RAČUNARSTVO
Sveska 1, Broj 2, 1992

UDK 519.681

FORMALNE GRAMATIKE[1]

MARICA D. PREŠIĆ

Razvitak lingvisike, tu prvenstveno mislimo na tzv. opštu lingvistiku koja se bavi opštim jezičkim istraživanjima zajedničkim za sve jezike, poklapa se u glavnim crtama sa razvitkom (matematičke) logike: burno u antičko doba, sa visokim i vekovima neprevaziđenim dometima, lagano i veoma usporeno sve do XIX veka; u tom su periodu uglavnom razrađivane antičke ideje. XIX vek, međutim, nije bio vek opšte lingvistike, već vek posebnih lingvističkih istraživanja u koja spadaju komparativne gramatike, biološki naturalizam, "humboltizam", psihologizam i dr.

Bura nastaje ponovo u XX veku: nagli razvitak matematičke logike bio je priprema i osnova naglog razvitka opšte lingvistike, koji počinje 60tih godina [Istorijskim datumom se smatra objavljivanje knjige "Syntactic Structures" Noama Čomskog (Chomsky) godine 1957]. Oslanjajući se na tradicionalno gledište racionalističkih filozofa sedamnaestog i osamnaestog veka po kome "strukturu jezika određuje struktura ljudskog uma i da univerzalnost izvesnih osobina svojstvenih jeziku svedoči da je bar taj deo ljudske prirode zajednički svim članovima vrste", a budući da je na univerzitetu dobio solidno znanje kako iz lingvistike, tako i iz filozofije i matematike (posebno se to odnosi na fundamentaltalna dostignuća u oblastima rekurzivnih funkcija i teorije algoritama), Čomski, nastavljajući ideje svog učitelja, "blumfildovca" Zeliga Harisa, postavlja sebi za cilj zasnivanje sintakse u obliku takozvane formalne teorije.

Napomenimo odmah, da je semantika bila kod Čomskog potpuno zanemarena u prvim istraživanjima [što je donekle kasnije izmenjeno], nalazila se u podređenom položaju u odnosu na sintaksu i nije, po mišljenju Čomskog, u pravom smislu pripadala lingvistici.

Ukratko o formalnoj teoriji: Pojam, u današnjem obliku, dugujemo Hilbertu, a koreni su mu u pojmu aksiomatske teorije, koji potice iz antičkog doba [Euklid je za to najveći zaslužnik]. Formalnu teoriju određuju izvesni polazni znaci koji čine

[1]Rad finansiran od Fonda za nauku Republike Srbije preko Matematičkog instituta, projekat 0401A.

takozvani alfabet, skup izvesnih reči nad tim alfabetom koje se nazivaju formulama [za koje se pretpostavlja jedno potpuno određeno pravilo prepoznavanja], skup aksioma [to su neke formule izdvojene kao polazne], i najzad pravila izvođenja koja su oblika:

(1) Iz formula F_1, F_2, ..., F_n se izvodi formula F

Pravila izvođenja su, ustvari, vrsta relacija među formulama. Formalne teorije su pravljene radi strogog zasnivanja najvažnijeg matematičkog pojma: teoreme. Aksiome su, ustvari, polazne teoreme, dok se sve druge druge teoreme dokazuju na osnovu aksioma primenom pravila izvođenja. Sam dokaz je jedan konačan niz formula koje su ili aksiome ili slede iz nekih prethodnih formula tog niza primenom izvesnog pravila izvođenja. U opštem slučaju dokazi teorema mogu biti veoma dugački i komplikovani, teorema može imati više različitih dokaza, a za datu formulu nije uvek lako, a najčeće je i nemoguće, odgovoriti da li ona jeste ili nije teorema neke formalne teorije. Međutim, za neke posebne vrste formalnih teorija rešavanje takvih pitanja je moguće i za to su u četvrtoj i petoj deceniji ovog veka razvijeni moćni logički aparati. Tu se prvenstveno misli na oblasti koje su se razvile u vezi sa naporima da se strogo zasnuje pojam algoritma (postupka), a to su: teorija rekurzivnih funkcija, Tjuringove mašine, kombinatorni sistemi, teorija algoritama. Mada su sve te teorije u određenom smislu međusobno ekvivalentne, Čomski se opredelio za kombinatorne sisteme [koje je Post razvio 1936. godine], budući da su oni bili najbliži idejama koje je imao u vezi sa strogim logičkim zasnivanjem sintakse prirodnih jezika.

Ukratko o kombinatornim sistemima: To je posebna vrsta formalne teorije nad konačnim alfabetom kod koje su sve reči (nad uočenim alfabetom) formule, konačno mnogo reči je uzeto za aksiome, dok su pravila izvođenja, a njih je takođe konačno mnogo, sva posebnog oblika:

(2) $$A_0X_1A_1X_2\ldots,X_rA_r \longrightarrow B_0X_{i_1}B_1X_{i_2}\ldots,X_{i_s}B_s$$

Tu su $A_0, A_1, \ldots, A_r, B_0, B_1, \ldots, B_s$ date, fiksirane reči, dok su $X_0, X_1, \ldots, X_r$ proizvoljne reči ($X_{i_1}, X_{i_2}, \ldots, X_{i_s}$ su neke od $X_1, X_2, \ldots, X_r$).

[Pravilo (2) je primer tzv. shema-pravila; budući da su $X_1, X_2, \ldots, X_r$ proizvoljne reči nad uočenim alfabetom, a njih je beskonačno mnogo, to je (2) zajednički zapis za beskonačno mnogo posebnih pravila izvođenja, otkuda i naziv shema-pravilo.]

Pravila prethodne vrste se, inače, nazivaju zamenama (ili produkcijama). [Umesto reči "Iz ... se izvodi ... " stoji strelica $\longrightarrow$ a samo pravilo $X \longrightarrow Y$ se čita "X se zamenjuje sa Y"]. Čest je slučaj da je polazni alfabet A kombinatornog sistema podeljen na dva međusobno razdvojena podskupa T (završni, terminalni) i N (pomoćni, neterminalni), pri čemu pomoćni simboli igraju, ustvari, ulogu promenljivih. Evo nekoliko jednostavnih primera kombinatornih sistema:

PRIMER 1.. *Alfabet je* $\{a,b\}$, *aksiome su* $\#$, a, b ($\#$ *je oznaka za praznu reč), zamene su shema-zamene:*

$$X \longrightarrow aXa, \quad X \longrightarrow bXb \qquad (X \text{ je proizvoljna reč})$$

Neke teoreme su na primer:

$\#$, a, b, aa, bb, aba, bab, $abba$, $baba$, $ababa$, $abbba$, $babab$, $baaab$, $bbabb$, $abbabba$.

Nije teško dokazati (indukcijom po dužini dokaza) da su sve teoreme simetrične reči (takozvani palindromi).

PRIMER 2.. *Alfabet je* $\{a,b\}$, *aksiome su* ab, $abab$, $ababab$, $abababab$, *zamena je:* $aXbY \longrightarrow XbY$ *(*X, Y *su ma koje reči, dakle ponovo shema-zamena). Teoreme su upravo ove reči:* ba, $baba$, $bababa$, $babababa$. *Recimo, teorema* $bababa$ *ima tri dokaza i svi su oni dužine dva: U prvom imamo* $X = \#$, $Y = abab$, *jednom primenom zamene neposredno dobijamo* $bababa$. *U drugom dokazu* $X = ba$, $Y = ab$, *a u trećem* $X = baba$, $Y = \#$ *i opet je dovoljno, u oba slučaja, primeniti zamenu samo jedanput. Očigledno je takođe da se iz dobijene teoreme* $bababa$ *ne može izvesti nijedna nova, budući da ta reč počinje sa* b. *Što se tiče teorema* ba, $baba$, $babababa$, *one imaju po redu: jedan, dva, četiri dokaza, a iz razloga sličnog onom maločas iz njih se ne može izvesti nijedna nova teorema pa je ukupnost svih teorema u ovom primeru upravo:* ba, $baba$, $bababa$, $babababa$.

