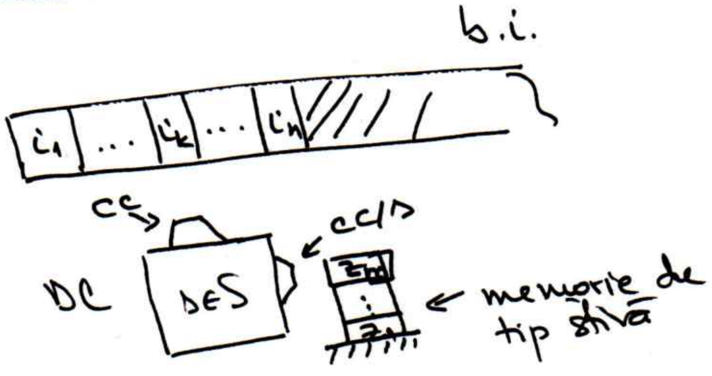
CURS

AUTOMATE PUSH-DOWN

APD – mecanisme de recunoaștere p/r. lb. independente de context

Schema:



$p = i_1 \dots i_n \in I^*$ cuvânt de analizat

$z_1 \dots z_m \in Z^*$ conținutul stivei

<u>Pas:</u> DC citește simbolul din dreptul cc și mută b.i. o poziție stânga (sau citește λ și lasă b.i. pe loc), extrage simbolul $z_m \in Z$ aflat în vârful stivei și în funcție de starea internă $s \in S$, simbolul citit de pe bandă (sau λ) și simbolul extras din stivă execută următoarele operații:

1) schimbă starea în s'
2) scrie pe stivă un cuvânt $q \in Z^*$

$$f : S \times (I \cup \{\lambda\}) \times Z \to \mathcal{P}(S \times Z^*)$$

Funcționare – șir de pași p/r. examinarea tuturor literelor cuvântului de pe b.i.

Inițial: $p = i_1 \dots i_n \in I^*$ pe b.i.
cc în dreptul lui i_1
DC în starea inițială $s_0 \in S$
$z_0 \in Z$ – simbol inițial în stivă.

Recunoașterea cuvintelor – după examinarea tuturor literelor se recunoaște cuvântul dacă stiva devine vidă. Spunem că un cuvânt e recunoscut prin golirea memoriei de stivă

<u>Obs:</u> 1) Citirea ultimei litere nu înseamnă neapărat și terminarea funcționării. Se mai pot citi eventual simboluri λ de pe b.i.

2) Scrierea cuvântului λ pe stivă echivalent cu ștergerea vf. stivei (ce tocmai a fost extras!)

3) În general funcționarea este nedeterministă, adică $|f(s,i,z)|>1$ pt $s,i,z\in S\times I\cup\{\lambda\}\times Z$

4) Blocarea are loc dacă

$$f(s,i,z)=\Phi$$

sau dacă stiva e vidă și mai sunt litere de citit de pe b.i.

DEFINIȚIA FORMALĂ:

$$APD=(S,I,Z,f,s_0,z_0)$$

S - alfabetul stărilor

Z - alfabetul stivei

I - alf. intrărilor (sb. bandă)

$s_0\in S$ starea inițială

$z_0\in Z$ simbol inițial în stivă

$f: S\times(I\cup\{\lambda\})\times Z\longrightarrow \mathcal{P}(S\times Z^*)$ funcția de evoluție

Ptr. definirea unui pas se consideră spațiul "configurațiilor".

$\delta=(s,p,q)\in S\times I^*\times Z^*$ se zice configurație, sau stare curentă a unui APD dacă $s\in S$ este starea curentă a DC, $p\in I^*$ este cuvântul rămas de analizat, începând cu litera din dreptul cc, iar $q\in Z^*$ este conținutul stivei.

Un pas se descrie printr-o relație între 2 config.

