0. OSNOVE

— Joseph

Tako to biva Josipe, stari moj,
Kad se osvojilo najljepšu
Među djevojkama Galileje,
Onu koja se zvaše Marija.

Mogao si Josipe, stari moj,
Uzeti Saru ili Deboru
I ništa se ne bi dogodilo,
Ali, odlučio si se za Mariju.

Mogao si Josipe, stari moj,
Ostati doma, tesati svoje drvo
Bolje nego otići u progonstvo
I skrivati se s Marijom.

Mogao si Josipe, stari moj,
Stvoriti djecu s Marijom
I učiti ih svome zanatu
Kao što je tebe tvoj otac učio.

Zašto je trebalo, Josipe,
Da tvoje dijete, to nevinašce,
Prihvaća te tuđe ideje
Koje su izazvale toliko plača kod Marije?

Katkad pomislim na tebe, Josipe,
Prijatelju moj siromašni, kad te se ismijava
Tebe koji se nisi upitao
Koliko je život s Marijom sretan bio.

Joseph

(Gorges MOUSTAKI/
(Zdravko DOVEDAN HAN)

U ovom su poglavlju sumirane definicije formalnih jezika dane u prethodne dvije knjige: definicija jezika, regularnih izraza, gramatika i automata i sintaksne analize. Potom su dani temeljni pojmovi o kompjuterima neophodni za potpunije razumijevanje tema obrađenih u ovoj knjizi, a koje se odnose na jezike za programiranje i njihove prevodioce.

0.1 JEZIK

Znak je jedinstven, nedjeljiv element. Alfabet je konačan skup znakova. Najčešće ćemo ga označivati sa $\mathcal{A}$. Ako se znakovi alfabeta $\mathcal{A}$ poredaju jedan do drugog dobije se niz znakova (engl. *string*) ili "povorka", ili "nizanica". Operacija dopisivanja znaka iza znaka, ili niza iza niza, naziva se nadovezivanje ili konkatenacija nizova.

Duljina niza znakova jest broj znakova sadržanih u njemu. Često se duljina niza znakova x označuje s $d(x)$ ili $|x|$. Niz znakova $aaaa\ldots a$, sačinjen od n jednakih znakova, piše se kao a^n.

Neka su x i y nizovi znakova nad alfabetom $\mathcal{A}$. Kaže se da je x prefiks ("početak"), a y sufiks ("dočetak") niza xy i da je y podniz niza xyz (x kao prefiks i y kao sufiks niza xy istodobno su i njegovi podnizovi). Ako je $x \neq y$ i x je prefiks (sufiks) niza y, kaže se da je x svojstveni prefiks (sufiks) od y.

Definira se i niz znakova koji ne sadrži nijedan znak. Naziva se prazan niz. Označivat ćemo ga s ε ili λ. Za bilo koji niz w vrijedi $\varepsilon w = w\varepsilon = w$. Duljina praznog niza jednaka je 0, $|\varepsilon|=0$, odnosno, ako je a bilo koji znak, vrijedi $a^0=\varepsilon$. Ako je $x=a_1\ldots a_n$ niz, obrnuti niz (ili "reverzni" niz) jest x^R, $x^R=a_n\ldots a_1$. Umjesto x^R može se pisati x^{-1}. Vrijedi $x=(x^{-1})^{-1}$.

Ako je $\mathcal{A}=\{a_1,\ldots,a_n\}$ alfabet i $x\in\mathcal{A}^+$ s $N_{a_i}(x)$ označit ćemo broj pojavljivanja znaka a_i u nizu x. Ako su $\mathcal{A}=\{a_1,a_2,\ldots,a_m\}$ i $\mathcal{B}=\{b_1,b_2,\ldots,b_n\}$ dva alfabeta, definira se njihov produkt:

$$\mathcal{AB}=\{a_ib_j:\ a_i\in\mathcal{A}, b_j\in\mathcal{B}\}$$

a to je skup svih nizova znakova duljine 2 u kojima je prvi znak iz alfabeta $\mathcal{A}$, a drugi iz skupa $\mathcal{B}$. Ako je $\mathcal{A} \neq \mathcal{B}$ tada je i $\mathcal{AB} \neq \mathcal{BA}$ (produkt dvaju alfabeta nije komutativan). Operacija potenciranja alfabeta, $\mathcal{A}^n$, $n\geq 0$, definirana je rekurzivno:

1) $\mathcal{A}^0 = \{\varepsilon\}$
2) $\mathcal{A}^n = \mathcal{AA}^{n-1}$ za $n>0$

Dakle, zaključujemo da će $\mathcal{A}^n$ biti skup svih nizova znakova nad alfabetom duljine n.

