~ Seminar 5 ~

➢ Definiție APD și mod de funcționare

Automat Pushdown (APD) / Pushdown Automaton (PDA)
APD = (Q, Σ, q_0, F, δ, Γ, Z_0)
Q mulţimea de stări (mulțime finită și nevidă)
Σ alfabetul de intrare („Sigma”) (mulțime finită)
$q_0 \in Q$ starea iniţială
$F \subseteq Q$ mulţimea de stări finale
Γ alfabetul stivei („Gamma”) (mulțime finită și nevidă)
$Z_0 \in \Gamma$ simbolul iniţial al stivei
$\delta: Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to 2^{Q \times \Gamma^*}$ 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)* sau cuvântul vid λ
  - **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 <u>înlocuim</u> **simbolul** din vârful stivei *(element din mulţimea Γ^*)*
    (*stânga* cuvântului va fi vârful stivei)

Atenţie! Simbolul din vârful stivei este mereu *eliminat* din stivă, înainte de a se adăuga noul cuvânt în locul lui.
În stivă ordinea de adăugare a literelor este de la dreapta spre stânga cuvântului.

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).

- **Modalităţi de acceptare a unui cuvânt de către un APD**

(a) cu stări finale
$L_F(APD) = \{w \mid (q_0, w, Z_0) \vdash^* (\boldsymbol{p}, \boldsymbol{\lambda}, \alpha), \boldsymbol{p} \in \boldsymbol{F}, \alpha \in \Gamma^*\}$
(b) cu vidarea stivei
$L_\lambda(APD) = \{w \mid (q_0, w, Z_0) \vdash^* (p, \boldsymbol{\lambda}, \boldsymbol{\lambda}), p \in Q\}$
(c) cu stări finale şi cu vidarea stivei
$L_{F\&\lambda}(APD) = \{w \mid (q_0, w, Z_0) \vdash^* (\boldsymbol{p}, \boldsymbol{\lambda}, \boldsymbol{\lambda}), \boldsymbol{p} \in \boldsymbol{F}\}$

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

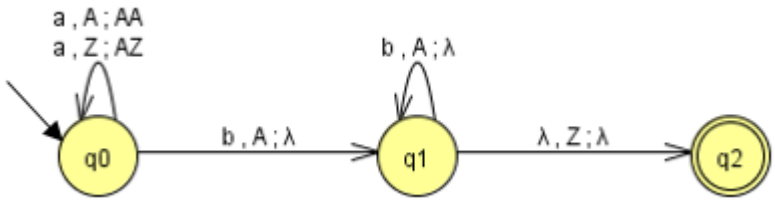
- **Verificare acceptare cuvânt de către un APD**

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 dacă s-a citit λ),
- iar vârful stivei se înlocuiește cu cuvântul spus de tranziție (cel de după “/” sau “;”), iar restul stivei se concatenează după.

- ***Exemplu rezolvat:*** $L1 = \{a^n b^n \mid n \geq 1\}$ *(discutat și la curs)*

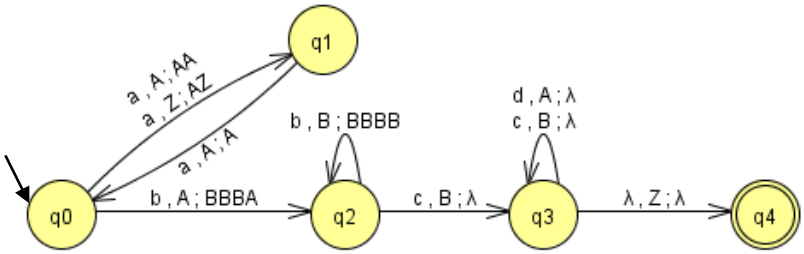


Verificare acceptare cuvânt
Fie n = 3 deci avem cuvântul a^3b^3.
$(q_0, a^3b^3, Z_0) \vdash (q_0, a^2b^3, AZ_0) \vdash (q_0, ab^3, AAZ_0) \vdash (q_0, b^3, AAAZ_0)$
$\vdash (q_1, b^2, AAZ_0) \vdash (q_1, b, AZ_0) \vdash (q_1, \lambda, Z_0) \vdash (q_2, \lambda, \lambda), q_2 \in F \Rightarrow a^3b^3$ cuvânt acceptat.