Navedeni primer se može uopštiti uzimanjem za aksiome ma koliko (ali konačno mnogo) reči oblika $(ab)^n$ [X^n je uobičajena oznaka za stepen reči X dobijen dopisivanjem X n puta. Tu je n ma koji dati prirodan broj uključujući i nulu, pri čemu se X^0 definiše kao $\#$.] Teoreme će tada biti sve odgovarajuće reči oblika $(ab)^n$ a svaka takva teorema će imati ukupno n dokaza (dužine dva).

PRIMER 3.. *Alfabet je* $\{a,b,S,(,)\}$, *završni simboli su* a, b, $($, $)$, *pomoćni simbol je* S, *aksioma je takođe* S, *a zamene su:*

$$XSY \longrightarrow X(S*S)Y, \qquad XSY \longrightarrow XaY, \qquad XSY \longrightarrow XbY$$
$$(X, Y \text{ su ma koje reči})$$

Neke teoreme su, na primer:

$$S,\ (S*S),\ (S*(S*S)),\ ((S*S)*(S*(S*S))),$$
$$(S*(a*b)),\ ((a*S)*(b*(S*a))),$$
$$a,\ b,\ (a*b),\ (b*a),\ (a*(a*b)),\ (b*(b*b)),\ ((a*b)*(b*(a*b)))$$

Teoreme u trećem redu su posebne vrste, takozvane završne, budući da u njima učestvuju jedino završni simboli. Kao što se vidi to su ustvari izrazi obrazovani od a, b, $*$ ($*$ je operacijski znak dužine dva). Indukcijom se dokazuje da su završne teoreme upravo svi takvi izrazi.

Kombinatorni sistem u prethodnom primeru je posebne vrste. Sve njegove zamene su oblika:

$$(3) \qquad XAY \longrightarrow XBY \qquad (X, Y \text{ su proizvoljne reči})$$

To je primer takozvanog polutuovskog sistema, pri čemu se zamene (3) obično zapisuju bez proizvoljnih reči X, Y, tj. u obliku:

$$(4) \qquad A \longrightarrow B$$

i podrazumeva se da **primena** takve zamene znači sledeće: U proizvoljnoj reči W (nad uočenim alfabetom) podreč A, i to bilo koje izabrano njeno pojavljivanje u W, zamenjuje se sa B. Tako, u prethodnom primeru teorema $(S*(S*S))$ se dokazuje na osnovu već dokazane teoreme $(S*S)$ zamenom drugog pojavljivanja slova S sa $(S*S)$, tj. primenom zamene $XSY \longrightarrow X(S*S)Y$, ili drugačije: primenom zamene $S \longrightarrow (S*S)$, ukoliko koristimo skraćeno pisanje (4) [X, Y su po redu: $(S*$, odnosno $))$].

PRIMER 4. *Polutuovski sistem je određen uslovima: Završni alfabet je* $\{a, b, c\}$, *pomoćni je* $\{S, T\}$, *aksioma je* S, *zamene su:* $S \longrightarrow aT$, $S \longrightarrow c$, $T \longrightarrow Sb$. *Uzastopnom primenom prve i treće zamene izvode se redom teoreme:*

$$S,\ aT,\ aSb,\ aaTb,\ aaSb,\ aaaTbb,$$
$$aaaaSbbb,\ aaaaTbbb,\ aaaaSbbbb,\ aaaaaTbbbb, \ldots$$

iz kojih, primenom pravila $S \longrightarrow c$ *slede završne teoreme:*

$$c,\ acb,\ aacbb,\ aaacbbb,\ aaaacbbbb, \qquad \text{odnosno uopšte}$$
$$a^n c b^n \qquad (n = 0, 1, 2, 3, \ldots)$$

Najzad još jedan polutuovski sistem čije se teoreme ne opisuju tako očigledno kao što je to bio slučaj u prethodnim primerima.

PRIMER 5.. *Završni alfabet je* $\{a, b, c\}$, *aksioma je* S *(to je ujedno i jedini pomoćni simbol), zamene su:* $S \longrightarrow aSb$, $S \longrightarrow c$, $aSb \longrightarrow bSa$.

Polutuovski sistemi u trećem i četvrtom primeru bili su posebne vrste, koju preciziramo narednom definicijom.

DEFINICIJA 1. *Polutuovski sistem čiji je alfabet A pode-ljen na razdvojene podskupove T, N (završnih i pomoćnih simbola) i koji ima samo jednu aksiomu, obično je to S, je takozvana (formalna) gramatika ili gramatika Čomskog. Skup svih završnih teorema čini takozvani jezik generisan formalnom gramatikom.*

[Prethodna definicija gramatike je generativna, što znači da se sve njene teoreme izvode, generišu iz jedne jedine aksiome S. Postoji i drugačiji pristup

gramatikama, dualan prethodnom, tzv. analitički. Neka reč je teorema analitičke gramatike ukoliko se iz nje (iz te reči) može izvesti aksioma S.]

Najznačajnija vrsta formalnih gramatika su kontekstno slobodne gramatike (što predstavlja nakaradan ali odomaćen prevod engleskog naziva context-free grammars), kod kojih su sve zamene oblika:

(5) $N \longrightarrow B$ (N je pomoćni simbol, B je reč iz A^*)

[A^* **je uobičajena oznaka za skup svih reči nad alfabetom** A, **uključujući i praznu reč** $\#$.]

Napomenimo samo da pored kontekstno slobodnih gramatika postoji još nekoliko drugih vrsta gramatika (podelu je načinio sam Čomski), kao što su kontekstno osetljive, čije su zamene oblika:

(6) $CND \longrightarrow CBD$ (N je pomoćni simbol, C, B, D su iz A^*, B nije $\#$)

zatim regularne gramatike, normalne i druge. Primenu u prirodnim jezicima posebno su našle kontekstno slobodne gramatike.

[Pitanje da li su prirodni jezici kontekstno slobodni ili nisu još uvek je otvoreno. Sam Čomski je na početku svojih istraživanja izjavljivao da ne zna odgovor a kasnije je počeo da zastupa stanovište da je odgovor negativan, što je čitav niz godina postalo široko prihvaćeno. Poslednjih godina, međutim, to gledište je bitno poljuljano, jer se ispostavilo da su svi do sada objavljeni dokazi o kontekstnoj osetljivosti prirodnih jezika uglavnom vrlo diskutabilni a često i pogrešni. Kako sada stvari stoje još uvek nema valjanog dokaza ni da su prirodni jezici kontekstno slobodni niti da su kontekstno osetljivi.]

Pre nego što pređemo na detaljije izlaganje o primeni gramatika Čomskog na prirodne jezike, vratimo se ponovo, nakratko, na opšti slučaj kombinatornih sistema. U čemu je njihova važnost i značaj u matematici? Već smo pominjali teoriju rekurzivnih funkcija i teoriju algoritama. U istu grupu teorija spadaju i Tjuringove mašine, URM-mašine (unlimited register machine), kombinatorni sistemi. Sve su se te teorije razvile u vezi sa jednin jedinim ciljem: da se strogo, matematički zasnuje intuitivni pojam algoritma (postupka).

[Pod tim pojmom se podrazumevaju kako raznorazni tehnološki postupci, tako i , na primer, postupak kucanja teksta pisaćom mašinom, kao i matematički postupci množenja dva prirodna broja ili određivanja njihovog najvećeg zajedničkog delitelja. Glavna karakteristika svakog takvog postupka jeste da se on obavlja u konačno koraka i da se prelazak sa jednog na drugi korak obavlja po tačno utvrđenim pravilima—uputstvima, kojih takođe ima konačno mnogo.]