Skup svih nizova znakova koji se mogu izgraditi nad alfabetom $\mathcal{A}$, uključujući i prazan niz ε i sam alfabet, označivat ćemo sa $\mathcal{A}^*$. Vrijedi:

$$\mathcal{A}^* = \bigcup_{n=0}^{\infty} \mathcal{A}^n$$

Sa $\mathcal{A}^+$ označivat ćemo skup $\mathcal{A}^* \setminus \{\varepsilon\}$. Primijetiti da su $\mathcal{A}^*$ i $\mathcal{A}^+$ beskonačni ali prebrojivi skupovi! Sa $\mathcal{A}^{*k}$ označivat ćemo konačan skup (podskup od $\mathcal{A}^*$) svih nizova znakova nad $\mathcal{A}$ duljine od 0 do k, a sa $\mathcal{A}^{+k}$ označivat ćemo konačan skup (podskup od $\mathcal{A}^+$) svih nizova znakova nad $\mathcal{A}$ duljine od 1 do k. Unarna operacija * poznata je i pod nazivom Kleeneova zvjezdica, jer ju je prvi put definirao Stephen Kleene. Operacija + je Kleeneov plus.

Ako je $\mathcal{A}$ alfabet i $\mathcal{A}^*$ skup svih nizova znakova nad $\mathcal{A}$, jezik $\mathcal{L}$ nad alfabetom $\mathcal{A}$ jest bilo koji podskup od $\mathcal{A}^*$, tj.

$$\mathcal{L} \subseteq \mathcal{A}^*$$

Često se piše $\mathcal{L}(\mathcal{A})$ da se naznači definiranost nekog jezika $\mathcal{L}$ nad alfabetom $\mathcal{A}$. Nizove znakova koji čine elemente jezika nazivamo rečenice. Za jezik $\mathcal{L}$ u kojem za sve njegove rečenice w vrijedi da nisu svojstveni prefiks (sufiks) ni jednoj rečenici x, $x \in \mathcal{L}$ i $x \neq w$, kaže se da ima svojstvo prefiksa (sufiksa).

Operacije nad jezicima

S obzirom na to da je jezik skup, primjenom poznatih operacija nad skupovima mogu se iz definiranih graditi novi jezici. Elementi jezika su nizovi znakova pa se može definirati i operacija nadovezivanja.

Ako su $\mathcal{L}_1$ i $\mathcal{L}_2$ jezici, $\mathcal{L}_1 \subseteq \mathcal{A}_1^*$ i $\mathcal{L}_2 \subseteq \mathcal{A}_2^*$, tada je $\mathcal{L}_1\mathcal{L}_2$ nadovezivanje (ulančavanje ili konkatenacija) ili produkt jezika $\mathcal{L}_1$ i $\mathcal{L}_2$:

$$\mathcal{L}_1\mathcal{L}_2 = \{xy:\ x \in \mathcal{L}_1,\ y \in \mathcal{L}_2\}$$

Simboli i nizovi simbola

Često se promatraju nizovi znakova konačne duljine koji se mogu smatrati jedinstvenom, nedjeljivom cjelinom. Takvi nizovi znakova nazivaju se simboli ili riječi.

Skup svih simbola definiran nad alfabetom $\mathcal{A}$ označivat ćemo s $\mathcal{V}$ i nazivati rječnik. To je uvijek konačan skup. Budući je $\mathcal{V} \subseteq \mathcal{A}^*$, zaključujemo da je $\mathcal{V}$ jezik. Definira se i jezik nad rječnikom, tj. $\mathcal{L}_{\mathcal{V}} \subseteq \mathcal{V}^*$. Rečenice takvog jezika i dalje su nizovi znakova iz $\mathcal{A}^*$, ali se mogu promatrati i kao nizovi simbola rječnika $\mathcal{V}$.

Klasifikacija jezika

Prema hijerarhiji Chomskog jezike klasificiramo u četiri skupine (ili tipa), kao što je prikazano u sljedećoj tablici:

tip	naziv
0	bez ograničenja
1	kontekstni
2	beskontekstan
3	linearan

Da bismo odredili kojoj klasi jezika pripada neka rečenica, korisno je znati svojstvo napuhavanja ("pumping lemma") koje kaže da svaki jezik dane klase može biti "napuhan" i pri-tom još uvijek pripadati danoj klasi. Jezik može biti napuhan ukoliko se dovoljno dugi niz znakova jezika može rastaviti na podnizove, od kojih neki mogu biti ponavljani proizvoljan broj puta u svrhu stvaranja novog, duljeg niza znakova koji je još uvijek u istom jeziku. Stoga, ukoliko vrijedi svojstvo napuhavanja za danu klasu jezika, bilo koji neprazni jezik u klasi će sadržavati beskonačan skup konačnih nizova znakova izgrađenih jednostavnim pravilom koje daje svojstvo napuhavanja.

Regularni skupovi

Neka je $\mathcal{A}$ alfabet. Regularni skup (regularni jezik) nad $\mathcal{A}$ definiran je rekurzivno na sljedeći način:

1) $\varnothing$ (prazan skup) je regularni skup nad $\mathcal{A}$.
2) $\{\varepsilon\}$ je regularni skup nad $\mathcal{A}$.
3) $\{\mathtt{a}\}$ je regularni skup nad $\mathcal{A}$, za sve $\mathtt{a}$ iz $\mathcal{A}$.
4) Ako su P i Q regularni skupovi nad $\mathcal{A}$, regularni skupovi su i:

   a) $P \cup Q$ b) PQ c) P^* d) (P)

Dakle, podskup od $\mathcal{A}^*$ jest regularan ako i samo ako je $\varnothing$, $\{\varepsilon\}$ ili $\{\mathtt{a}\}$, za neki $\mathtt{a}$ u $\mathcal{A}$, ili se može dobiti iz njih konačnim brojem primjena operacija unije, produkta i (Kleeneove) operacije *. Izraz s regularnim skupovima može imati i zagrade da bi se naznačio prioritet izvršavanja ove tri operacije, pa najviši prioritet ima podizraz unutar zagrada, potom Kleenova operacija (potenciranje), produkt i, na kraju, unija.