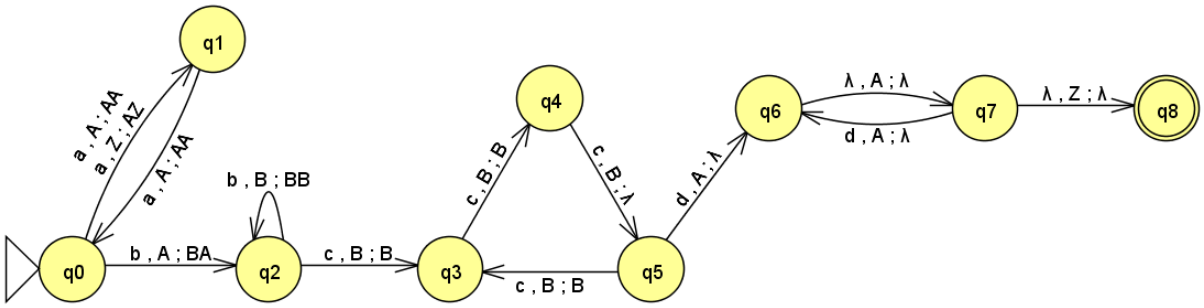
- ***Exemple:***

$\boldsymbol{L2 = \{a^{2n}b^k c^{3k} d^n \mid n \geq 1, k \geq 1\}}$

Idee rezolvare 1:
1) Scriem n de A în stivă: pentru fiecare 2 de a,
primul adaugă un A în stivă, al doilea nu modifică stiva.
2) Scriem 3k de B în stivă: fiecare b adaugă câte 3 de B în stivă.
3) Ștergem 3k de B din stivă: fiecare c elimină câte un B din stivă.
4) Ștergem n de A din stivă: fiecare d elimină câte un A din stivă.

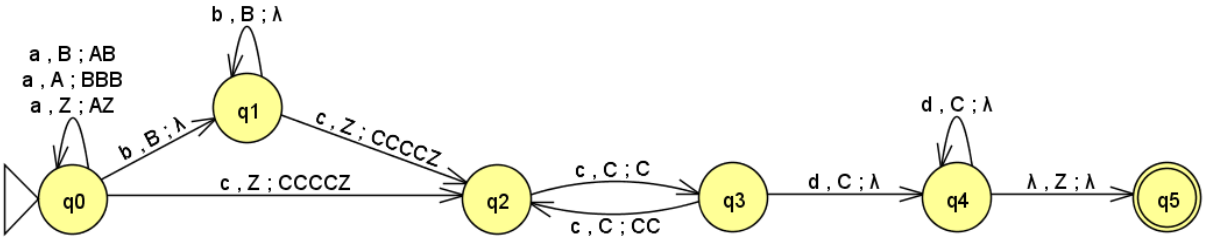


Idee rezolvare 2:
1) Scriem 2n de A în stivă: pentru fiecare 2 de a, ambii adaugă câte un A în stivă.
2) Scriem k de B în stivă: fiecare b adaugă câte un B în stivă.
3) Ștergem k de B din stivă: pentru fiecare 3 de c,
primii 2 de c nu modifică stiva, iar al 3-lea c elimină un B din stivă.
4) Ștergem 2n de A din stivă: pentru fiecare 2 de A,
un d elimină un A din stivă, iar o lambda-tranziție elimină al 2-lea A din stivă.



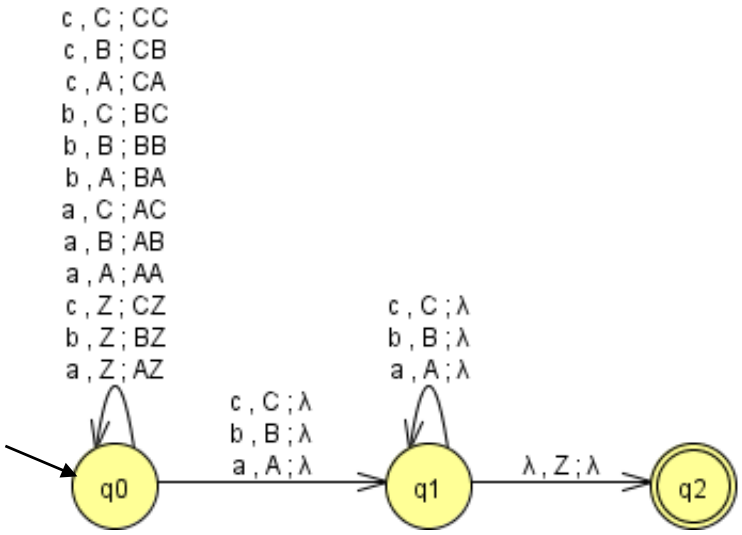
$$\boldsymbol{L3} = \{\boldsymbol{a^{2n}b^{3n}c^{2k}d^{k+3}} \mid \boldsymbol{n \geq 0, k \geq 1}\}$$

