◇ Automate push down (stivă) APD

Def $A = (Q, \Sigma, \underline{\underline{\Gamma}}, q_0, \underline{\underline{Z_0}}, \delta, F)$ automat stivă

Γ = alfabetul stivei

Z_0 = simbol de start pe stivă

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \underset{\text{un simbol}}{\Gamma} \to 2^{Q \times \Gamma^*} = \mathcal{P}(Q \times \Gamma^*)$$

def descriere instantanee (configurație)

$$(q, a\alpha, \boxed{Z}\beta) \vdash (p, \alpha, \boxed{\gamma}\beta) \iff$$

$$\delta(q, a, Z) \ni (p, \gamma) \quad \underline{\underline{\text{desen}}} \quad q \xrightarrow{a,Z/\gamma} p$$

$\overset{*}{\vdash}$ e închiderea reflexivă și tranzitivă pt $\vdash$

$$L_F(A) = \{ w \mid (q_0, w, \underline{\underline{Z_0}}) \overset{*}{\vdash} (q_f, \lambda, \alpha), \underline{\underline{q_f \in F}} \}$$

$$L_\lambda(A) = \{ w \mid (q_0, w, \underline{\underline{Z_0}}) \overset{*}{\vdash} (q, \lambda, \underline{\underline{\lambda}}) \}$$

def APD determinist $\iff$ $|\delta(q, a, X) \cup \delta(q, \lambda, X)| \leq 1$

$$\mathcal{L}(\text{APDD}) \subsetneq \mathcal{L}(\text{APD}) \qquad L = \{ w w^R \mid w \in \{a,b\}^* \}$$

$$L \in \mathcal{L}(\text{APD}) \setminus \mathcal{L}(\text{APDD})$$

②

Obs $(q, a\alpha, \boxed{Z}\beta) \vdash (p, \alpha, \boxed{\gamma}\beta)$, $\gamma \in \Gamma^*$

$\gamma = \lambda$ $\quad (q, a\alpha, \boxed{Z}\beta) \vdash (p, \alpha, \underline{\beta}) \simeq$ **se șterge** vârful stivei

$\gamma = \delta Z$ $\quad (q, a\alpha, \boxed{Z}\beta) \vdash (p, \alpha, \boxed{\delta Z}\beta) \simeq$ se adaugă δ pe stivă

Ex: Construiți APD pentru: $L = \{ a^m b^n \mid m > n \geq 1 \}$

Idee: - pentru fiecare "a" (simbol curent) pun pe stivă A.
- când ajung la analiza "b"-urilor - trebuie

! schimbată starea - pentru a nu permite alte "a"-uri
- fiecare "b" (simbol curent) va scoate din stivă un "A"
- când s-au terminat simbolurile din cuvânt $\rightarrow$ pe stivă mai este cel puțin un "A"

λ var 1 - sau scot 'A' urile până se videază stiva

F var 2 - cu primul 'A' ajung în stare finală

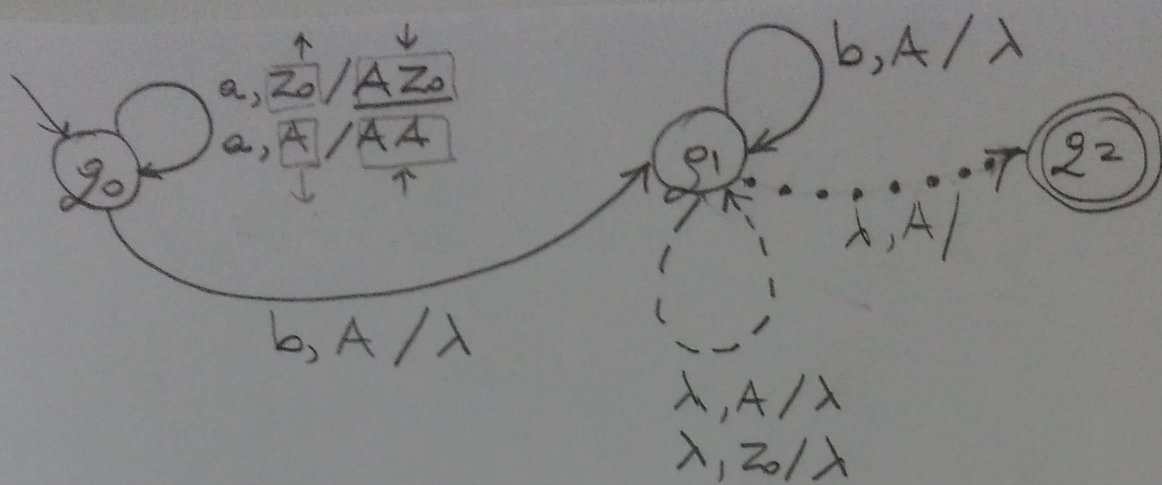
în ambele cazuri cuvântul e acceptat

ex pt a^3b^2 $\quad (q_0, \underline{a}^3 b^2, Z_0) \vdash (q_0, \underline{a}^2 b^2, \underline{A} Z_0)$