SVOJSTVA REGULARNIH SKUPOVA

Sada ćemo dati jedno svojstvo regularnih skupova čime će biti određeno je li dani skup regularni. To je "svojstvo napuhavanja", a definirano je u slijedećoj lemi napuhavanja regularnih skupova: Neka je $\mathcal{R}$ regularni skup. Tada postoji konstanta k takva da ako je $w \in \mathcal{R}$ i $|w| \geq k$, tada se w može napisati kao xyz, gdje je $0 < |y| \leq k$ i $xy^iz \in \mathcal{R}$ za sve $i \geq 0$.

Lema napuhavanja definira osnovno svojstvo nizova regularnog skupa da svi proizvoljno dugi nizovi (rečenice) regularnog jezika mogu biti "napuhane", tj. postoji središnji dio niza koji se ponavlja proizvoljan broj puta da bi proizveo novi niz koji je u istom jeziku. U praksi se lema napuhavanja često koristi da bi se dokazalo da dani jezik nije regularan.

0.2 REGULARNI IZRAZI

Regularni izrazi (još i "pravilni izrazi") na alfabetu $\mathcal{A}$ označuju (generiraju) određene regularne skupove. Regularni izraz definira se rekurzivno, na sljedeći način:

(1) $\varnothing$ je regularni izraz koji označuje regularni skup $\varnothing$.
(2) ε je regularni izraz koji označuje regularni skup $\{\varepsilon\}$.
(3) $\mathtt{a}$ iz $\mathcal{A}$ je regularni izraz koji označuje regularni skup $\{\mathtt{a}\}$.
(4) Ako su p i q regularni izrazi koji označuju regularne skupove P i Q, redom, tada su:

(a) $(p+q)$ ili $(p|q)$ regularni izraz koji označuje regularni skup $P \cup Q$.
(b) (pq) regularni izraz koji označuje regularni skup PQ.
(c) $(p)^*$ regularni izraz koji označuje regularni skup P^*.

Operaciju "+" ili "|" čitamo "ili". Za regularni izraz p^* koristit ćemo i notaciju $\{p\}$ (što ne treba poistovjećivati sa skupom). Ako je pp^* regularni izraz, može se napisati kao p^+ ili $p\{p\}$. Također ćemo koristiti i notaciju $\{p\}_m^n$ koja ima značenje dopisivanje izraza p najmanje m, najviše n puta, $m \le n$. Izostavljeno m ima značenje 0, a izostavljeno n ima značenje *. Ako ne postoji dvoznačnost u nekom regularnom izrazu, suvišne se zagrade mogu izbaciti. Može se zamisliti da operacija * (i $^+$) ima najviši proritet, potom operacija nadovezivanja i, na kraju, operacija + (ili |).

Algebarska svojstva regularnih izraza

Reći ćemo da su dva regularna izraza jednaka (=) ako označuju isti regularni skup. Ako su α, β i γ regularni izrazi, tada vrijede sljedeća algebarska svojstva:

1) $\alpha+\beta = \beta+\alpha$	2) $\alpha+(\beta+\gamma) = (\alpha+\beta)+\gamma$	3) $\alpha(\beta\gamma) = (\alpha\beta)\gamma$
4) $\alpha(\beta+\gamma) = \alpha\beta+\alpha\gamma$	5) $(\alpha+\beta)\gamma = \alpha\gamma+\beta\gamma$	6) $\alpha\varepsilon = \varepsilon\alpha = \alpha$
7) $\alpha^* = \alpha+\alpha^*$	8) $(\alpha^*)^* = \alpha^*$	9) $\alpha+\alpha = \alpha$

0.3 GRAMATIKE

Gramatika je četvorka $\mathcal{G}=(\mathcal{N}, \mathcal{T}, \mathcal{P}, \mathcal{S})$, gdje su:

$\mathcal{N}$ konačan skup neterminalnih znakova,

$\mathcal{T}$ konačan skup terminalnih znakova (alfabet) uz uvjet da je

$$\mathcal{T} \cap \mathcal{N} = \emptyset$$

$\mathcal{P}$ konačan skup parova nizova:

$$\{(\alpha, \beta):\ \alpha=\alpha_1\gamma\alpha_2;\ \alpha_1, \alpha_2, \beta \in (\mathcal{N} \cup \mathcal{T})^*,\ \gamma \in \mathcal{N}\}$$

(niz α je iz $(\mathcal{N} \cup \mathcal{T})^+$ i mora sadržati bar jedan znak iz skupa $\mathcal{N}$),

$\mathcal{S}$ poseban znak iz $\mathcal{N}$, $\mathcal{S} \in \mathcal{N}$, nazvan početni znak (ili *početni simbol*).