U svakoj od tih teorija centralni pojmovi su izračunljivost (funkcija je izračunljiva ukoliko postoji postupak kojim se za dati original, u oblasti definisanosti, određuje njegova slika), nabrojivost (skup je nabrojiv ukoliko postoji postupak nabrajanja njegovih elemenata), odlučivost (problem je odlučiv ukoliko postoji postupak koji u svakom konkretnom slučaju problema daje odgovor da ili ne). Pokazalo se da su pristupi svakom od tih pojmova, razvijani uporedo u pomenutim teorijama, svi međusobno ekvivalentni, što predstavlja jednu od potvrda Čerčove teze po kojoj:

Izračunljivost, definisana na ma koji od međusobno ekvivalentnih načina, pokriva intuitivni pojam algoritma.

Pomenimo još da se teorijski razvitak pojma algoritma poklapa sa tehničkim dostignućima u oblasti elektronskih računskih mašina. Razlog za to je veoma prirodan: mašina može obavljati samo one radnje za koje postoji algoritmi.

Kontekstno slobodne gramatike, kojima se sada vraćamo, karakteriše posebna jednostavnost u postupku dokazivanja teorema. Budući da svaka takva gramatika ima samo jednu aksiomu, S na primer, to svaki njen dokaz počinje sa S. Dalje, sve njene zamene su oblika (5) čijom primenom se dužina teorema ne smanjuje (jer se slovo, tj. pomoćni znak N zamenjuje rečju B čija dužina je veća ili jednaka 1). Kako je svaka reč konačna, to u svakoj već dokazanoj teoremi može učestvovati najviše konačno mnogo pomoćnih znakova. Otuda se iz svake teoreme može izvesti najviše konačno mnogo novih teorema, a postupak njihovog izvođenja je potpuno određen: svaka primena po jedne zamene daje po jednu novu teoremu. Pored toga, ako je A data reč, onda postoji postupak utvrđivanja da li ona jeste ili nije teorema. Dosta je, naime, uočiti sve reči (nad alfabetom razmatrane gramatike) čija je dužina manja ili jednaka dužini reči A (a njih je konačno na broju budući da je alfabet konačan), i najzad među dokazima koji "idu" preko takvih reči (a i njih je konačno mnogo) pronaći jedan koji se završava sa A. Ukoliko takav dokaz postoji A jeste teorema, a ukoliko ne postoji A nije teorema. U skladu sa uobičajenom terminologijom napred izložene karakteristike pojma "teoremnosti" se iskazuju kratko: Skup teorema kontekstno slobodne gramatike je rekurzivan, a svojstvo "biti teorema" je odlučivo. To je razlog što su te gramatike podesne za rad na računskim mašinama. Evo kako izgleda jedan BASIC program kojim se za datu kontekstno slobodnu gramatiku generišu sve njene teoreme ili se za datu reč ispituje da li ona jeste ili nije teorema. Program je upitno-odgovorni, za promenljive se može uzeti ma koji početni komad ovog spiska

$$s,\ t,\ u,\ v,\ w,\ x,\ y,\ z$$

Ako se, recimo, opredelimo za tri promenljive, onda će to biti upravo: s, t, u. Sto se tiče završnog alfabeta, on se ne precizira posebno: svi znaci koji učestvuju u pravilima a koji nisu promenljive smatraju se završnim. Promenljiva s je uzeta za aksiomu. Pravila se zadaju, na odgovarajući upit, zadavanjem odgovarajućih desnih strana. Program teče ovako: Najpre se na programom određenim načinom uočava gramatika. U tu svrhu prvo se zadaje koliko promenljivih ima ta gramatika. Dalje se za svaku promenljivu navodi broj njenih zamena, kao i same te zamene. Tačnije, za svaku promenljivu se navode sve reči sa kojima se ta promenljiva sme zameniti. Najzad se odgovara na programom postavljeno pitanje da li se želi nizanje svih teorema do željenog broja, ili se želi utvrđivanje da li data reč jeste teorema. Evo i samog programa [urađenog u saradnji sa S. B. Prešićem]:

```
10 PRINT "Kakav zadatak uraditi: N I Z A T I teoreme do zeljenog broja ili
20 PRINT "za datu rec ispitati da li je T E O R E M A (gramatike Comskog
30 PRINT "koja ce se naknadno zadati) ?"
40 PRINT:PRINT
50 PRINT "Ako treba NIZATI teoreme kucati rec NIZATI"
60 INPUT "a u drugom slucaju kucati rec TEOREMA";PIT$
70 IF PIT$<>"NIZATI" AND PIT$<>"TEOREMA" THEN PRINT:PRINT:PRINT " G R E S K A
80 PRINT
90 PRINT "Najpre cemo zadati gramatiku"
100 DIM VAR$(8),MM(50),D$(8,10),T$(1000),ZT$(100)
110 PRINT "Koliko promenljivih ?":INPUT M
120 REM promenljive su s,t, ... ,z; m<=8
130 FOR I=1 TO M:VAR$(I)=CHR$(114+I):NEXT I
140 FOR I=1 TO M:PRINT "Za promenljivu ";VAR$(I);" koliko zamena?":INPUT MM(I)
150 FOR J=1 TO MM(I):PRINT "Koja je ";J;"-zamena?  ":INPUT D$(I,J):NEXT J,I
160 IF PIT$="TEOREMA" THEN GOSUB 380:GOTO 680
170 T$(1)=VAR$(1)
180 INPUT "Do kog broja redom nizati teoreme";BROJ
190 PRINT " Teorema ";1;":",T$(1)
200 L=1:T=1
210 TT=T
220 FOR I=L TO TT
230 PAM=0
240 FOR K=1 TO LEN(T$(I))
250 S$=MID$(T$(I),K,1)
260 IF ASC(S$)<115 OR ASC(S$)>122 THEN GOTO 330
270 FOR J=1 TO MM(ASC(S$)-114)
280 PAM=1:T=T+1
290 T$(T)=MID$(T$(I),1,K-1)+D$(ASC(S$)-114,J)+MID$(T$(I),K+1,LEN(T$(I))-K)
300 PRINT " Teorema ";T;":",T$(T)
310 IF T>BROJ-1 THEN GOTO 370
320 NEXT J
330 NEXT K
340 NEXT I
350 KK=KK+1
360 L=TT+1:GOTO 210
370 GOTO 680
380 INPUT "Koju rec ispitivati ";W$
390 T$(1)=W$
400 L=1:T=1
410 TT=T
420 FOR I=L TO TT
```

```
430 FOR X=1 TO M:FOR Y=1 TO MM(X)
440 MER=LEN(D$(X,Y))
450 FOR Z=1 TO LEN(T$(I))-MER+1
460 S$=MID$(T$(I),Z,MER)
470 IF S$<>D$(X,Y) THEN GOTO 530
480 BR=BR+1
490 T$=MID$(T$(I),1,Z-1)+CHR$(X+114)+MID$(T$(I),Z+MER,LEN(T$(I))-Z-MER+1)
500 GOSUB 630
510 IF P=0 THEN T=T+1:T$(T)=T$
520 IF T$(T)="s" THEN GOTO 600
530 NEXT Z
540 NEXT Y
550 NEXT X
560 NEXT I
570 KK=KK+1
580 IF BR=0 THEN GOTO 610
590 BR=0:L=TT+1:GOTO 410
600 PRINT "Jeste teorema":GOTO 620
610 PRINT "nije teorema"
620 STOP
630 FOR II=1 TO T
640 IF T$=T$(II) THEN P=1:GOTO 670
650 NEXT II
660 P=0
670 RETURN
680 END
```

Glavna primena gramatika Čomskog jeste u istraživanjima o prirodnim jezicima, radi kojih su one, uostalom, i pravljene. Ta su se istraživanja nadovezala na već zaokrugljena znanja o gramatičkim kategorijama. Određivan i definisan stolećima, još od Platona i Aristotela, spisak gramatičkih kategorija je dobrim delom uobličen radovima Ajdukeviča 30-tih godina ovog veka. Kao najčešće i najznačajnije gramatičke kategorije ističu se sledeće:

CN	(kategorija zajedničkih imenica)
NP	(kategorija imeničkih izraza)
IV	(kategorija neprelaznih glagola)
TV	(kategorija prelaznih glagola)
ADJ	(kategorija prideva)
ADV	(kategorija priloga)
PRP	(kategorija predloga)
CON	(kategorija veznika)
S	(kategorija rečenica)

[Istina, ubrzo posle objavljivanja glavnih rezultata Čomskog u istraživanja

prirodnih jezika se uključuje čitav niz matematičara, posebno logičara, na čelu sa Montegjuom. Tada se ispostavilo da se spisak gramatičkih kategorija mora bitno menjati: neke su potpuno nestale, neke su se raspale na više novih kategorija, a pojavile su se i sasvim nove kategorije koje lingvisti nisu nikada bili u stanju da uoče. Ali o tom potom.]

