Noțiuni de bază în limbaje formale

Curs 2

Exemple de limbaje

- naturale (ex. Engleza, română)
- de programare (ex. C,C++)
- formale

- Exemplu: Un copil are un câine.

S→PV
P→ *un* N
N→ *copil* sau N→ *caine*
(N→*copil*|*caine*)
V → QC
Q → *are*
C → AN
A → *un* | *o*

- A→ α = **regula**
- S,P,V,N,Q,C,A = **simboluri neterminale**
- *o,un,femeie,caine,are* = **simboluri terminale**

Observatii

1. Propozitia = cuvant sau secventa (contine doar simboluri terminale) si se va nota cu w.
2. S⇒PV⇒o NV⇒o NQC⇒o N are C -forma propozitionala

   În general : $w=a_1a_2...a_n$

1. Regula de mai sus asigura corectitudinea sintaxei,dar NU asigura corectitudinea semantică

Gramatică

- ***Definiție***: O **gramatica** (formala) este un cvadruplu : $G=(N,\Sigma,P,S)$ avand urmatoarele semnificatii:
  - N - multimea simbolurilor neterminale si $|N| < \infty$
  - Σ - multimea simbolurilor terminale (alfabetul) si $|\Sigma|$¡∞
  - P - o multime finita,reprezentand multimea productiilor, avand proprietatea: $P \subseteq (N \cup \Sigma)^* N(N \cup \Sigma)^* X(N \cup \Sigma)^*$
  - $S \in N$ - simbol de start/axioma

Observații :

1. $(\alpha,\beta) \in P$ este o productie notată $\alpha \rightarrow \beta$
2. $N \cap \Sigma = \varnothing$

Relații binare definite pe $(N \cup \Sigma)^*$

- ***derivare directă***

  $\alpha \Rightarrow \beta$, $\alpha,\beta \in (N \cup \Sigma)^*$ dacă $\alpha=x1xy1$, $\beta=x1yy1$ si $x \rightarrow y \in P$

  (x este transformat în y)

- ***k derivare***

$\alpha \overset{k}{\Rightarrow} \beta$,$\alpha,\beta \in (N \cup \Sigma)^*$

secventa de k derivari directe $\alpha \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_{k-1} \Rightarrow \beta$, $\alpha,\alpha_1,\alpha_2,\ldots\alpha_{k-1},\beta \in (N \cup \Sigma)^*$

- ***+ derivare***

$\alpha \overset{+}{\Rightarrow} \beta$ daca exista k>0 astfel incat $\alpha \overset{k}{\Rightarrow} \beta$ (exista cel putin o derivare directa)

- **** derivare***

$\alpha \overset{*}{\Rightarrow} \beta$ daca exista k≥0 astfel incat $\alpha \overset{k}{\Rightarrow} \beta$ adică, $\alpha \overset{*}{\Rightarrow} \beta \Leftrightarrow \alpha \overset{+}{\Rightarrow} \beta$ SAU $\alpha \overset{0}{\Rightarrow} \beta$ $(\alpha=\beta)$

Definitie: ***Limbajul generat*** de o gramatica G=(N,Σ,P,S) este:

$$L(G)=\{w\in\Sigma^* \mid S \overset{*}{\Rightarrow} w\}$$

Observatii

1. $S\overset{*}{\Rightarrow} \alpha, \alpha\in(N\cup\Sigma)^*$ = formă propozițională
   $S\overset{*}{\Rightarrow} w, w\in\Sigma^*$ = cuvânt / secvență
2. Operatiile definite pentru limbaje (mulțime) :
   $L1\cup L2$, $L1\cap L2$, L1-L2 , $\overline{L}$ (complementara) , $L^+=\bigcup_{k>0} L^k$, $L^*=\bigcup_{k\geq 0} L^k$
   Concatenarea: $L=L_1L_2 = \{w_1w_2 \mid w_1\in L_1 , w_2\in L_2\}$
3. $|w|=0$ (cuvântul vid - notatie ε)