Element (α, β) iz $\mathcal{P}$ piše se $\alpha \to \beta$ i naziva produkcija. Simbol "$\to$" čita se "producira", "može biti zamijenjeno s" ili "preobličuje se u". Ako $\mathcal{P}$ u nekoj gramatici sadrži produkcije:

$$\alpha \to \beta_1 \quad \ldots \quad \alpha \to \beta_n$$

piše se

$$\alpha \to \beta_1 | \ \beta_2 | \ldots \ | \beta_n$$

Znak "|" čita se "ili". β_i su alternative za α. Ako u $\mathcal{P}$ postoji produkcija oblika

$$\alpha \to \varepsilon | \ \beta | \ \beta\beta | \ \beta\beta\beta | \ \ldots$$

piše se $\alpha \rightarrow \{\beta\}$. Vitičaste zagrade omeđuju niz koji može biti izostavljen ili napisan jedanput, dvaput, triput, itd. Produkcija oblika:

$$\alpha \rightarrow \varepsilon \mid \beta$$

piše se $\alpha \rightarrow [\beta]$. Dakle, uglate zagrade omeđuju niz koji može biti izostavljen ili napisan jedanput. U daljnjem ćemo tekstu neterminale označivati velikim slovima engl. abecede. Terminali će biti mala slova engl. abecede i ostali znakovi (brojke, +, -, *, /, (,), itd.). Neterminal na početku prve produkcije bit će početni simbol.

Gramatika kao generator jezika

Rečenična forma gramatike $\mathcal{G}=(\mathcal{N}, \mathcal{T}, \mathcal{P}, \mathcal{S})$ definirana je rekurzivno:

1) Početni znak je rečenična forma.
2) Ako je $\alpha\delta\gamma$, gdje su $\alpha,\gamma \in (\mathcal{N} \cup \mathcal{T})^*$, rečenična forma i $\delta \rightarrow \beta$ produkcija u $\mathcal{P}$, tada je $\alpha\beta\gamma$ također rečenična forma.

Rečenična forma koja ne sadrži nijedan neterminal naziva se rečenica. Nad skupom $(\mathcal{N} \cup \mathcal{T})^*$ gramatike $\mathcal{G}=(\mathcal{N}, \mathcal{T}, \mathcal{P}, \mathcal{S})$ definira se relacija $\Rightarrow$, čita se izravno izvodi, na sljedeći način: Ako je $\alpha\delta\gamma$ niz iz $(\mathcal{N} \cup \mathcal{T})^*$ i $\delta \rightarrow \beta$ produkcija iz $\mathcal{P}$, tada

$$\alpha\delta\gamma \Rightarrow \alpha\beta\gamma$$

Ako za $\alpha_0, \alpha_1, \ldots, \alpha_n, \alpha_i \in (\mathcal{N} \cup \mathcal{T})^*, n \geq 1$, vrijedi

$$\alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_n$$

tada je $\alpha_0 \overset{n}{\Rightarrow} \alpha_n$ niz izvođenja duljine n. Općenito se piše

$$\alpha_0 \overset{*}{\Rightarrow} \alpha_n, n \geq 0, \alpha_0 \overset{+}{\Rightarrow} \alpha_n, n > 0$$

i kaže da α_0 izvodi α_n.

Shodno dvjema prethodnim definicijama jezik generiran gramatikom $\mathcal{G}$ može se napisati kao

$$\boxed{\mathcal{L}(\mathcal{G}) = \{\omega \in \mathcal{T}^* : \mathcal{S}^* \Rightarrow \omega\}}$$

što čitamo: "Jezik $\mathcal{L}$ generiran gramatikom $\mathcal{G}$ jest skup rečenica dobivenih nizom svih mogućih izvođenja krenuvši od početnog simbola $\mathcal{S}$".

Klasifikacija gramatika

Gramatike se mogu klasificirati prema obliku svojih produkcija. Za gramatiku $\mathcal{G}=(\mathcal{N}, \mathcal{T}, \mathcal{P}, \mathcal{S})$ kaže se da je:

1) Tipa 3 ili linearna zdesna ako je svaka produkcija iz $\mathcal{P}$ oblika

$$A \rightarrow xB \text{ ili } A \rightarrow x \qquad A, B \in \mathcal{N},\ x \in \mathcal{T}^*$$

ili linearna slijeva ako je svaka produkcija iz $\mathcal{P}$ oblika

$A \rightarrow Bx$ ili $A \rightarrow x$ $\quad A, B \in \mathcal{N},\ x \in \mathcal{T}^*$

Gramatika linearna zdesna naziva se regularna gramatika ako je svaka produkcija oblika

$A \rightarrow aB$ ili $A \rightarrow a$ $\quad A, B \in \mathcal{N},\ a \in \mathcal{T}$

i jedino je dopuštena produkcija $\mathcal{S} \rightarrow \varepsilon$, ali se tada $\mathcal{S}$ ne smije pojavljivati niti u jednoj alternativi ostalih produkcija.

2) Tipa 2 ili beskontekstna ako je svaka produkcija iz $\mathcal{P}$ oblika:

$A \rightarrow \alpha$ $\quad A \in \mathcal{N},\ \alpha \in (\mathcal{N} \cup \mathcal{T})^*$

3) Tipa 1 ili kontekstna ako je svaka produkcija iz $\mathcal{P}$ oblika

$\alpha \rightarrow \beta$ $\quad$ uz uvjet da je $|\alpha| \leq |\beta|$

4) Bez ograničenja ili tipa 0 ako produkcije ne zadovoljavaju nijedno od navedenih ograničenja.

