~ Seminar 4 ~

Gramatici

G = (N, T, S, P)
N = {A, B, C, …} mulţimea de simboluri neterminale
T = {a, b, c, ...} mulţimea de simboluri terminale
S $\in$ N simbolul de start
P mulţimea de producţii (sau reguli de producţie)

Gramatici independente de context (GIC) / Context-free grammars (CFG)

$P \subseteq N \times (N \cup T)^*$

Exemplu:
L1 = {$a^n b^n$ | n ≥ 0} = {$\lambda = a^0b^0$, ab, a^2b^2, a^3b^3, a^4b^4, …}

$$S \to \overset{(1)}{aSb} \mid \overset{(2)}{\lambda}$$

Ideea acestei gramatici este de a genera cuvântul dinspre margini spre mijlocui lui, astfel pentru fiecare aplicare a producţiei (1) vor apărea un „a” şi un „b”, iar între ele va rămâne S-ul pentru a continua generarea. La final, se aplică producţia (2) pentru a nu mai avea neterminal în cuvântul obţinut.
Cel mai mic cuvând este într-adevăr cuvântul vid, pentru că putem aplica direct a doua producţie fără să o aplicăm deloc pe prima.

Pentru a genera cuvântul a^3b^3:
$$S \overset{(1)}{\Rightarrow} aSb \overset{(1)}{\Rightarrow} aaSbb \overset{(1)}{\Rightarrow} aaaSbbb \overset{(2)}{\Rightarrow} aaabbb$$

EXERCITII:
L2 = {$a^n b^{2n}$ | n ≥ 1} = {ab^2, a^2b^4, a^3b^6, a^4b^8…}

L3 = {$a^{2n} b^k c^{3n}$ | n ≥ 0, k ≥ 1}
= { b, b^2, b^3, b^4, …,
a^2bc^3, $a^2b^2c^3$, $a^2b^3c^3$, a^2b^4c3, …, a^4bc^6, $a^4b^2c^6$, $a^4b^3c^6$, ... }

L4 = {$a^{2n} c^{3n} b^k$ | n ≥ 1, k ≥ 1}
= { a^2c^3b, $a^2c^3b^2$, $a^2c^3b^3$, $a^2c^3b^4$, …, a^4c^6b, $a^4c^6b^2$, $a^4c^6b^3$, ... }
Verificare generare cuvânt: $a^4c^6b^3$.

L5 = {$a^n b^k c^{2k} d^n$ | n ≥ 1, k ≥ 0}
= { ad, a^2d^2, a^3d^3, a^4d^4, ...,
abc^2d, ab^2c^4d, ab^3c^6d, ..., $a^2bc^2d^2$, $a^2b^2c^4d^2$, $a^2b^3c^6d^2$, ... }

L6 = {$a^n b^k c^{2k} d^n$ | n ≥ 0, k ≥ 1}
= { bc^2, b^2c^4, b^3c^6, ...,
abc^2d, ab^2c^4d, ab^3c^6d, ..., $a^2bc^2d^2$, $a^2b^2c^4d^2$, $a^2b^3c^6d^2$, ... }

L7 = {$a^n b^{2n} c^{2k} d^k$ | n ≥ 0, k ≥ 1}
= { c^2d, c^4d^2, c^6d^3, ...,
ab^2c^2d, $ab^2c^4d^2$, $ab^2c^6d^3$, ..., $a^2b^4c^2d$, $a^2b^4c^4d^2$, $a^2b^4c^6d^3$, ... }

$L8 = \{a^n b^n a^k b^k \mid n, k \geq 0\}$
$= \{ \lambda, ab, a^2b^2, a^3b^3, a^4b^4, \ldots, abab, aba^2b^2, aba^3b^3, a^2b^2ab, a^3b^3ab, a^2b^2a^2b^2, \ldots \}$

$L9 = \{a^n\, b^{2n}\, c^k\, d^p\, e^{3k} \mid n, k, p \geq 0\}$
$= \{ \lambda, ab^2, a^2b^4, a^3b^6, \ldots, ce^3, c^2e^6, c^3e^9, \ldots, d, d^2, d^3, d^4, \ldots,$
$cde^3, c^2de^6, cd^2e^3, \ldots, ab^2d, a^2b^4d, ab^2d^2, \ldots, ab^2ce^3, a^2b^4ce^3, ab^2c^2e^6, \ldots \}$

$L10 = \{w \cdot w^R \mid w \in \{a,b,c\}^+ \}$ **(R=reversed** = cuvântul oglindit**)**
$= \{ aa, bb, cc, aaaa, abba, acca, baab, bbbb, bccb, caac, cbbc, cccc, \ldots\}$

