusa. Ako DA, idemo na POL(ožaj). Ako ne, idemo na PIŠI.
POL.

(16) Ispitamo da li je to prva riječ u rečenici, tj. da li je u koloni 2 oznaka P.

(17) Ako DA, dodamo 1 u BSUB i idemo na POČ.

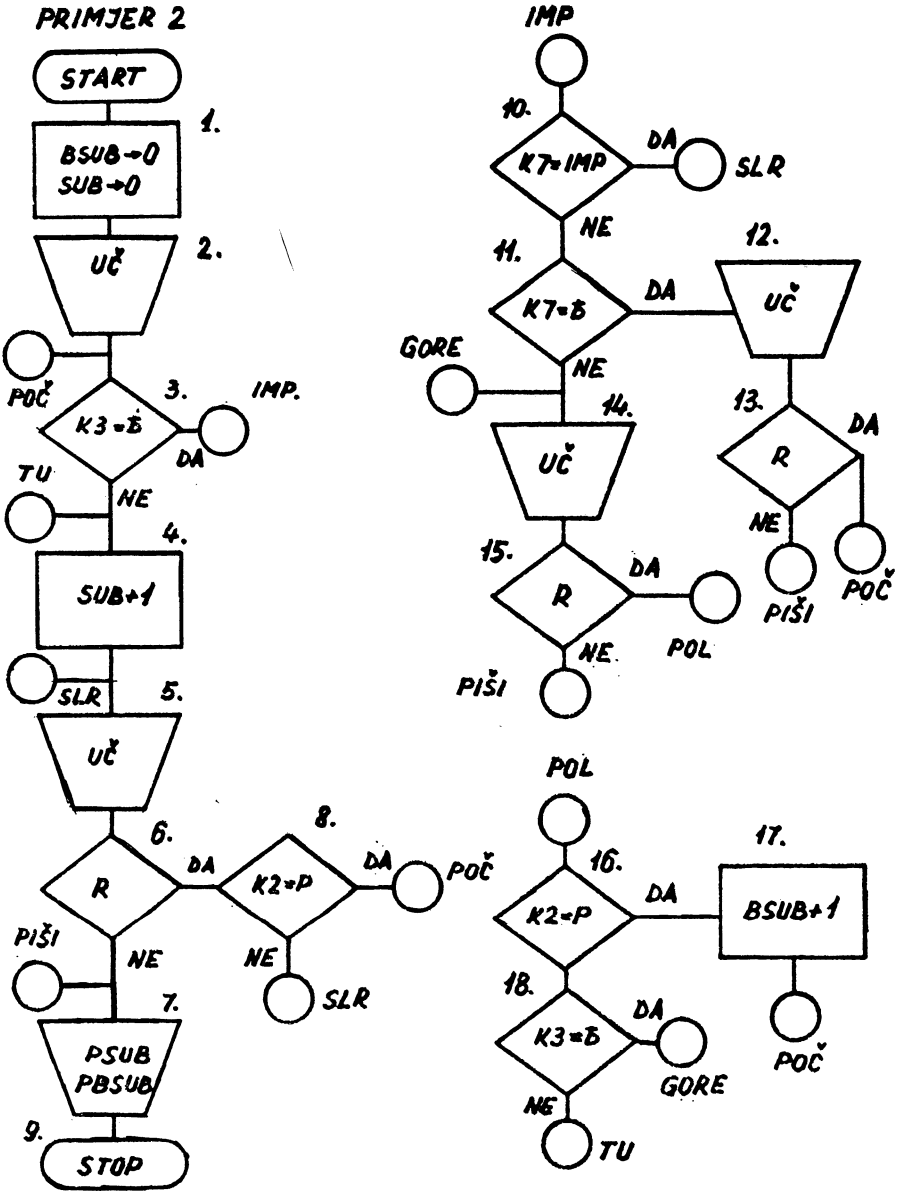
(18) Ako NE, ispitamo da li je u koloni 3 praznina? Ako DA. idemo na GORE. Ako NE, idemo na TU, tj. dodavanje 1 u registar za rečenice sa subjektom.

* * *

Kako se vidi iz prethodnog, operacije koje vrši stroj vrlo su jednostavne i postepene, za svaki pomak u programu uputa mora točno označiti što da se radi, a ako postoji izbor, on mora biti ograničen na postojanje nekog elementa ili na njegovu odsutnost. Važno je međutim to da se sve operacije vrše velikom brzinom, koja se ne može usporediti s vremenom koje bi za isti posao morala utrošiti nekolicina ljudi. Što stroj uradi za nekoliko sati ili minuta, istraživač bi uz nekolicinu pomoćnika morao raditi nekoliko godina ili barem nekoliko mjeseci. Poslove oko obrade teksta, te raznih statističkih analiza, prebrojavanja i sl., koji su monotoni za čovjeka, stroj obradi mnogo brže i, što nije manje važno, mnogo preciznije.

TABELA II

Red	Tekst	Gramatičke kategorije							
		1	2	3	4	5	6	7	8
1	Ponoć	G	P	R		Nom	Sing		Im
	već	G	S						Pril
	je	G	S		P		Sing	Sad	Gl
	prošla	G	K	I	K		Sing	Pp	Gl
	svjetlo	G	P	R		Nom	Sing		Im
	mi	G	S			Dat	Sing		Z
	se	G	S		P	Ak	Sing		Z
	gasi	G	K		K		Sing	Sad	Gl
2	Na	G	P		P				Pdl
	baršunu	G	S		S	Lok	Sing		Im
	crnom	G	S		S	Lok	Sing		Pr
	leži	G	S		K			Sad	Gl
	teška	G	S	P		Nom	Sing		Pr
	noć	G	K	K		Nom	Sing		Im


PRIMJER 2
START
1.
BSUB←0
SUB←0
2.
UČ
POČ
3.
K3=B
IMP.
DA
TU
NE
4.
SUB+1
SLR
5.
UČ
6.
R
DA
8.
K2=P
DA
POČ
PIŠI
NE
NE
SLR
7.
PSUB
PBSUB
9.
STOP
IMP
10.
K7=IMP
DA
SLR
NE
11.
K7=B
DA
12.
UČ
GORE
NE
14.
UČ
13.
R
DA
NE
PIŠI
POČ
15.
R
DA
POL
NE
PIŠI
POL
16.
K2=P
DA
17.
BSUB+1
18.
K3=B
DA
GORE
POČ
NE
TU

Gramatička pravila za editiranje teksta

1. Vrsta rečenice: *G*lavna, *A*dverbna, itd.
2. Poredak riječi unutar rečenice: *P*očetak, *S*redina, *K*raj
3. Poredak riječi unutar imeničke fraze: *P*očetak, *S*redina, *K*raj, *R*iječ, *N*išta
4. Poredak unutar glagolske fraze: isto kao pod 3.
5. Padež: *Nom*inativ, *Gen*itiv, *Dat*iv, *Ak*uzativ, *Vok*ativ, *Lok*ativ, *Instr*umental
6. Broj: *Sing*ular, *Plur*al, *Du*al, *Neod*ređeno, Ništa
7. Glagolski oblik: *Sada*šnjost, *Pro*šlost, *P*articip sadašnji, *P*articip prošli, *Inf*initiv, *Imp*erativ, Ništa
8. Vrsta riječi: *Im*enica, *Z*amjenica, *Gl*agol, *Pr*idjev, *Pril*og, *V*eznik, *P*rije*dlo*g, itd.