Evo sada kako izgleda jedna gramatika Čomskog kojom se generišu jednostavne rečenice engleskog jezika sastavljene od jedne vlastite imenice i jednog neprelaznog glagola.

PRIMER 6. *Pomoćni simboli te gramatike su* NP , IV , S *(oznake kategorija imeničkih izraza, neprelaznih glagola i rečenica), aksioma je S, završni simboli su izvesne vlastite imenice, na primer: John, Mary, Bill, Peter, kao i izvesni neprelazni glagoli (treće lice jednine sadašnjeg vremena), recimo: runs, walks, talks, writes (glagol writes je uzet kao neprelazan, tj. sa značenjem* **baviti se pisanjem**, *tj.* **biti pisac**). *Zamene su:*

$$\begin{array}{llll} \mathrm{NP} \longrightarrow \textit{John}, & \mathrm{NP} \longrightarrow \textit{Mary}, & \mathrm{NP} \longrightarrow \textit{Bill}, & \mathrm{NP} \longrightarrow \textit{Peter} \\ \mathrm{IV} \longrightarrow \textit{runs}, & \mathrm{IV} \longrightarrow \textit{walks}, & \mathrm{IV} \longrightarrow \textit{talks}, & \mathrm{IV} \longrightarrow \textit{writes} \end{array}$$

$$\mathrm{S} \longrightarrow \mathrm{NP\ VP}$$

Kao završne teoreme dobijaju se rečenice:

John runs, Mary runs, Bill runs, Peter runs, John walks, i sl.,

ukupno 4x4=16 rečenica. Dokaz prve od njih izgleda:

$$\begin{array}{rcll} \mathrm{S} & \longrightarrow & \mathrm{NP\ IV} & \text{(zadnje pravilo)} \\ \mathrm{NP\ IV} & \longrightarrow & \textit{John}\ \mathrm{IV} & \text{(primena prvog pravila)} \\ \textit{John} & \longrightarrow & \textit{runs} & \text{(primena pravila IV} \longrightarrow \textit{runs}) \end{array}$$

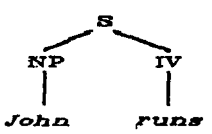
Taj se dokaz može skraćeno zapisati i u obliku ovakvog produženog niza:

$$\mathrm{S} \longrightarrow \mathrm{NP\ IV} \longrightarrow \mathrm{John\ IV} \longrightarrow \mathrm{John\ runs},$$

što je uobičajeno pisanje u ma kojoj gramatici Čomskog, a može se zgodno prikazati navedenim drvetom. Pored prethodnog, rečenica John runs očigledno ima i ovaj dokaz:

$$\mathrm{S} \longrightarrow \mathrm{NP\ IV} \longrightarrow \mathrm{NP\ runs} \longrightarrow \mathrm{John\ runs}$$

kome takođe odgovara prikazano drvo.



Prethodna gramatika se može dalje dograđivati tako da se kao nove teoreme pojavljuju i rečenice u kojima učestvuju prelazni glagoli, složeniji imenički izrazi, pridevi, rečenični veznici i dr. Evo jedne takve gramatike:

PRIMER 7. *Njeni pomoćni simboli su:* CN , NP , IV , TV , ADJ , DET, CON , S, *aksioma je* S , *završni simboli su članovi narednih skupova:*

$A_1 = \{man, woman, apple, cat, letter, fish\}$
$A_2 = \{John, Mary, Bill, Peter, he_0, he_2, he_3, \ldots , he_{10}\}$
$A_3 = \{runs, walks, talks, writes\}$
$A_4 = \{eats, loves, seeks, reads, writes\}$
$A_5 = \{tall, sweet, long, black, old, odd\}$
$A_6 = \{the, a, every\}$
$A_7 = \{and, or\}$

[U imeničke izraze je uključeno i deset zamenica trećeg lica jednine muškog roda, nešto što nije baš uobičajeno u govornom jeziku ali je postalo svakodnevna praksa u današnjim matematičkim istraživanjima tih jezika. Uostalom, neophodnost uvođenja više zamenica se lako uočava u raznim jezičkim vratolomijama koje se pojavljuju u slučajevima kada se prirodno nametne potreba za više takvih ili nekih drugih imeničkih izraza iste vrste. Evo kako se tada obično dovijamo, kao u primeru:

> **"Petar je dobio pismo od jedne žene koje je donela druga žena, a ta druga žena je prijateljica treće žene koju je Petar upoznao prošle godine na letovanju."**

Pored toga kao gramatička kategorija pojavljuje se DET-kategorija određivača. To je jedna od potpuno novih i veoma značajnih kategorija koje lingvisti duguju matematičarima. Pomenimo još jednu značajnu kategoriju koju su uveli matematičari, a o kojoj će u daljem još biti reči. To je kategorija QU-kategorija kvantora, kojoj pripadaju izrazi kao:

the man, a man, every man, every woman, most men, both man and woman.]

Zamene su:

$$\begin{array}{llll}
\text{CN} & \longrightarrow \text{ma koja reč skupa } A_1, & \text{NP} & \longrightarrow \text{ma koja reč skupa } A_2 \\
\text{IV} & \longrightarrow \text{ma koja reč skupa } A_3, & \text{TV} & \longrightarrow \text{ma koja reč skupa } A_4 \\
\text{ADJ} & \longrightarrow \text{ma koja reč skupa } A_5, & \text{CON} & \longrightarrow \text{ma koja reč skupa } A_6
\end{array}$$

$$\text{S} \longrightarrow \text{NP IV}, \quad \text{S} \longrightarrow \text{S CON S}, \quad \text{IV} \longrightarrow \text{TV NP},$$
$$\text{CN} \longrightarrow \text{ADJ CN}, \quad \text{NP} \longrightarrow \text{DET CN}$$

U toj gramatici se mogu izvesti složene rečenice kakve su na primer:

John loves Mary, John loves every woman, Every man loves a woman, Peter writes a long letter, A woman eats a sweet apple, John runs and John eats a sweet apple, A man seeks a black cat and he runs,

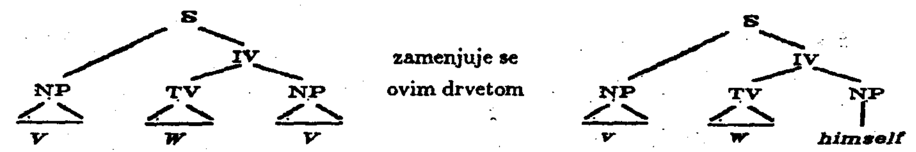
ali takođe i rečenice sintaksno potpuno ispravne ali neobičnog značenja, kao što su ove, na primer:

Peter writes a black apple, The black apple runs and every cat writes a long letter, The long odd letter reads every black cat, A tall black cat writes every old woman, The woman seeks a tall odd fish or he_1 *writes*