$\delta_1=(s_1,p_1,q_1)\longmapsto\delta_2=(s_2,p_2,q_2)$ APD "trece direct" din configurația δ_1 în configurația δ_2 dacă se execută un pas de funcționare, adică

Caz 1 (citim λ)

$$\begin{cases} p_2=p_1 \\ q_1=zq', \quad z\in Z \\ f(s_1,\lambda,z)\ni(s_2,q), \quad q\in Z^* \\ q_2=qq' \end{cases}$$

Caz 2 (citim $i\in I$)

$$\begin{cases} p_1=ip' \\ q_1=zq', \quad z\in Z \\ f(s_1,i,z)\ni(s_2,q) \\ p_2=p' \\ q_2=qq' \end{cases}$$

Considerăm $\overset{*}{\mapsto}$ închiderea reflexivă și tranzitivă a relației $\mapsto$

Spunem că APD evoluează din δ_1 în δ_2 dacă $\delta_1 \overset{*}{\mapsto} \delta_2$

DEFINIȚIE: $L(APD) = \{ p \in I^* \mid (s_0, p, z_0) \overset{*}{\mapsto} (s, \lambda, \lambda), s \in S \}$

Exemplu: APD pentru recunoașterea lb. $\{a^n b^n \mid n \geq 0\} \notin R$.

#

Ideea: Citim simbolurile a ~~de~~ la stânga la dreapta și le aruncăm în stivă. La citirea primului b schimb starea DC și eliminăm a din vf. stivei. În continuare citirea unui b elimină un a din stivă.

Cuvintele corecte vor goli stiva!

Descrierea funcției de evoluție:

1) Pornirea APD

$$f(s_0, a, z_0) = (s_0, az_0)$$

2) Umplerea stivei

$$f(s_0, a, a) = (s_0, aa)$$

3) Găsirea mijlocului

$$f(s_0, b, a) = (s_1, \lambda)$$

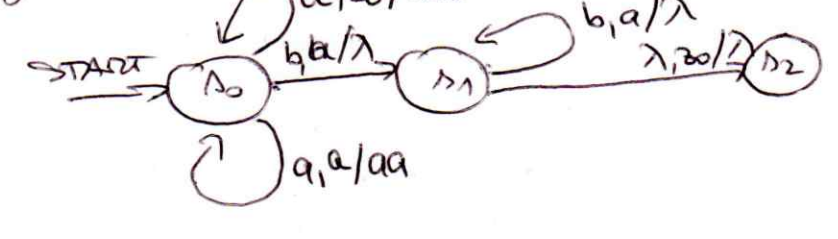
4) Golirea stivei

$$f(s_1, b, a) = (s_1, \lambda)$$

5) Recunoașterea cuvintelor

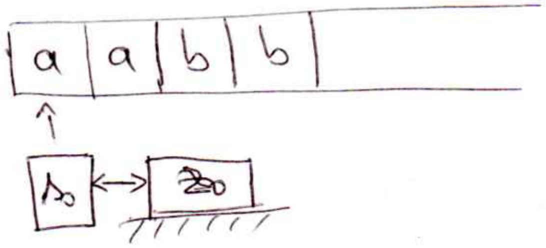
$$f(s_1, \lambda, z_0) = (s_2, \lambda)$$

Diagrama de stări

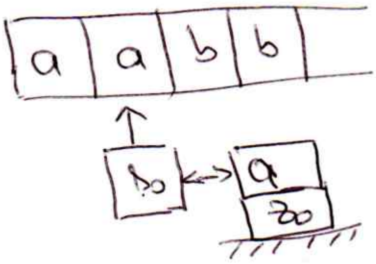


Descrierea funcționării pt. un cuvânt
$p = a^2 b^2$

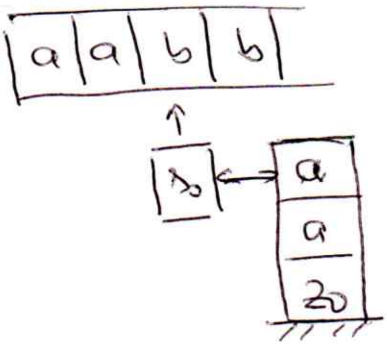
inițial



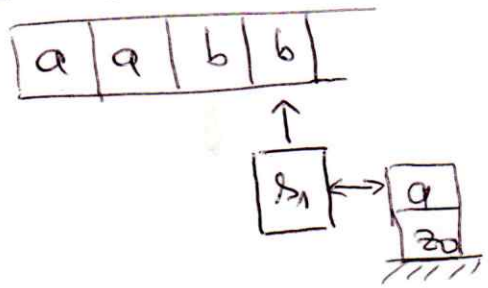
pas 1 (pornirea APD)



