Automate push-down (APD)

•

Automat Push Down (APD)

Definitie:

Un automat push-down (APD) este un ansamblu

$M = (Q, \Sigma, \Gamma, \delta, q_o, Z_o, F)$, unde:

- Q alfabetul starilor;
- Σ alfabetul de intrare;
- Γ alfabetul memoriei stivă; ;
- $q_o \in Q$ stare iniţială;
- $Z_o \in \Gamma$ simbolul de start al memoriei stivă;
- $F \subseteq Q$ mulţimea stărilor finale;
- $\delta: Q x(\Sigma \cup \{\varepsilon\}) x \Gamma \rightarrow \mathcal{P}(Q x \Gamma^*)$ funcţia de tranziţie

Reprezentare

- enumerare
- tabelara
- sub forma de graf

Reprezentare folosind enumerare

Exemplu:

- $M=(\{q_0,q_1\},\{a,b\},\{A,Z\}, \delta,q_0,Z,\{q_0\})$
- δ:

$\delta(q_0,a,Z) = \{(q_0,AZ)\}$

$\delta(q_0,a,A) = \{(q_0,AA)\}$

$\delta(q_0,b,A) = \{(q_1,\varepsilon)\}$

$\delta(q_1,b,A) = \{(q_1,\varepsilon)\}$

$\delta(q_0,\varepsilon,Z) = \{(q_0,\varepsilon)\}$

$\delta(q_1,\varepsilon,Z) = \{(q_0,\varepsilon)\}$ … si Φ in celelalte cazuri

Reprezentare tabelara

Exemplu:

		a	b	ε	
q_0	**Z**	(q_0, AZ)		(q_0, ε)	**1**
	A	(q_0, AA)	(q_1, ε)		
q_1	**Z**			(q_1, ε)	**0**
	A		(q_1, ε)		

Care este limbajul acceptat dupa criteriul stivei vide?

Dar dupa criteriul starii finale?

Dar daca starea finala ar fi q1?

Configuratie

- **formal:**

  $$(q,x,\alpha) \in Q \times \Sigma^* \times \Gamma^*$$

- automatul se găseşte în starea **q**, pe banda de intrare urmează să se citească (accepte) secvenţa **x**, iar în memoria stivă avem secvenţa **α**
- configuratie initiala

  $$(q_0,w,Z_0)$$

Tranzitii

- $\vdash$ ***tranziţie directă***

  $(q,aw,Z\alpha) \vdash (p,w,\gamma\alpha) <=> \delta(q,a,Z) \ni (p,\gamma)$

  sau

  $(q,aw,Z\alpha) \vdash (p,aw,\gamma\alpha) <=> \delta(q,\varepsilon,Z) \ni (p,\gamma)$ **(ε-tranziție)**

  – unde **p,q**∈ **Q, a**∈**Σ**, **Z**∈Γ, w∈Σ^*, **α,γ** din **Γ***

- $\vdash^k$ ***k* tranziţia** (k tranziţii directe) ~AF
- $\vdash^+$ + ***tranziţia*** ~AF
- $\vdash^*$ *** tranziţia*** ~AF

Secventa acceptata de automat

- dupa criteriul stivei vide

  $L_\varepsilon(M) = \{w \mid w \in \Sigma^*, (q_o,w,Z_o) \vdash^* (q,\varepsilon,\varepsilon), q \in Q\}$

  - $(q,\varepsilon,\varepsilon)$ - configuraţia finală după criteriul stivei vide

- dupa criteriul starii finale

  $L_f(M) = \{w \mid w \in \Sigma^*, (q_o,w,Z_o) \vdash^* (q,\varepsilon,\gamma), q \in F, \gamma \in \Gamma^*\}$

  - (q,ε,γ), $q \in F$ configuraţie finală după criteriul stării finale

$L_\varepsilon(M)=\{a^n b^n \mid n \in \mathbf{N}\}$

		a	b	ε	
q_0	Z	(q_0, AZ)		(q_0, ε)	1
	A	(q_0, AA)		(q_1, A)	
q_1	Z			(q_1, ε)	0
	A		(q_1, ε)		

<u>De ce</u>? (justificare:)

in q_0 – se accepta oricate simboluri a

cu ramanere in q_0 si adaugare cate un A in stiva

adica: la fiecare citire de a adaug in stiva un A **(1)**

sau – se trece in starea q_1 (dupa ce am citit cel putin un a, adica am A in stiva)

<u>obs.</u>: se poate trece in q_1 oricand, fara modificarea stivei **(2)**

sau - se scoate Z din stiva

(acest lucru se poate intampla numai inainte de citirea unui simb)

=> **se accepta secventa vida**

in q_1 – cand in varful stivei este un A, se citeste un b

adica: **fiecare citire de b scoate un A din stiva** **(3)**

sau: daca in varful stivei este un Z, acesta se scoate (goleste stiva)

din (1) , (2) , (3) => nr(a) = nr(b)

(4) q_0 citeste a (oricati)

(5) q_1 citeste numai b; nu se poate trece inapoi in q_0

din (2) , (4) , (5) => simb. a citite inaintea simb b

•

Pornind de la:

- un APD care accepta un limbaj dupa criteriul stivei vide

construiti

- un APD echivalent

  care obtine acelasi limbaj dupa criteriul starii finale

Teoreme de echivalenta

<u>Teoremă</u>.

Fie automatul push-down **M**. Există întotdeauna un automat push-down **M'** astfel încât $L_\varepsilon(M') = L_f(M)$; si reciproc.

<u>Teoremă</u>.

Oricare ar fi G – o gramatica independenta de context, există un automat push-down **M** astfel încât $L_\varepsilon(M) = L(G)$;

si reciproc.

G.I.C. => APD echivalent

Fie: **$G = (N, \Sigma, P, S)$** – gram. independenta de context

Cine este M - APD astfel incat $L(G)=L_{\varepsilon}(M)$ **?**

constructia:

$M = (\{q\}, \Sigma, N \cup \Sigma, \delta, q, S, \Phi)$

1. dacă $(A \rightarrow \alpha) \in P$ atunci $(q,\alpha) \in \delta(q,\varepsilon,A)$;
2. $\delta(q,a,a) = \{(q,\varepsilon)\}\ \forall\ a \in \Sigma$;
3. $\delta(.,.,.) = \Phi$ în celelalte cazuri.

Determinism

$\mathbf{M = (Q, \Sigma, \Gamma, \delta, q_o, Z_o, F)}$

este ***determinist*** ddacă:

$\forall \mathbf{Z} \in \Gamma, \forall \mathbf{q} \in \mathbf{Q}, \forall \mathbf{a} \in \Sigma$

1) $|\boldsymbol{\delta}(\mathbf{q},\boldsymbol{\varepsilon},\mathbf{Z})| = \mathbf{0}$ si $|\boldsymbol{\delta}(\mathbf{q},\mathbf{a},\mathbf{Z})| <= \mathbf{1}$

2) $|\boldsymbol{\delta}(\mathbf{q},\boldsymbol{\varepsilon},\mathbf{Z})| = \mathbf{1}$ si $|\boldsymbol{\delta}(\mathbf{q},\mathbf{a},\mathbf{Z})| = \mathbf{0}$

in caz contrar, automatul nu este determinist

- multimea limbajelor acceptate de APD nedeterministe este strict mai larga decat multimea limbajelor acceptate de APD deterministe