Sada možemo reći da je jezik bez ograničenja ako je generiran gramatikom tipa 0, kontekstan ako je generiran gramatikom tipa 1, beskontekstan ako je generiran gramatikom tipa 2 i linearan (ili regularan) ako je generiran gramatikom tipa 3 (ili regularnom gramatikom). Četiri tipa gramatika i jezika uvedenih prethodnom definicijom nazivaju se hijerarhija Chomskog.

Svaka regularna gramatika istodobno je beskontekstna, beskontekstna bez ε–produkcija je kontekstna i, konačno, kontekstna gramatika je istodobno gramatika bez ograničenja. Ako s $\mathcal{L}_i$ označimo jezik tipa i, vrijedi $\mathcal{L}_{i+1} \subseteq \mathcal{L}_i$, $0 \leq i < 3$. Regularne gramatike generiraju najjednostavnije jezike koji mogu biti generirani regularnim izrazima.

Prikaz gramatika

U prethodnim definicijama i primjerima razlikovali smo neterminalne i terminalne simbole prema vrsti znakova: velika slova bila su rezervirana za neterminale, a mala slova i ostali znakovi za terminale. Takvim dogovorom nije bilo neophodno uvijek posebno navoditi skupove neterminala i terminala. Bilo je dovoljno napisati produkcije i zadati početni simbol. Na taj način zadana je sintaksa jezika. Dva su najčešća načina prikaza gramatika: Backus-Naurovom formom i sintaksnim dijagramima.

BACKUS-NAUROVA FORMA (BNF)

Formalizam pisanja produkcija (ili "pravila zamjenjivanja") kojim smo dosad zadavali produkcije gramatike modifikacija je formalizma poznatog kao Backus-Naurova forma ili BNF. Prvi je put bio primijenjen u definiciji jezika ALGOL 60, 1963. godine i još uvijek je prisutan u praksi. Piše se prema sljedećim pravilima:

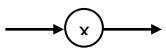
1) Neterminalni simboli pišu se između znakova "<" i ">".
2) Umjesto "→" koristi se simbol "::=" i čita "definirano je kao".

Pišući neterminale između znakova "<" i ">" moguće je izborom njihovih imena uvesti "značenje" u produkcije, jer će nas imena podsjećati na vrstu rečenica koja će se generirati u nekom podjeziku.

SINTAKSNI DIJAGRAMI

Produkcije gramatike $\mathcal{G}$ mogu biti prikazane i u obliku koji se naziva sintaksni dijagram. Sve je veća prisutnost sintaksnih dijagrama u novijoj literaturi, prije svega što se njihovom uporabom bolje uočava struktura jezika. Pravila konstruiranja sintaksnih dijagrama su sljedeća:

1) Terminalni simbol x prikazan je kao

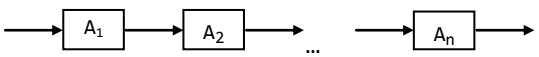


2) Neterminalni simbol A prikazan je kao

3) Produkcija oblika

$A \rightarrow A_1 A_2 \ldots A_n \qquad A_i \in (\mathcal{N} \cup \mathcal{T})$

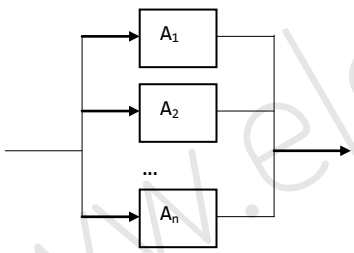
predstavlja se dijagramom



4) Produkcija oblika

$A \rightarrow A_1 | A_2 | \ldots | A_n \qquad A_i \in (\mathcal{N} \cup \mathcal{T})$

prikazuje se dijagramom

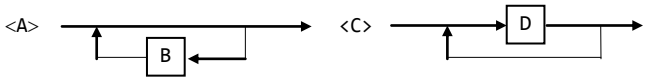


gdje je svaki A_i prikazan prema pravilima od (1) do (4). Ako je $A_i = \varepsilon$, ta se alternativa prikazuje punom crtom.

5) Produkcije oblika

$A \rightarrow \{B\}$ i $C \rightarrow [D] \qquad B, D \in (\mathcal{N} \cup \mathcal{T})$

prikazuju se dijagramima:



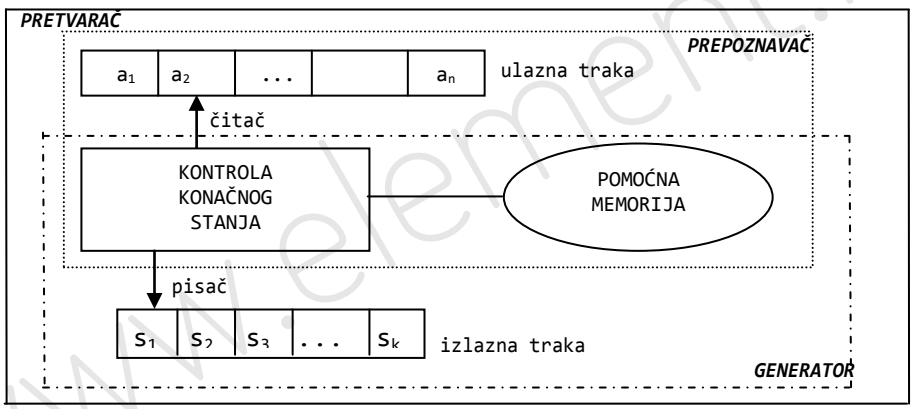
gdje su B i D prikazani dijagramima prema pravilima (1) do (4).