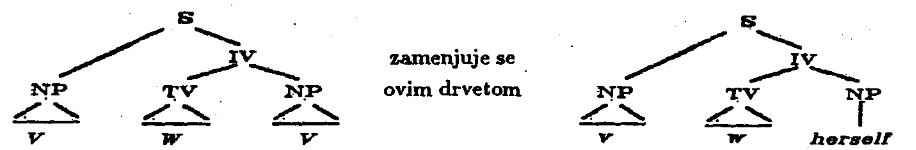
Takve rečenice, međutim, uopšte ne smetaju; one su neuobičajene u svakodnevnom pisanom i govornom jeziku ali su u poeziji sasvim moguće, dozvoljene, prihvatljive. Smetaju, međutim, rečenice:

John loves John, John loves he_3*, The tall woman loves the tall woman*

za koje se ne može reći samo da su neuobičajene već i da su gramatički neispravne. Kako izbeći takve rečenice? Jedan način je da se odgovarajuća gramatika obogati takozvanim **transformacijama**. Na primer, umesto rečenice *John loves John* treba da se pojavi rečenica *John loves himself*. Odgovarajuća trasformacija kojom se to postiže izgleda:



ukoliko je v muškog roda a u slučaju ženskog roda umesto prethodne usvajamo transformaciju:



[Kao što je uobičajeno trouglasti završeci drveta, kakav je recimo: NP v označavaju ispušteno poddrvo čiji je koren NP , u uočenom primeru.]

Kao što se uočava, tim transformacijama se jedno izvođenje (prikazano odgovarajućim drvetom) zamenjuje drugim izvođenjem. Pri tome su NP , IV ma koji imenički izraz, odnosno ma koji prelazani glagol (u ovom našem primeru NP je element skupa A_2, a IV je element skupa A_4). Transformacije se obično zadaju mnogo kraće, bez isticanja drveta izvođenja (ali tada je često neophodno uvesti više pomoćnih simbola.) Tako se prethodne dve transformacije obično zadaju u obliku:

$$\text{NP TV NP} \longrightarrow \text{NP TV } \textit{himself} \quad \text{(Ukoliko je NP muškog roda jednine)}$$
$$\text{NP TV NP} \longrightarrow \text{NP TV } \textit{herself} \quad \text{(Ukoliko je NP ženskog roda jednine)}$$

Slično se može uvesti transformacija kojom se rečenica *John loves* he_3 prevodi u rečenicu *John loves* him_3, koja je gramatički ispravna, ili transformacija kojom se ta ista rečenica prevodi u pasivni oblik: He_3 *is loved by John*, a takođe i transformacije kojima se od rečenice kakva je na primer:

Every man likes some woman

prelazi na izraze:

(7) *Every man such that that man likes some woman*
(8) *Some woman such that every man likes that woman*

Transformacije kojima se to postiže su dužine dva i izgledaju:

DET CN , DET CN TV NP $\longrightarrow$ DET CN *such that that* CN TV NP
DET CN , NP TV DET CN $\longrightarrow$ DET CN such that NP TV that CN

Odgovarajućim transformacijama se rečenice izvedene u nekoj gramatici Čomskog mogu prevesti u rečenice u pluralu, u nekom prošlom vremenu, u budućem vremenu, dalje u rečenice u kojima se sva pojavljivanja istog imeničkog izraza (izuzev prvog) zamenjuju odgovarajućom ličnom zamenicom i tako dalje.

Gramatike Čomskog u kojima se pored pravila zamene pojavljuje i izvestan broj transformacija nazivaju se, inače, **transformacionim** gramatikama; kao i kontekstno slobodne i one takođe potiču od Čomskog. Iz prethodnih primera se vidi da se transformacione gramatike dobiju iz običnih kontekstno slobodnih gramatika dodavanjem izvesnih transformacija, koje nisu ništa drugo do pravila izvođenja nad skupom svih reči uočene gramatike.

Očigledno je da su kontekstno slobodne gramatike bitno obogaćene uvođenjem transformacija. Međutim, ne izgleda baš prirodno da se, recimo, rečenica srpskohrvatskog jezika:

Ona je upravo otišla

dobija odgovarajućom transformacijom iz rečenice:

On je upravo otišao

[Ako ni zbog čega drugog ono zbog izbegavanja vekovne dominacije muškog roda. Ali ne vide se ni valjani razlozi, mislim na gramatičke, po kojima bi trebalo usvojiti da se prva rečenica dobija transformacijom druge.]

Slično se odnosi i na uzajamnu vezu rečenica u svakom od narednih parova:

Zoran je lep, Milica je lepa
Zoran svira, Zoran i Milica sviraju
Zoran spava, Zoranu se spava

Uvođenje transformacija kojima se povezuju rodovi, brojevi, padeži, vremena i razne druge promenljive odrednice kojima obiluju naročito jezici sa izrazitom fleksijom, može biti veoma komplikovano ili, što je još veća mana, veoma neprirodno. To je bio jedan od razloga što je stvoren još moćniji matematički aparat za istraživanje prirodnih jezika. To su **gramatike metamorfoze** [potiču od Kolmeroera (Colmerauer) iz 1975. godine], kod kojih "svet" na kome se događaju jezička zbivanja nije više tako širok i neuhvatljiv kao kod kontekstno slobodnih i transformacionih gramatika [Kod njih je to, setimo se, skup svih reči nad datim alfabetom], već je uži, ali zato pravilniji i svojstveno bogatiji, pa se stoga njime lakše "hvataju" razna flektivna svojstva. To je, naime jezik izraza, terma nad nekim operacijskim jezikom, takozvani **Erbranov svet** (Herbrand). Podsećamo da se Erbranov svet, označimo ga sa $Term[F]$, nad operacijskim jezikom F i fiksiranim skupom promenljivih, recimo $\{x_1, x_2, x_3, \dots\}$, uvodi ovako:

To je najmanji skup koji sadrži sve promenljive, sve operacijske znake dužine nula (tj. sve znake konstanata) iz F, i za svaki operacijski znak $f \in F$ dužine n sadrži reč $f(t_1, t_2, \dots, t_n)$, gde su t_1, t_2, ..., t_n ma koji termi iz $Term[F]$.

Gramatike metamorfoze se, tada, uvode definicijom:

DEFINICIJA 2. *Neka je F izvestan skup operacijskih znakova, za koji se zahteva da obavezno sadrži znake $\cdot$ (dužine dva) i nil (dužine nula) [nil, ustvari, ima ulogu prazne reči], i neka je $Term[F]$ odgovarajući Erbranov svet. Gramatika metamorfoze (nad F) je određena sa tri skupa (podskupa od $Term[F]$):*

T (skup završnih, terminalnih izraza)

N (skup pomoćnih, neterminalnih izraza)

A (skup polaznih izraza, tj.aksioma)

i izvesnim pravilima zamene nad skupom $(T \cup N)$. Pri tome su T i N međusobno disjunktni, dok je A podskup od N. Za zamene se zahteva da ne budu oblika $X \longrightarrow nil$.

Kao i kod prethodnih gramatika, teoreme su sve one reči, bliže svi oni izrazi iz $Term(F)$, koji slede iz aksioma, pri čemu i u ovom slučaju razlikujemo završne od nezavršnih teorema. Skup svih završnih teorema čini **jezik** generisan uočenom gramatikom metamorfoze.

PRIMER 8. *Skup F je $\{nil,\ zero,\ a,\ b,\ suite,\ bs,\ suc,\ \cdot\}$, pri čemu:*

$$red(nil) = 0, \quad red(zero) = 0, \quad red(a) = 0, \quad red(b) = 0$$
$$red(suite) = 1, \quad red(bs) = 1, \quad red(suc) = 1$$
$$red(\cdot) = 2$$

[Sa $red(f)$ smo označili red, tj. dužinu operacijskog znaka f.]