Idee rezolvare:
1) (Pentru verificarea parității a-urilor nu vom folosi circuit între 2 stări diferite,
ci vom avea buclă pe aceeași stare dar vom folosi simboluri diferite în stivă.)
Scriem 3n de B în stivă: pentru fiecare 2 de a,
primul a adaugă un A în stivă, al doilea a elimină A din stivă și adaugă 3 de B.
2) Ștergem 3n de B din stivă: fiecare b elimină câte un B din stivă.
3) Scriem k+3 de C în stivă: pentru fiecare 2 de c,
primul c din pereche adaugă un C în stivă
(*excepție:* primul c din prima pereche adaugă 1+3=4 de C),
al doilea c din pereche nu modifică stiva.
4) Ștergem k+3 de C din stivă: fiecare d elimină câte un C din stivă.



$$\boldsymbol{L4} = \{\boldsymbol{w \cdot w^R} \mid \boldsymbol{w} \in \{\boldsymbol{a, b, c}\}^+\}$$

Idee rezolvare:
1) Cât timp citim prima parte din cuvânt îl salvăm în stivă:
fiecare literă mică a/b/c adaugă în stivă litera corespunzătoare A/B/C.
2) Când presupunem că am ajuns la mijlocul cuvântului (schimbăm starea),
cât timp citim a doua parte din cuvânt verificăm oglindirea cu cuvântul din stivă:
fiecare literă mică a/b/c trebuie să șteargă din stivă exact aceeași literă A/B/C.



➢ Gramatici / Grammars

G = (N, T, S, P)
N = {A, B, C, …} mulţimea de simboluri **neterminale** (mulțime finită și nevidă)
T = {a, b, c, …} mulțimea de simboluri **terminale** (mulțime finită)
$S \in N$ **simbolul de start**
P mulțimea de **producții** (sau reguli de producție)

Obs: $\boldsymbol{T = \Sigma}$ (alfabetul peste care sunt definite cuvintele *generate* de gramatică)

❖ *Gramatici regulate / Regular Grammars*

Au doar producții de forma: $\boldsymbol{A \to aB \mid a \mid \lambda}$, unde $A, B \in N, a \in T$

- Echivalența dintre Gramaticile regulate și Automatele finite

Gramatici regulate	**Automate finite**
N = mulțimea de simboluri **neterminale**	Q = mulțimea de **stări**
T = mulțimea de simboluri **terminale**	Σ = **alfabetul** de intrare
$S \in N$ **simbolul de start**	$q_0 \in Q$ **starea inițială**
- producția $\boldsymbol{A \to aB}$, cu $A, B \in N, a \in T$	- tranziția $\boldsymbol{\delta(q_A, a) = q_B}$, cu $q_A, q_B \in Q, a \in \Sigma$
- producția $\boldsymbol{A \to a}$, cu $A \in N, a \in T$	- tranziția $\boldsymbol{\delta(q_A, a) = q_f}$, cu $q_A \in Q, a \in \Sigma, q_f \in F$
- producția $\boldsymbol{A \to \lambda}$, cu $A \in N$	- tranziția $\boldsymbol{\delta(q_A, \lambda) = q_f}$, cu $q_A \in Q, q_f \in F$ (sau direct *stare finală* $\boldsymbol{q_A \in F}$)

- Mod funcționare gramatică:
  - Se începe de la cuvântul format doar din simbolul de start S.
  - Cât timp e posibil, se aplică un pas de derivare:
    - se alege un neterminal din cuvântul curent și
    - se alege o producție care are ca membru stâng acest neterminal,
    - apoi se înlocuiește în cuvântul curent acel neterminal (o singură apariție a sa) cu membrul drept al acelei producții și se obține noul cuvânt curent.
  - Când cuvântul nu mai conține niciun neterminal înseamnă că s-a terminat generarea (și acel cuvânt obținut face parte din limbajul generat de gramatică).