Često se u praksi pojednostavljuje pisanje sintaksnih dijagrama, posebno ako je nedvojbena razlika u pisanju terminala i neterminala. Na primjer, neterminali su riječi napisane malim slovima, a terminali su riječi napisane velikim slovima ili su to brojevi i ostali znakovi.

0.4 AUTOMATI

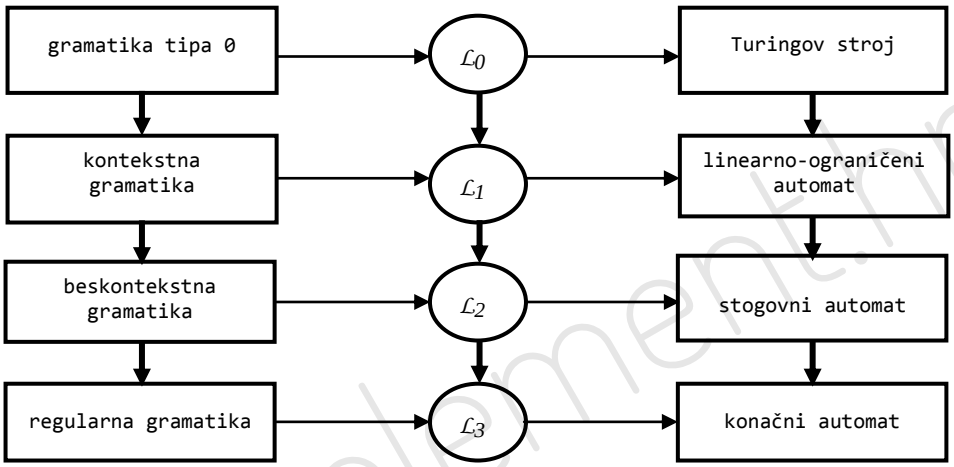
Osim gramatika, uvodimo automate kao važnu klasu generatora, prepoznavača i prevodilaca jezika, posebno pogodnih za implementaciju na računalima. Teorija je konačnih automata koristan instrument za razvoj sustava s konačnim brojem stanja čije mnogobrojne primjene nalazimo i u informatici. Programi, kao što je npr. tekstovni editor, često su načinjeni kao sustavi s konačnim brojem stanja. Na primjer, računalo se zasebno također može promatrati kao sustav s konačno mnogo stanja. Upravljačka jedinica, memorija i vanjska memorija nalaze se teoretski u svakom trenutku u jednom od vrlo velikog broja stanja, ali još uvijek u konačnom skupu stanja. Iz svakodnevnog je života upravljački mehanizam dizala još jedan dobar primjer sustava s konačno mnogo stanja. Prirodnost koncepta sustava s konačno mnogo stanja je razlog primjene tih sustava u različitim područjima, pa je i to vjerojatno najvažniji razlog njihova proučavanja.

Opći model automata dan je na sl. 0.1. Automat koji sadrži sve navedene "dijelove" naziva se pretvarač, automat bez ulazne trake je generator, a automat bez izlazne trake je prepoznavač.



Sl. 0.1 – *Opći model automata.*

Automati najčešće imaju ulogu prepoznavača. Ovisno o jeziku koji se prepoznaje, odnosno o tipu gramatike koja generira takav jezik, postoje i vrste prepoznavača dane na sljedećem crtežu.



Sl. 0.2 - *Chomskyjeva hijerarhija gramatika, njihovi odgovarajući jezici i prepoznavači.*

Konačni automat

Konačni automat nema pomoćne memorije. To je matematički model sustava koji se nalazi u jednom od mnogih konačnih stanja. Stanje sustava sadrži obavijesti koje su dobivene na temelju dotadašnjih podataka i koje su potrebne da bi se odredila reakcija sustava iz idućih podataka. Drugim riječima, radi se o prijelaznim stanjima koje izvodi dani ulazni znak iz danog alfabeta Σ pod utjecajem funkcije prijelaza. Svaki se ulazni znak može nalaziti samo u jednom prijelaznom stanju, pri čemu je dopušten povratak na prethodno stanje.