Skupovi T, A, N su sledeći:

$$T = \{a, b\}, \quad A = \{suite(x) : x \in Ter(F)\}, \quad N = A \cup \{bs(x) : x \in Ter(F)\}$$

Zamene su shema-zamene;

$$suite(x) \longrightarrow a \cdot suite(suc(x)),\ suite(x) \longrightarrow bs(x)$$
$$bs(suc(x)) \longrightarrow b \cdot bs(x),\ bs(zero) \longrightarrow nil$$

gde je x ma koji izraz iz $Term(F)$.

Jedna teorema (završna) je, na primer $a \cdot (b \cdot (b \cdot (b \cdot nil)))$ a jedan njen dokaz je:

$$\begin{aligned} suite(suc(suc(zero))) &\longrightarrow a \cdot suite(suc(suc(suc(zero)))) \\ &\longrightarrow a \cdot bs(suc(suc(suc(zero)))) \\ &\longrightarrow a \cdot (b \cdot bs(suc(suc(zero)))) \\ &\longrightarrow a \cdot (b \cdot (b \cdot bs(suc(zero)))) \\ &\longrightarrow a \cdot (b \cdot (b \cdot (b \cdot bs(zero)))) \\ &\longrightarrow a \cdot (b \cdot (b \cdot (b \cdot \mathrm{nil}))) \end{aligned}$$

Pri tome su redom primenjivane zamene: prva, druga, treća (tri puta uzastopce), i najzad četvrta. Nije teško dokazati da ta gramatika generiše jezik kome pripadaju reči:

$$a \cdot (b \cdot (b \cdot nil)), a \cdot (a \cdot (b \cdot (b \cdot (b \cdot nil)))), \quad a \cdot (a \cdot (a \cdot (b \cdot (b \cdot (b \cdot (b \cdot nil))))))$$

i uopšte reči koje su prozvod i a-ova, odnosno j b-ova i konstante nil, pri čemu su i, j ma koji prirodni brojevi i još je $j \geq i$, a proizvod je takozvani "desni", tj. sa zagradama grupisanim na desnu stranu. Posebna uloga koju imaju operacijski znaci $\cdot$ i nil u vezi je sa jednom važnom teoremom o algebarskim strukturama prema kojoj se svaka algebarska struktura (na proizvoljnom operacijskom jeziku) može izomorfno potopiti u strukturu sa jednom binarnom operacijom i izvesnim brojem konstanata. [**Štaviše, ta binarna operacija može biti i asocijativna, a može imati i jedinični element.**] Upravo tome služe operacijski znaci $\cdot$ i nil. Pomoću njih se ma koja operacija f dužine n može ovako izraziti

$$f(x_1, x_2, \ldots, x_n) = (f \cdot (x_1 \cdot (x_2 \ldots, (x_n \cdot nil) \ldots))) \tag{9}$$

Pri tome, znaci f sa leve i desne strane jednakosti, iako su označeni na isti način, imaju i različit smisao i različitu ulogu: sa leve strane je to operacijski znak dužine n koji se jednakošću (9) definiše, a sa desne strane je to znak konstante, koji za svaku novu operaciju mora takođe biti nov, do tada neupotrebljen.

Inače, uobičajeno je da se izraz oblika $(a_1 \cdot (a_2 \cdot (a_3 \ldots, (a_n \cdot nil) \ldots)))$ naziva **listom** i da se označava ovako: $[a_1, a_2, \ldots, a_n]$. Pri tome se prvi član a_1 naziva **glavom** a ostatak liste, tj. $[a_2, a_3, \ldots, a_n]$ je takozvani **rep**. Koristeći tu oznaku jednakost (9) se zapisuje u obliku:

$$f(x_1, x_2, \ldots, x_n) = [f, x_1, x_2, \ldots, x_n] \tag{10}$$

Pomenimo da je izražavanje operacija pomoću $\cdot$ i nil dato jednakostima (9), odnosno (10) ugrađeno u neke programske jezike, na primer u LISP i PROLOG.

Evo sada jednog primera koji se odnosi na prirodne jezike (i to ponovo na engleski) iz koga se jasno vidi kako se podesno izabranom gramatikom metámorfoze mogu "uhvatiti" razna flektivna svojstva kao što su rod i broj.

PRIMER 9. *Operacijski jezik čine znaci: s, cn, np, iv, tv, det, singular, plural, person, thing, musician, musicians, violin, plays, play, the i, naravno, znaci ·, nil, koji se podrazumevaju. Pri tome su s, cn, np, iv, tv, det, kao i svi termi u kojima oni učestvuju, pomoćni znaci a sve preostale reči su završni znaci. Dalje, np i cn su dužine dva, tv i iv su dužine jedan a svi ostali znaci su dužine nula, odnosno oni su konstante. Jedina aksioma je s dok su zamene :*

```
                    s ⟶ np(Number, person), iv(Number)
     np(Number, Type) ⟶ det, cn(Number, Type)
           iv(Number) ⟶ tv(Number), np(Number1, thing)
cn(singular, person) ⟶ [musician]
  cn(plural, person) ⟶ [musicians]
 cn(singular, thing) ⟶ [violin]
         tv(singular) ⟶ [plays]
           tv(plural) ⟶ [play]
                  det ⟶ [the]
```

U toj gramatici reči koje počinju velikim slovom su promenljive, dok su gramatičke kategorije, kao što se vidi, označene malim slovima. Najzad, zarez koji razdvaja terme na desnoj strani zamena ima ulogu konkatenacije (za liste). Usvojene oznake su u skladu sa uobičajenim oznakama u PROLOG-u, programskom jeziku koji je tesno povezan i razvijan uporedo sa gramatikama metamorfoze [i njegov tvorac je Kolmaroe], te je stoga i najpodesniji za prevođenje date gramatike metamorfoze u odgovarajući kompjuterski program. Gramatikom u ovom primeru generišu se rečenice:

The musician plays the violin, The musicians play the violin

koje se dobijaju u obliku lista:

[*the, musician, plays, the, violin*], [*the, musicians, play, the, violin*]

Prevođenje gramatike metamorfose u programski jezik PROLOG vrši se na standardan način, uvođenjem pomoćnih promenljivih. Tako imamo da se

```
a(Prom1) ⟶ [b],     a(Prom1) ⟶ b(Prom2),
a(Prom1) ⟶ b(Prom2),c(Prom3)
```

redom prevode u ove naredbe PROLOG-a:

```
a(Prom1,[b|U],U),   a(Prom1,U,V)-- > b(Prom2,U,V),
a(Prom1,[b|U],U),   a(Prom1,U,V)-- > b(Prom2,U,V),
                    a(Prom1,U,W)-- > b(Prom2,U,V),   c(Prom3,U,W).
```

gde su sa *Prom*1, *Prom*2, *Prom*3 označeni proizvoljni nizovi promenljivih. Na primer, zamenama gramatike iz prethodnog primera odgovara ovaj program u PROLOG-u:

```
np(Number,Type,U,W):-det(U,V),cn(Number,Type,V,W).
                s(U,W):-np(Number,person,U,V),iv(Number,V,W).
np(Number,Type,U,W):-det(U,V),cn(Number,Type,V,W).
      iv(Numer,U,W):-tv(Number,U,V),np(Number1,thing,V,W).
                    cn(singular,person,[musician|U],U).
                    cn(plural,person,[musician|U],U).
                    cn(singular,thing,[violin|U],U).
                    tv(singular,[plays|U],U).
                    tv(plural,[play|U],U).
                    det([the|U],U).
```

Naravno prethodna gramatika se jednostavno proširuje do gramatike kojom se generišu slične rečenice obrazovane od jednog imeničkog izraza, prelaznog glagola i još jednog imeničkog izraza; za to je dovoljno rečnik polaznih reči engleskog jezika proširiti novim rečima i dodati odgovarajuće zamene.