Automatul Push-Down (APD) / Automatul Stivă

APD = $(Q, \Sigma, q_0, F, \delta, \Gamma, Z_0)$
Q mulţimea de stări
Σ alfabetul de intrare („sigma”)
$q_0 \in Q$ starea iniţială
$F \subseteq Q$ mulţimea de stări finale
Γ alfabetul stivei („gamma”)
$Z_0 \in \Gamma$ simbolul iniţial al stivei
$\delta : Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma^*$ funcţia de tranziţie („delta”)

Modul în care funcţionează tranziţiile pentru un APD
- La intrare avem nevoie de 3 parametri:
    - starea curentă *(element din mulţimea Q)*
    - caracterul curent din cuvântul de intrare *(element din mulţimea Σ)*
    - **simbolul** aflat în vârful stivei *(element din mulţimea Γ)*
- La ieşire vom avea 2 parametri:
    - starea în care ajungem *(element din mulţimea Q)*
    - **cuvântul** cu care înlocuim **simbolul** din vârful stivei *(element din mulţimea Γ^*)*

Atenţie! Simbolul din vârful stivei este mereu *eliminat* din stivă, înainte de a se adăuga noul cuvânt în locul lui.

Deci avem mai multe variante pentru această înlocuire:
- Dacă scriem λ în stivă, înseamnă că doar „am şters” simbolul din vârful stivei fără a adăuga nimic în locul lui.
- Dacă scriem acelaşi caracter în stivă (cel care fusese eliminat), înseamnă că dorim să „păstrăm stiva nemodificată”.
- Dacă scriem în stivă un cuvânt de lungime minim 2 avem două cazuri:
  -- dacă dorim să păstrăm şi caracterul care era deja în stivă, atunci cuvântul introdus va avea ultima literă exact cea care fusese eliminată *(deci partea de început, din stânga, a cuvântului reprezintă vârful stivei)*.
  -- dacă nu dorim să păstrăm caracterul care era în vârful stivei, ci chiar să-l înlocuim, atunci cuvântul se poate termina cu alt caracter decât cel eliminat.

Obs: La fel ca automatele finite, şi automatele push-down pot fi:
- *„deterministe”* (definite anterior, dar sunt acceptate şi λ–tranziţiile! Dar dacă între două stări avem λ–tranziţie, nu putem avea tranziţie şi cu un caracter din Σ.)
- *„nedeterministe”* (când avem mai multe posibilităţi de continuare: fie ajungând în stări diferite, fie scriind pe stivă cuvinte diferite, fie ambele (ajungând în stări diferite şi scriind pe stivă cuvinte diferite))
- *„nedeterministe cu λ–tranziţii”* (când între două stări avem voie să avem tranziţii şi cu λ şi cu caractere din Σ şi în plus avem nedeterminismul descris anterior)

Modalităţi de acceptare a unui cuvânt de către un APD

(a) cu stări finale

$$L_F(APD) = \{w \mid (q_0, w, Z_0) \overset{*}{\mapsto} (\underline{p}, \underline{\lambda}, \alpha),\ p \in F, \alpha \in \Gamma^*\}$$

(Obs: semnul cu steluţă deasupra ar trebui să fie doar T-ul întors, fără săgeată în dreapta, dar nu am găsit simbolul potrivit...)

Citim: Cuvântul w este acceptat de automat dacă pornind din configuraţia având starea q_0, cuvântul de intrare w şi pe stivă simbolul iniţial Z_0, după ce aplicăm un număr oarecare de tranziţii ajungem în configuraţia finală.

În acest caz (a) ne interesează:
- prima componentă să fie o stare finală;
- a doua componentă să fie cuvântul vid (adică să fi terminat de parcurs tot cuvântul de intrare)

În acest caz (a) NU ne interesează a treia componentă (conţinutul stivei): poate să fie vidă sau poate să conţină un cuvânt cu simboluri din alfabetul Γ.

(b) cu vidarea stivei

$$L_\lambda(APD) = \{w \mid (q_0, w, Z_0) \overset{*}{\mapsto} (p, \underline{\lambda}, \underline{\lambda}),\ p \in Q\}$$

În acest caz (b) ne interesează:
- a doua componentă să fie cuvântul vid (adică să fi terminat de parcurs tot cuvântul de intrare)
- a treia componentă să fie cuvântul vid (adică stiva să fie vidă)

În acest caz (b) NU ne interesează prima componentă: starea în care ajungem poate să fie finală sau poate să fie nefinală.

Obs: **Atenţie**, dacă ştergeţi tot conţinutul stivei, *automatul îşi opreşte funcţionarea*, pentru că nu poate aplica funcţia delta dacă nu mai are parametrul al treilea (simbolul din vârful stivei). Deci asiguraţi-vă că NU se va întâmpla asta înainte de a fi citit tot cuvântul de intrare. În general e bine să-l păstrăm pe Z_0 la început şi să-l eliminăm abia la final (eventual folosind o λ–tranziţie).