Konačni automat je uređena petorka $\mathcal{M}=(Q, \Sigma, \delta, q_0, \mathcal{F})$, gdje su:

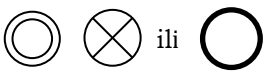
- Q konačni skup stanja
- Σ alfabet
- δ funkcija prijelaza, definirana kao $\delta: Q\times\Sigma\rightarrow\mathcal{P}(Q)$ gdje je $\mathcal{P}(Q)$ particija od Q
- q_0 početno stanje, $q_0\in Q$
- $\mathcal{F}$ skup završnih stanja, $\mathcal{F}\subseteq Q$

DIJAGRAM PRIJELAZA

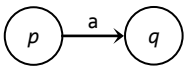
Iz definicije funkcije prijelaza zaključujemo da je to označeni, usmjereni graf čiji čvorovi odgovaraju stanjima automata, a grane su označene znakovima iz alfabeta. Ako, dakle, funkciju prijelaza prikažemo kao graf, dobivamo dijagram prijelaza. Označimo li u tom dijagramu početno stanje i skup završnih stanja, dobivamo konačni automat prikazan grafički. Početno ćemo stanje označivati s:


ili

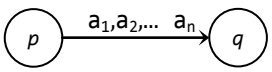
a završno s:


ili

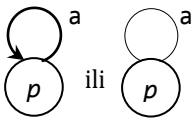
Ako je $p=\delta(q,a)$, tada postoji prijelaz iz stanja q u stanje p pa će grana u dijagramu prijelaza koja spaja q i p, s početkom u q i završetkom u p, biti označena s a:



Ako postoji više grana s početkom u q i završetkom u p, tj. ako je $p=\delta(q,a_1)=...=\delta(q,a_n)$, to će biti prikazano s:



Povratak nekim prijelazom a u isto stanje, tj. ako je $p=\delta(p,a)$, dijagram prijelaza je:



TABLICA PRIJELAZA

Funkcija prijelaza može se prikazati tablično. Tada se kaže da je to tablica prijelaza. Redovi tablice predstavljaju stanja, a stupci su označeni znakovima iz alfabeta i predstavljaju prijelaze. Na mjestu u redu označenom s q, $q\in Q$, i stupcu označenom s x, $x\in\Sigma$, upisan je skup narednih stanja (bez vitičastih zagrada) ako je funkcija $\delta(q, x)$ definirana, odnosno nije ništa upisano ako $\delta(q, x)$ nije definirano. Početno ćemo stanje označiti s $\rightarrow$ ili >, a konačno s $\otimes$ ili *.

DETERMINISTIČKI I NEDETERMINISTIČKI AUTOMAT

Iz definicije funkcije prijelaza vidimo da je moguć prijelaz iz tekućeg stanja u više narednih stanja s istim prijelazom a, $a\in\Sigma$, tj. da je ponašanje automata općenito nedeterminističko. Neka je $\mathcal{M}=(Q, \Sigma, \delta, q_0, \mathcal{F})$ konačni automat. Kažemo da je automat deterministički ako za sve $a\in\Sigma$ i sva stanja q, q' i q'' iz $\mathcal{F}$ i

$$\delta(q, a) = \{q', q''\}$$

slijedi da je $q'=q''$. Ako postoji barem jedan $a\in\Sigma$ tako da je $q'\neq q''$, automat je nedeterministički.

Stogovni automat

Stogovni automat PDA (Push Down Automaton) jest uređena sedmorka, $\mathcal{P}=(Q, \Sigma, \Gamma, \delta, q_0, Z_0, \mathcal{F})$, gdje su:

- Q konačan skup stanja (kontrole konačnog stanja)
- Σ ulazni alfabet
- Γ alfabet znakova stoga (potisne liste)
- δ funkcija prijelaza, definirana kao $\delta: Q\times(\Sigma\cup\{\varepsilon\})\times\Gamma \rightarrow Q\times\Gamma^*$

q_0 početno stanje, $q_0 \in Q$
Z_0 početni znak stoga (potisne liste), $Z_0 \in \Gamma$
$\mathcal{F}$ skup završnih stanja, $\mathcal{F} \subseteq Q$

Dvostruko-stogovni automat

Dvostruko-stogovni automat definiran je kao uređena sedmorka

$\mathcal{P}t = (Q, \Sigma, \Gamma_1, \Gamma_2, \Delta, s, \mathcal{F})$

gdje su:

Q konačan skup stanja (kontrole konačnog stanja)
Σ ulazni alfabet
Γ_1, Γ_2 alfabet prvog i drugog stoga
Δ funkcija prijelaza, definirana kao $\Delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma_1 \times \Gamma_2 \rightarrow Q \times \Gamma_1^* \times \Gamma_2^*$
s početno stanje, $s \in Q$
$\mathcal{F}$ skup konačnih stanja, $\mathcal{F} \subseteq Q$

Vidimo da se ovaj automat razlikuje od stogovnog automata u definiciji funkcije prijelaza Δ koja koristi dva stoga kao pomoćnu memoriju. Kaže se da je $\mathcal{P}t$ linearno-ograničen jer je duljina oba stoga linearno proporcionalna duljini generiranog niza.