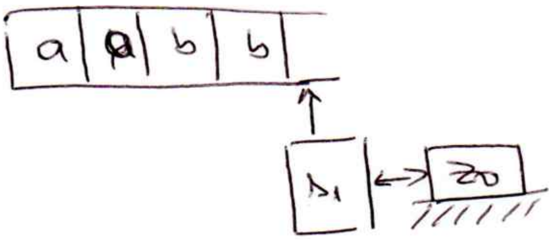
pas 2 (umplere stivă)



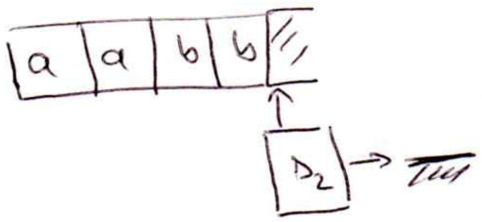
pas 3 (găsire mijloc)



pas 4 (golire stivă)



pas 5 (recunoaștere cuvânt)



$$(s_0, aabb, z_0) \xrightarrow{1)} (s_0, abb, az_0) \xrightarrow{2)} (s_0, bb, aaz_0) \xrightarrow{3} (s_1, b, az_0) \xrightarrow{4)} (s_1, \lambda, z_0) \xrightarrow{5)} (s_2, \lambda, \lambda)$$

$$(s_0, aabb, z_0) \xrightarrow{*} (s_2, \lambda, \lambda)$$

Exemplul 2: Recunoașterea palindroamelor

$$L = \{ w1\tilde{w} \mid w \in \{a,b\}^+ \}$$

#

Similar cu APD precedent - 2 stări s_0 - umplere, s_1 - golire

1) Pornirea automatului

$$f(s_0, x, z_0) = (s_0, xz_0), \quad x \in \{a,b\}$$

2) Umplerea stivei

$$f(s_0, x, y) = (s_0, xy), \quad x, y \in \{a,b\}$$

3) Găsirea mijlocului

$$f(s_0, 1, x) = (s_1, x), \quad x \in \{a,b\}$$

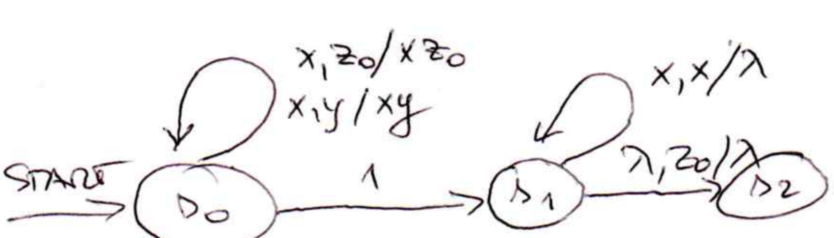
4) Golire stivă

$$f(s_1, x, x) = (s_1, \lambda), \quad x \in \{a,b\}$$

5) Recunoaștere cuvânt

$$f(s_1, \lambda, z_0) = (s_2, \lambda)$$

Diagrama



recunoașterea cuvântului $p = abb1bba$

$(s_0, abb1bba, z_0) \mapsto (s_0, bb1bba, az_0) \mapsto (s_0, b1bba, baz_0) \mapsto$
$\mapsto (s_0, 1bba, bbaz_0) \mapsto (s_1, bba, bbaz_0) \mapsto$
$\mapsto (s_1, ba, baz_0) \mapsto (s_1, a, az_0) \mapsto (s_1, \lambda, z_0) \mapsto (s_2, \lambda, \lambda)$

$$(s_0, abb1bba, z_0) \overset{*}{\mapsto} (s_2, \lambda, \lambda)$$

Exemplul 3: Recunoaștere palindroame de lungime pară

$L = \{ w\tilde{w} \mid w \in \{a,b\}^* \}$

Rezolvare

Obs - trebuie recunoscut λ și nu știu unde e mijlocul.
Ideea - ghicesc mijlocul, unde am două litere adiacente egale

1) Pornirea APD (sau recunoaștere λ)

$f(s_0, x, z_0) = (s_0, x z_0)$, $x \in \{a,b\}$

$f(s_0, \lambda, z_0) = (s_{100}, \lambda)$

2) Umplere stivă

$f(s_0, x, y) = (s_0, xy)$, $x, y \in \{a,b\}$ $x \neq y$

3) Ghicire mijloc sau umplere

$f(s_0, x, x) = \{(s_1, \lambda), (s_0, xx)\}$, $x \in \{a,b\}$

4) Golire stivă

$f(s_1, x, x) = (s_1, \lambda)$, $x \in \{a,b\}$

5) Recunoaștere cuvânt

$f(s_1, \lambda, z_0) = (s_2, \lambda)$

TEOREMĂ: Familia limbajelor independente de context coincide cu familia limbajelor recunoscute de AF.