Definitie : Doua gramatici G_1 si G2 sunt echivalente daca ele genereaza acelasi limbaj

$$L(G_1)=L(G_2)$$

Ierarhia lui Chomsky (bazată pe forma $\alpha \to \beta \in P$)

- tipul 0 : nici o restrictie
- tipul 1 : gramatici dependente de context ($x1Ay1 \to x1\gamma y1$)
- tipul 2 : gramatici independente de context ($A \to \alpha \in P$,unde $A \in N$ si $\alpha \in (N \cup \Sigma)^*$)
- tipul 3 : gramatici regulare ($A \to aB|a \in P$)

Observație :

tipul 3 $\subseteq$ tipul 2 $\subseteq$ tipul 1 $\subseteq$ tipul 0

Notații

- $A,B,C,...$ - pentru simboluri neterminale
- $S \in N$ - pentru simbolul de start
- $a,b,c,... \in \Sigma$ - pentru simboluri terminale
- $\alpha,\beta,\gamma \in (N \cup \Sigma)^*$ - pentru forme propozitionale
- ε - pentru cuvântul vid
- $x,y,z,w \in \Sigma^*$ - pentru cuvinte
- $X,Y,U,... \in (N \cup \Sigma)$ - pentru simboluri din gramatică (neterminal sau terminal)

Automate finite

Definiție: Un ***automat finit (AF)*** este un 5-tuplu

$$M = (Q,\Sigma,\delta,q0,F)$$

unde:

- Q este o multime finita de stari ($|Q|<\infty$)
- Σ este un alfabet finit ($|\Sigma|<\infty$)
- δ este o functie de tranzitie : $\delta:Q\times\Sigma\rightarrow P(Q)$
- q_0 reprezinta starea initiala a automatului finit $q_0 \in Q$
- $F\subseteq Q$ reprezinta multimea starilor finale

Observații

1. $Q \cap \Sigma = \varnothing$
2. $\delta: Q \times \Sigma \rightarrow P(Q)$, $\varepsilon \in \Sigma^0$ - relatia $\delta(q,\varepsilon)=p$ **nu** este permisa
3. Daca $|\delta(q,a)| \leq 1$ => un automat finit determinist (AFD)
4. Daca $|\delta(q,a)| > 1$ (mai mult de o stare e obtinuta ca rezultat) => automat finit nedeterminist (AFN)

Proprietate: Pentru orice AFN M exista un AFD M' echivalent cu M

Configurație C=(q,x)

Unde:

- q este stare
- x este o secvență necitita de pe banda de intrare: $x \in \Sigma^*$

Configurația inițială : (q_0,w) , w - întreaga secventa

Configurarea finala : (q_f ,ε) , $q_f \in F$, ε este secventa vidă

(corespunde intotdeauna acceptarii)

Relații definite între configurații

- $\vdash$ **tranziție** (simplă, într-un pas, en. *move*)
  $(q,ax) \vdash (p,x)$, $p \in \delta(q,a)$

- $\overset{k}{\vdash}$ **k tranziție** = o secvență de k tranziții directe) $C_0 \vdash C_1 \vdash \ldots \vdash C_k$
- $\overset{+}{\vdash}$ **+ tranziție**
  $C \overset{+}{\vdash} C'$: $\exists$ $k>0$ astfel încât $C \overset{k}{\vdash} C'$
- $\overset{*}{\vdash}$ *** tranziție (tranziție stelată)**
  $C \overset{*}{\vdash} C'$: $\exists$ $k\geq 0$ astfel încât $C \overset{k}{\vdash} C'$

Definitie : ***Limbajul*** acceptat de AF M = (Q,Σ,δ,q0,F) este :

$$L(M)=\{ w \in \Sigma^* \mid (q_0,w) \vdash^* (q_f ,\varepsilon) , q_f \in F \}$$

Observații

1. 2 automate finite M_1 si M_2 sunt echivalente daca si numai daca genereaza acelasi limbaj

$$L(M_1)=L(M_2)$$

1. $\varepsilon \in L(M) \Leftrightarrow q_0 \in F$ (starea inițială este și stare finală)

Reprezentări AF

- $\vdash$