Gramatike metamorfoze su posebno dobro primenljive u izrazito flektivnim jezicima kakav je srpskohrvatski. Naravno flektivnost povlači za sobom uvođenje većeg broja parametara nego što je to slučaj u engleskom jeziku, te su stoga i odgovarajuće gramatike komplikovanije. Navodimo jedan program u PROLOG-u kojim se, polazeći od datog rečnika srpskohrvatskog jezika, generišu sve elementarne rečenice (na tom rečniku), tj. sve rečenice sagrađene primenom nekog glagola na odgovarajući broj imeničkih izraza koji su, uz to, u odgovarajućim padežnim oblicima. Program izgleda ovako:

```
imenica(jednina,muski,zivo)-- >[muzicar,muzicara,muzicaru,muzicara,
                                muzicarom,muzicaru].

imenica(jednina,muski,zivo)-- >[covek,coveka,coveku,coveka,covekom,
                                coveku].

imenica(jednina,zenski,zivo)-- >[zena,zene,zeni,zenu,zenom,zeni].

imenica(jednina,zenski,nezivo)-- >[violina,violine,violini,violinu,
                                   violinom,violini].

imenica(jednina,zenski,nezivo)-- >[jabuka,jabuke,jabuci,jabuku,
                                   jabukom,jabuci].

padez_imenice(Broj,Rod,Vrsta,nom,[A|S],S): -imenica(Broj,Rod,Vrsta,
                                             [A,B,C,D,E,F|S],S).
```

```
padez_imenice(Broj,Rod,Vrsta,gen,[B|S],S): -imenica(Broj,Rod,Vrsta,
                                            [A,B,C,D,E,F|S],S).

padez_imenice(Broj,Rod,Vrsta,dat,[C|S],S): -imenica(Broj,Rod,Vrsta,
                                            [A,B,C,D,E,F|S],S).

padez_imenice(Broj,Rod,Vrsta,akuz,[D|S],S): -imenica(Broj,Rod,Vrsta,
                                             [A,B,C,D,E,F|S],S).

padez_imenice(Broj,Rod,Vrsta,inst,[E|S],S): -imenica(Broj,Rod,Vrsta,
                                             [A,B,C,D,E,F|S],S).

padez_imenice(Broj,Rod,Vrsta,lok,[F|S],S): -imenica(Broj,Rod,Vrsta,
                                            [A,B,C,D,E,F|S],S).

glagol(jednina,[zivo],[nom])-- >[spava];
                                [pise];
                                [peva].

glagol(mnozina,[zivo],[nom])-- >[spavaju];
                                [pisu];
                                [pevaju].

glagol(jednina,[zivo],[dat])-- >[se_spava];
                                [se_zeva];
                                [se_putuje].

glagol(mnozina,[zivo],[dat])-- >[se_spava];
                                [se_zeva].
                                [se_putuje];

glagol(jednina,[zivo,nezivo],[nom,akuz])-- >[jede];
                                            [svira];
                                            [pise].

glagol(mnozina,[zivo,nezivo],[nom,akuz])-- >[jedu];
                                            [sviraju];
                                            [pisu].

glagol(jednina,[zivo,zivo,nezivo],[nom,dat,akuz])-- >[pise];
                                                     [daje];
                                                     [nosi].

glagol(mnozina,[zivo,zivo,nezivo],[nom,dat,akuz])-- >[pisu];
                                                     [daju];
                                                     [nose].
```

```
glagol(jednina,[zivo,zivo],[nom,dat])-- >[pise];
                                         [se_divi];
                                         [se_okrece].

glagol(mnozina,[zivo,zivo],[nom,dat])-- >[pisu];
                                         [se_dive];
                                         [se_okrecu].

glagol(jednina,[zivo,nezivo],[nom,dat])-- >[se_divi];
                                           [se_okrece].

glalol(mnozina,[zivo,nezivo],[nom,dat])-- >[se_dive];
                                           [se_okrecu].

glagol(Broj,[Vrste|[ ]],[nom])-- >glagol(Broj,[Vrste|S],[nom,akuz]),
                                  padez_imenice(Broj1,Rod1,S,akuz).

glagol(Broj,[Vrste|[ ]],[nom])-- >glagol(Broj,[Vrste|S],[nom,dat]),
                                  padez_imenice(Broj1,Rod1,S,dat).

glagol(Broj,[Vrste|[ ]],[nom,akuz])-- >glagol(Broj,[Vste|S],[nom,dat,akuz]),
                                       padez_imenice(Broj1,Rod1,S,dat).

prod([A1,A2,A3,A4,A5,A6],[B1,B2,B3,B4,B5,B6],[C1,C2,C3,C4,C5,C6]):-
              C1=[A1,B1],C2=[A2,B2],C3=[A3,B3],
              C4=[A4,B4],C5=[A5,B5],C6=[A6,B6].

term(jednina,zenski,zivo)-- >[vera,vere,veri,veru,verom,veri].

term(jednina,muski,zivo)-- >[jova,jove,jovi,jovu,jovom,jovi].

term(Broj,Rod,Vrsta,[C1,C2,C3,C4,C5,C6|S],S):-
            odredivac(Broj,Rod,Vrsta,[A1,A2,A3,A4,A5,A6|S],S),
            imenica(Broj,Rod,Vrsta,[B1,B2,B3,B4,B5,B6|S],S),
            prod([A1,A2,A3,A4,A5,A6],[B1,B2,B3,B4,B5,B6],
                 [C1,C2,C3,C4,C5,C6]).

append([ ],X,X).

append([X|Y],Z,U):-append(Y,Z,W),
             U=[X|W].

padez_terma(Broj,Rod,Vrsta,nom,T,S):-term(Broj,Rod,Vrsta,
                                   [A,B,C,D,E,F|S],S),
                                   a(A,S,T).
```

```
padez_terma(Broj,Rod,Vrsta,gen,T,S): -term(Broj,Rod,Vrsta,
                                         [A,B,C,D,E,F|S],S),
                                         a(B,S,T).

padez_terma(Broj,Rod,Vrsta,dat,T,S): -term(Broj,Rod,Vrsta,
                                         [A,B,C,D,E,F|S],S),
                                         a(C,S,T).

padez_terma(Broj,Rod,Vrsta,akuz,T,S): -term(Broj,Rod,Vrsta,
                                          [A,B,C,D,E,F|S],S),
                                          append(D,S,T).

padez_terma(Broj,Rod,Vrsta,inst,T,S): -term(Broj,Rod,Vrsta,
                                          [A,B,C,D,E,F|S],S),
                                          a(E,S,T).

padez_terma(Broj,Rod,Vrsta,lok,T,S): -term(Broj,Rod,Vrsta,
                                         [A,B,C,D,E,F|S],S),
                                         a(F,S,T).

a(W,S,T): -ifthenelse((W=[U|V]),append(W,S,T),(T=[W|S])).

odredivac(jednina,muski,zivo)-- >[neki,nekog,nekom,nekog,nekim,nekom];
                                 [svaki,svakog,svakom,svakog,svakim,
                                 svakom].

odredivac(mnozina,muski,zivo)-- >[neki,nekih,nekim,neke,nekim,nekim];
                                 [svaki,svakih,svakim,svake,svakim,
                                 svakim].

odredivac(jednina,muski,nezivo)-- >[neki,nekog,nekom,neki,nekim,nekom];
                                   [svaki,svakog,svakom,svaki,svakim,
                                   svakom].

odredivac(mnozina,muski,nezivo)-- >[neki,nekih,nekim,neke,nekim,nekim];
                                   [svaki,svakih,svakim,svake,svakim,
                                   svakim].

odredivac(jednina,zenski,zivo)-- >[neka,neke,nekoj,neku,nekom,nekoj];
                                  [svaka,svake,svakoj,svaku,svakom,
                                  svakoj].

odredivac(mnozina,zenski,zivo)-- >[svake,svakih,svakim,svake,svakim,
                                  svakim];
                                  [neke,nekih,nekim,neke,nekim,nekim].
```

```
odredivac(jednina,zenski,nezivo)-- >[neka,neke,nekoj,neku,nekom,nekoj];
                                    [svaka,svake,svakoj,svaku,svakom,
                                    svakoj].

odredivac(mnozina,zenski,nezivo)-- >[neke,nekih,nekim,neke,nekim,nekim];
                                    [svake,svakih,svakim,svake,svakim,
                                    svakim].

padez_odredivaca(Broj,Rod,Vrsta,nom,[A|S],S): -odredivac(Broj,Rod,Vrsta,
                                              [A,B,C,D,E,F|S],S).

padez_odredivaca(Broj,Rod,Vrsta,gen,[B|S],S): -odredivac(Broj,Rod,Vrsta,
                                              [A,B,C,D,E,F|S],S).

padez_odredivaca(Broj,Rod,Vrsta,dat,[C|S],S): -odredivac(Broj,Rod,Vrsta,
                                              [A,B,C,D,E,F|S],S).

padez_odredivaca(Broj,Rod,Vrsta,akuz,[D|S],S): -odredivac(Broj,Rod,Vrsta,
                                               [A,B,C,D,E,F|S],S).

padez_odredivaca(Broj,Rod,Vrsta,inst,[E|S],S): -odredivac(Broj,Rod,Vrsta,
                                               [A,B,C,D,E,F|S],S).

padez_odredivaca(Broj,Rod,Vrsta,lok,[F|S],S): -odredivac(Broj,Rod,Vrsta,
                                              [A,B,C,D,E,F|S],S).

recenica-- >padez_terma(Broj1,Rod1,Vrsta1,nom),
            glagol(Broj1,[Vrsta1],[nom]).

recenica-- >padez_terma(Broj1,Rod1,Vrsta1,dat),
            glagol(Broj1,[Vrsta1],[dat]).

recenica-- >padez_terma(Broj1,Rod1,Vrsta1,nom),
            glagol(Broj1,[Vrsta1,Vrsta2],[nom,akuz]),
            padez_terma(Broj2,Rod2,Vrsta2,akuz).

recenica-- >padez_terma(Broj1,Rod1,Vrsta1,nom),
            glagol(Broj1,[Vrsta1,Vrsta2],[nom,dat]),
            padez_terma(Broj2,Rod2,Vrsta2,dat).
```

```
recenica-- >padez_terma(Broj1,Rod1,Vrsta1,nom),
            glagol(Broj1,[Vrsta1,Vrsta2,Vrsta3],[nom,dat,akuz]),
            padez_terma(Broj2,Rod2,Vrsta2,dat),
            padez_terma(Broj3,Rod3,Vrsta3,akuz).
```