(c) cu stări finale şi cu vidarea stivei

În unele cărţi apare şi acest al treilea caz, care este de fapt intersecţia primelor două.

$$L_{F\&\lambda}(APD) = \{w \mid (q_0, w, Z_0) \overset{*}{\mapsto} (\underline{p}, \underline{\lambda}, \underline{\lambda}),\ p \in F\}$$

În acest caz (c) ne interesează:
- prima componentă să fie o stare finală;
- a doua componentă să fie cuvântul vid (adică să fi terminat de parcurs tot cuvântul de intrare)
- a treia componentă să fie cuvântul vid (adică stiva să fie vidă)

Obs:
1) În general vom folosi *litere mari pentru alfabetul stivei*, pentru a fi mai clară diferența între caracterele citite din cuvântul de intrare ci cele din stivă.
2) În general vom folosi a 3-a metodă de acceptare (*cu stări finale și vidarea stivei*).

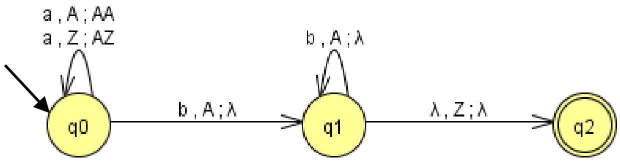
Exemplu:

L1 = {$a^n b^n$, n ≥ 1}

Vrem ca numărul de „a”-uri să fie egal cu numărul de „b”-uri, deci vom reţine acest număr cu ajutorul stivei astfel:

- pentru fiecare „a” citit din cuvântul de intrare, vom adăuga un „A” în stivă;
- pentru fiecare „b” citit vom şterge un „A” din stivă;

Trebuie să ne asigurăm că citim cel puţin un „a” şi un „b” (pentru că n ≥ 1) şi că literele nu sunt amestecate. De aceea, după ce citim primul „b” va trebui să schimbăm starea.



Obs: *Tranziţiile se reprezintă cu structura următoare:* caracterul citit la intrare, apoi virgulă, apoi caracterul care va fi eliminat din vârful stivei, apoi punct şi virgulă (sau slash „ / ”), apoi cuvântul care va fi scris în stivă (cu aceeaşi convenţie, ***prima literă din cuvânt este cea care va fi în vârful stivei*** după adăugarea cuvântului).

Verificare acceptare cuvânt de către un APD

Folosim „configuraţii” (sau „descrieri instantanee”) având 3 componente: starea curentă, ce a mai rămas de citit din cuvântul de intrare şi (întreg!) conţinutul stivei (partea stângă fiind vârful stivei, unde se fac ştergerile şi scrierile).

La fiecare pas, căutăm pe graf tranziția care
- pleacă din starea curentă,
- citește prima literă din cuvântul curent (sau eventual λ) și
- citește din stivă primul caracter din cuvântul care reprezintă întreg conținutul stivei

=> în configurația următoare :
- punem starea spre care a ajuns acea tranziție,
- prima literă din cuvânt dispare dacă a fost citită (sau rămâne daca s-a citit λ),
- iar vârful stivei se înlocuiește cu cuvântul spus de tranzție (cel de după “/” sau “;”), iar restul stivei se concatenează după.

Fie n = 3 deci avem cuvântul a^3b^3.

$(q_0, a^3b^3, Z_0) \vdash (q_0, a^2b^3, AZ_0) \vdash (q_0, ab^3, AAZ_0) \vdash (q_0, b^3, AAAZ_0)$
$\vdash (q_1, b^2, AAZ_0) \vdash (q_1, b, AZ_0) \vdash (q_1, \lambda, Z_0) \vdash (q_2, \lambda, \lambda)$, $q_2 \in F$ => a^3b^3 cuvant acceptat.

EXERCITII:

L2 = {$a^{2n} b^k c^{3k} d^n$ | n ≥ 1, k ≥ 1}

L3 = {$a^{2n} b^{3n} c^{2k} d^k$ | n ≥ 0, k ≥ 1}

L4 = {$w \cdot w^R$ | w ∈ $\{a,b,c\}^+$ }

L5 = {$a^{n+2}b^{k+1}$ | n ≥ 1, k ≥ 1, n ≠ k}

L6 = {w ∈ $\{a,b\}^*$ | $|w|_a = |w|_b$} (nr egal de a-uri si b-uri)

L7 = {$a^ib^jc^k$ | i = j sau j = k}

L8 = { $a^ib^jc^k$ | j > i + k; i,j,k ≥ 1}