$\vdash (q_0, \underline{a} b^2, \underline{A}A Z_0) \vdash (q_0, b^2, \underline{A}AAZ_0) \vdash$

$\vdash (\underline{q_1}, \underline{b}, \underline{AAZ_0}) \vdash (q_1, \lambda, AZ_0) \vdash (q_1, \lambda, Z_0) \vdash (q_1, \lambda, \underline{\lambda})$

$\vdash (\underline{q_2}, \lambda, Z_0)$ $\underline{q_2 \in F}$



Ex: Construiți APD pt $L = \{ \alpha \in \{a,b\}^* \mid N_a(\alpha) = N_b(\alpha) \}$

<u>Idee</u>:

- pentru 'a' (simbol curent) – dacă vf stivei e 'Z_0' – adaug 'A'
  – dacă vf stivei e 'A' – adaug 'A'
- pentru 'b' (simbol curent) – dacă vf stivei e 'Z_0' – adaug 'B'
  – dacă vf stivei e 'B' – adaug 'B'
  – dacă vf stivei e 'A' – îl șterg
  (un 'b' șterge un 'A' pus de un 'a')
- pentru 'a' (simbol curent) – dacă vf stivei e 'B' – îl șterg
  (un 'a' șterge un 'B' pus de un 'b')
- când s-a terminat cuvântul – dacă în stivă e doar Z_0 => nr de 'a' = nr de 'b' => cuvântul e acceptat

! <u>Obs</u> – în orice moment pe stivă vor fi sau $A^i Z_0$ sau $B^i Z_0$ (adică nu $A^i B^j A^k B^l Z_0$)

pentru ca | orice b – scoate A => nu se poate $BA\alpha$
| orice a – scoate B => nu pot avea $AB\alpha$

<u>Obs</u> Z_0 e foarte important pe stivă = capăt stivă
semnalează stiva fără A și B (dacă nu
ar întâlni <u>1 simbol</u> pe stivă → se oprește

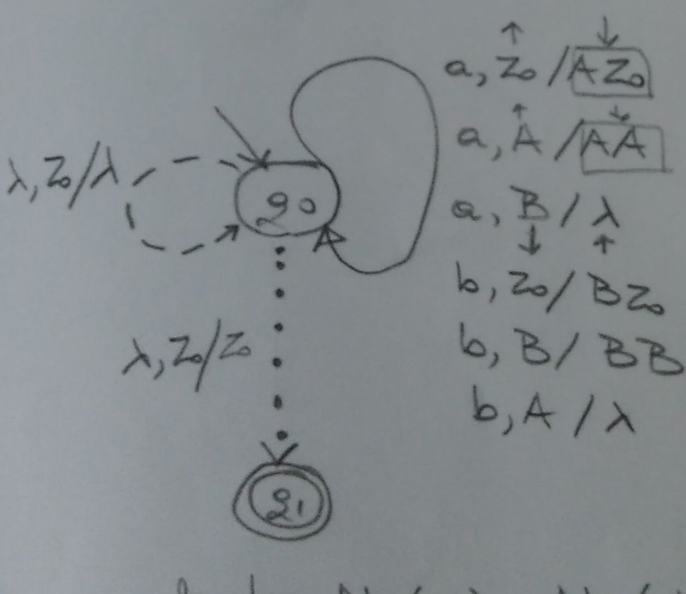
- fie recunoaște cuvântul - dacă nu mai sunt simboluri în cuvântul de intrare
- fie se blochează - dacă mai sunt simboluri în cuvântul de intrare

ex: a^2b^3a $\quad (q_0, \underline{a}^2b^3a, Z_0) \vdash (q_0, \underline{a}b^3a, \underline{A}Z_0)$

$\vdash (q_0, b^3a, \underline{A}AZ_0) \vdash (q_0, b^2a, AZ_0) \vdash (q_0, ba, Z_0)$

$\vdash (q_0, a, BZ_0) \vdash (q_0, \lambda, Z_0) \vdash (q_1, \lambda, Z_0)$ $\quad \underline{q_1 \in F}$

$\vdash (q_0, \lambda, \underline{\lambda})$



<u>Obs</u> se acceptă și λ

$(q_0, \lambda, Z_0) \vdash (q_0, \lambda, \lambda)$

$\vdash (q_1, \lambda, Z_0)$

analog
- $N_a(\alpha) > N_b(\alpha)$ → pe stivă rămân 'A'-uri (cel puțin unul) pe care le vom scoate
- $N_b(\alpha) \geq N_a(\alpha)$ → pe stivă rămân 'B'-uri (posibil nici unul)
- $N_a(\alpha) = N_b(\alpha) + 2$ → pe stivă 'AA' pe care le scoatem