Exemplu:
Pentru cuvântul $\alpha \boldsymbol{X} \beta$, cu $X \in N$ și $\alpha, \beta \in (N \cup T)^*$
și producția $\boldsymbol{X \to w}$, cu $X \in N$ și $w \in (N \cup T)^*$,
rezultă cuvântul $\alpha \boldsymbol{w} \beta$.

- ***Exemple:***

$\boldsymbol{L1 = \{a^n \mid n \geq 0\}}$
$S \to aS \mid \lambda$

$\boldsymbol{L2 = \{a^n \mid n \geq 1\}}$
$S \to aS \mid a$

$\boldsymbol{L3 = \{a^{2n} \mid n \geq 0\}}$
$S \to aA \mid \lambda$
$A \to aS$

$\boldsymbol{L4 = \{a^{2n} \mid n \geq 1\}}$
$S \to aA$
$A \to aS \mid a$

$\boldsymbol{L5 = \{a^{2n+1} \mid n \geq 0\}}$
$S \to aA \mid a$
$A \to aS$

$\boldsymbol{L6 = \{a^{2n+1} \mid n \geq 1\}}$
$S \to aA$
$A \to aS \mid aB$
$B \to a$

sau
$S \to aA$
$A \to aB$
$B \to aA \mid a$

$\boldsymbol{L7 = \{a^{2n}b^{3k} \mid n \geq 1, k \geq 0\}}$
$S \to aA$ (1)
$A \to aS$ (2) | aB (3)
$B \to bC$ (4) | λ (5)
$C \to bD$ (6)
$D \to bB$ (7)

- **Verificare generare cuvânt de către gramatică:**

Verificare generare cuvânt a^2b^3:
$\boldsymbol{S}$ (1) $\Rightarrow a\boldsymbol{A}$ (3) $\Rightarrow aa\boldsymbol{B}$ (4) $\Rightarrow aab\boldsymbol{C}$ (6) $\Rightarrow aabb\boldsymbol{D}$ (7) $\Rightarrow aabbb\boldsymbol{B}$ (5) $\Rightarrow aabbb$

❖ *Gramatici independente de context (GIC) / Context-free Grammars (CFG)*

Au producții de forma: $\boldsymbol{P \subseteq N \times (N \cup T)^*}$

Adică membrul stâng al producției este un neterminal, iar membrul drept al producției poate fi cuvântul vid sau un cuvânt de orice lungime (finită) care conține oricâte neterminale și/sau terminale amestecate oricum.

- ***Exemple:***

$\boldsymbol{L1 = \{a^n b^n \mid n \geq 0\}}$
$S \to aSb\ (1) \mid \lambda\ (2)$

Pentru a genera cuvântul a^3b^3:
$\boldsymbol{S}\ (1) \Rightarrow a\boldsymbol{S}b\ (1) \Rightarrow aa\boldsymbol{S}bb\ (1) \Rightarrow aaa\boldsymbol{S}bbb\ (2) \Rightarrow aaabbb$

$\boldsymbol{L2 = \{a^n b^n \mid n \geq 1\}}$
$S \to aSb \mid ab$

$\boldsymbol{L3 = \{a^n b^{2n} \mid n \geq 1\}}$
$S \to aSbb \mid abb$

$\boldsymbol{L4 = \{a^{2n} b^k c^{3n} \mid n \geq 0, k \geq 1\}}$
$S \to aaSccc \mid bA$
$A \to bA \mid \lambda$

sau
$S \to aaSccc \mid A$
$A \to bA \mid b$

$\boldsymbol{L5 = \{a^{2n} b^k c^{3n} \mid n \geq 1, k \geq 1\}}$
$S \to aaSccc \mid aaAccc$
$A \to bA \mid b$

$\boldsymbol{L6 = \{a^n b^k c^{2k} d^n \mid n \geq 1, k \geq 0\}}$
$S \to aSd \mid aAd$
$A \to bAcc \mid \lambda$

$\boldsymbol{L7 = \{a^n b^k c^{2k} d^n \mid n \geq 0, k \geq 1\}}$
$S \to aSd \mid A$
$A \to bAcc \mid bcc$

sau
$S \to aSd \mid bAcc$
$A \to bAcc \mid \lambda$

$\boldsymbol{L8 = \{a^n b^{2n} c^{2k} d^k \mid n \geq 0, k \geq 1\}}$
$S \to AB$
$A \to aAbb \mid \lambda$
$B \to ccBd \mid ccd$