Nekoliko objašnjenja u vezi sa programom: nazivi svih gramatičkih kategorija su prevedeni na srpskohrvatski. Imenice se zadaju kao uređene šestorke, tj. kao odgovarajuće liste, padežnih oblika (vokativ je izostavljen budući da on ne učestvuje u izgradnji rečenice, već ima isključivo ulogu dozivanja). Kategorija imenica zavisi od broja, roda i vrste (živo, neživo). Glagoli su dužine jedan, dva i tri i zavise od broja, vrsta i padeža. Pri tome su sada vrste i padeži liste čija je dužina jednaka gužini glagola a članovi tih lista su popunjeni odgovarajućim vrstama i padežima u kojima moraju biti imenički izrazi na koje se glagol primenjuje. Na primer glagol **dati** je dužine tri, i ima kao odrednice: jedninu, [zivo,zivo,nezivo],[nom,dat,akuz]. Od tog glagola i tri imenička izraza čije su vrste redom zivo,zivo,nezivo, koji su po redu u nominativu, dativu, akuzativu, i od kojih je prvi izraz u jednini obrazuje se elementarna rečenica kakva je na primer:

Neki čovek daje Jovi neku violinu

Slično imenicama, i određivači se zadaju kao liste odgovarajućih padežnih oblika. Kategoriji terma pripadaju vlastite imenice (zadane listama padežnih oblika) i imenički izrazi čija su pravila izgradnje data. Za imenice, određivače i terme data su odvojeno pravila izgradnje svih padežnih oblika (nominativa, genitiva, dativa, akuzativa, instrumentala i lokativa za koje koristimo skraćenice po redu: nom, gen, dat, akuz, inst, lok). Najzad, u program je uvršćena i definicija operacije append (dopisivanje lista) koja u nekim varijantama prologa nije ugrađena. Navedenim programom se generišu, recimo, rečenice:

Vera spava, Veri se spava, Svi ljudi pevaju, Svaki čovek se divi nekoj ženi, Jova svira violinu, Svaki muzičar se okreće nekoj violini, Svaki čovek nosi nekoj ženi neku jabuku, Svakoj ženi se jede neka jabuka, itd.

Naravno, proširenjem polaznog rečnika mogu se dobiti razne druge rečenice (elementarne), obrazovane od drugih glagola i drugih imeničkih izraza. Za to je dovoljno u prethodni program dodati odgovarajuće padežne oblike imenica, novih određivača, a za glagole navesti sve njegove odrednice (ukoliko glagol može imati više dužina, za svaku odabranu dužinu navode se odgovarajuće odrednice).

I pored širokih mogućnosti koje su se otvorile pojavom gramatika metamorfoze problemi koji stalno iskrsavaju u vezi sa strogim matematičkim istraživanjima prirodnih jezika su i dalje mnogobrojni. Glavni razlog je razuđenost i bogatstvo prirodnih jezika, kao i njihov neprekidni razvoj. Pomenimo samo kvantore (kolikovnike) koje u ovom članku nismo ni dotakli. To je jedna potpuno nova kategorija koju su uveli matematičari (kvantori su na primer: svaki čovek, mnogi ljudi, obe žene, većina stvari i sl). U vezi sa kvantorima, i sa pitanjima izgradnje jednog strogog jezika koji neće patiti od nedostatka dvosmislenosti (što je, inače glavna mana svih formalnih gramatika), Ričard Montegju (Montague) je početkom 70-tih godina zasnovao jedan potpuno nov pravac u matematičkim istraživanjima prirodnih jezika, poznat danas pod imenom Montegjuove gramatike. Njegova osnovna ideja je bila paralelna izgradnja kako sintakse, tako i semantike prirodnih jezika, pa je u vezi sa tim izgradio jednu novu logiku, takozvanu intenzionalnu, prilagođenu zahtevima prirodnih jezika. No, o tome u novom članku.

LITERATURA

[1] A. Colmerauer, *Les grammaires de metamorphose,* G.I.A., Université d' Aix-Marseillw, November 1975.

[2] Noam Chomsky, *Syntactic Structures,* Mouton, Hague 1957.

[3] M. Gross & A. Lentin, *Notions sur les grammaires formelles,* Gauthier–Villars, Paris 1967

[4] Milka Ivić, *Pravci u lingvistici,* Državna založba Slovenije, Ljubljana 1970

[5] Olga Mišeska Tomić, *Syntax and Syntaxes,* Savremena administracija, Beograd 1987

[6] R. Montague, *The Proper Treatment of Quantification in Ordinary English,* U knjizi J. Hintikka, J. Mooravcsik, and P. Suppes (eds.) Approaches to Natural Languages: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics, D. Reidel Publishing Company Dordrecht 1973

[7] F. Pereira, & D. H. D. Warren, *Definite Clause Grammars for Language Analysis—a Survey of the Formalism and Comparison with Augmented Transition Networks,* Artificial Intelligence 13 (1980), pp. 231–278

[8] A. B. Tucker, *Programming Languages,* McGraw–Hill, New York, 1986

Matematički fakultet
Studentski trg 16
11000 Beograd