0.5 PARSIRANJE

Jezik generiran gramatikom jest

$\mathcal{L}(\mathcal{G}) = \{w \in \mathcal{T}^*: \mathcal{S} \Rightarrow^* w\}$

a to je skup rečenica w dobivenih svim mogućim izvođenjima iz $\mathcal{S}$. U praksi, poslije izvođenja gramatike nekog jezika i njezine verifikacije, dalje će njezina uporaba biti u provjeri je li dani niz w rečenica jezika $\mathcal{L}(\mathcal{G})$. Tada se problem svodi na nalaženje niza izvođenja, počevši od $\mathcal{S}$, koji bi rezultirao tim nizom (rečenicom). Takav se postupak sintaksne analize naziva parsiranje. Ovdje treba napomenuti da je termin "parsiranje" izravni kalk iz engleskog "parsing". Primjerenije bi bilo "posuditi" ga iz latinskog, pa je to parsanje, kako je prije nekoliko godina predložio prof. dr. sc. Marko Tadić. No, s obzirom na to da smo u prethodnoj knjizi rabili termin "parsiranje", zadržat ćemo ga i ovdje. Ustrojbu postupka parsiranja na računalu (program u izabranom jeziku za programiranje) nazivat ćemo parser. Stablo izvođenja dobiveno iz niza izvođenja $\mathcal{S} \Rightarrow^* w$ sada ćemo zvati stablo sintaksne analize.

Lijevo i desno parsiranje

Neka je $\mathcal{G}=(\mathcal{N},\mathcal{T},\mathcal{P},\mathcal{S})$ beskontekstna gramatika u kojoj su produkcije iz $\mathcal{P}$ označene (numerirane) s 1,2,...,p i neka je $\alpha \in (\mathcal{N} \cup \mathcal{T})^*$. Tada je

(1) lijevo parsiranje za α niz produkcija rabljenih u krajnjem izvođenju slijeva krenuvši od $\mathcal{S}$: $\mathcal{S} \Rightarrow_{lm}^* \alpha$
(2) desno parsiranje za α reverzni niz produkcija rabljenih u krajnjem izvođenju zdesna krenuvši od $\mathcal{S}$: $\mathcal{S} \Rightarrow_{rm}^* \alpha$

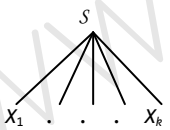
Ako i predstavlja i-tu produkciju, pisat ćemo $\alpha \Rightarrow_i \beta$ ako je $\alpha \Rightarrow_{lm} \beta$, odnosno $\alpha \Rightarrow_i \beta$ ako je $\alpha \Rightarrow_{rm} \beta$. Također ćemo pisati $\alpha \Rightarrow_{i_1 i_2} \gamma$ ako je $\alpha \Rightarrow_{i_1} \beta$ i $\beta \Rightarrow_{i_2} \gamma$, odnosno $\alpha \Rightarrow_{i_1 i_2} \gamma$ ako je $\alpha \Rightarrow_{i_1} \beta$ i $\beta \Rightarrow_{i_2} \gamma$.

Lijevo i desno parsiranje primjenljivi su samo nad beskontekstnim (i linearnim) jezicima. U lijevom parsiranju stablo sintaksne analize izvodi se s vrha ili silazno, od korijena prema listovima (top-down). U desnom parsiranju stablo sintaksne analize izvodi se odozdo prema vrhu ili uzlazno, od listova prema korijenu (bottom-up).

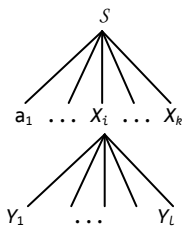
Silazna sintaksna analiza

Znajući lijevo parsiranje $\pi = i_1 i_2 \ldots i_n$ rečenice $w = a_1 a_2 \ldots a_n$ iz $\mathcal{L}(\mathcal{G})$ može se izvesti stablo parsiranja (sintaksne analize) u značenju "s vrha" ili silazno, pa kažemo da je to <u>silazna sintaksna analiza</u>. Počinje se s korijenom stabla, označenim sa S. Parsiranje i_1 daje produkciju koja je upotrijebljena u ekspanziji stabla. Pretpostavimo da i_1 označuje produkciju $S \rightarrow X_1 \ldots X_k$ pa se može kreirati k sljedbenika iz čvora S koji su označeni s $X_1, X_2, \ldots, X_k$

$S \rightarrow X_1 \ldots X_k$



Ako su $X_1, X_2, \ldots, X_{i-1}$ terminali, tada prvih i-1 simbola (znakova) ulaznog niza w moraju biti $X_1 X_2 \ldots X_{i-1}$. Produkcija i_2 mora biti oblika $X_i \rightarrow Y_1 \ldots Y_l$ pa se nastavlja ekspanzija stabla iz čvora X_i:



Postupak se nastavlja ekspanzijom na opisani način sve dok se ne izgradi stablo čiji će li-stovi biti znakovi niza w. Na primjer, neka je $\mathcal{G}_E$ gramatika s numeriranim produkcijama

(1) E → E+T (2) E → T (3) T → T*F (4) T → F (5) F → (E) (6) F → a

Lijevo parsiranje rečenice a*(a+a) je 23465124646, što je dobiveno iz niza izvođenja slijeva:

$$E \Rightarrow_2 T \Rightarrow_3 T*F \Rightarrow_4 F*F \Rightarrow_6 a*F \Rightarrow_5 a*(E) \Rightarrow_1 a*(E+T) \Rightarrow_2 a*(T+T) \Rightarrow_4 a*(F+T) \Rightarrow_6 a*(a+T) \Rightarrow_4 a*(a+F) \Rightarrow_6 a*(a+a)$$

što se može napisati kao

$$E \Rightarrow_{23465124646} a*(a+a)$$