#

Vom demonstra doar implicația $\mathcal{L}_2 \subseteq \mathcal{L}_{AF}$

Fie $L \in \mathcal{L}_2$ și $G=(V_N, V_T, x_0, P) \in \mathcal{G}_2$ ai $L=L(G)$.
Presupunem că G este în formă normală GREIBACH.
Regulile au forma $A \to ip$, $A \in V_N, i \in V_T, p \in V_N^*$
Admitem și regula de completare $S \to \lambda$, $S \notin dr$.
Construim următorul $APD = (\{s\}, V_T, V_N, f, s, x_0)$
Funcția de evoluție se construiește astfel
Dacă $A \to ip \in P$ atunci $f(s, i, A) \ni (s, p)$
În rest $f(s, j, X) = \phi$, $j \in V_T \cup \{\lambda\}$, $X \in V_N$.

Rămâne de arătat că $L(APD) = L(G)$, adică dubla incluziune.
$L(G) \subseteq L(APD)$

Fie $p \in L(G)$, $p = i_1 \dots i_n$ și $x_0 \overset{*}{\Rightarrow} p$ derivarea extrem stângă ce produce p.

(A) $x_0 \Rightarrow i_1 X_1 u_1 \Rightarrow i_1 i_2 X_2 u_2 u_1 \Rightarrow i_1 i_2 i_3 X_3 u_3 u_2 u_1 \Rightarrow \dots \Rightarrow i_1 \dots i_n$

unde $u_1, u_2, u_3, \dots \in V_N^* = Z^*$

Obs: Aparent, partea $u_s u_{s-1} \dots u_1$ se mărește cu fiecare derivare directă. În realitate, unele din cuvintele u_j sunt vide, atunci când se aplică o regulă de forma $X \to i$; în particular, în ultimele derivări directe se aplică numai astfel de reguli.

Considerând şirul de reguli din derivarea (A) obţinem

$$x_0 \to i_1 X_1 u_1 \Rightarrow (s, X_1 u_1) \in f(s, i_1, x_0)$$

(B) $$X_1 \to i_2 X_2 u_2 \Rightarrow (s, X_2 u_2) \in f(s, i_2, X_1)$$

$$X_2 \to i_3 X_3 u_3 \Rightarrow (s, X_3 u_3) \in f(s, i_3, X_2)$$

$$\dots$$

Prin urmare automatul poate avea următoarea evoluţie:

(C) $$(s, i_1 i_2 i_3 \dots i_n, x_0) \mapsto (s, i_2 i_3 \dots i_n, X_1 u_1) \mapsto$$

$$\mapsto (s, i_3 \dots i_n, X_2 u_2 u_1) \mapsto (s, i_4 \dots i_n, X_3 u_3 u_2 u_1) \mapsto \dots$$

Comparăm evoluţia (C) cu derivarea (A) şi observăm că pe b.i. avem la fiecare pas partea complementară a cuvântului (faţă de derivare) iar în memoria APD se reproduce partea de neterminale din formele propoziţionale. Cum în derivare se ajunge la cuvântul $i_1 \dots i_n$ rezultă că în evoluţie se va ajunge la configuraţia (s, λ, λ).

Deci $p \in L(APD)$ şi $L(G) \subseteq L(APD)$.

Pentru implicaţia inversă $L(APD) \subseteq L(G)$ se consideră $p \in L(APD)$ şi se poate arăta următoarea implicaţie

$$(s, p, u) \overset{*}{\mapsto} (s, \lambda, \lambda) \Rightarrow u \overset{*}{\underset{G}{\Rightarrow}} p, \forall u \in V_N^*$$

Demonstraţie prin inducţie după lg. lui p (vezi curs!)

Ptr. $u = x_0$ se obţine afirmaţia

$$(s, p, x_0) \overset{*}{\mapsto} (s, \lambda, \lambda) \Rightarrow x_0 \overset{*}{\underset{G}{\Rightarrow}} p, \text{ deci } p \in L(G).$$

Obs: Recunoaşterea cuvântului λ pt. un APD înseamnă $f(s, \lambda, x_0) \ni (s, \lambda)$ şi conform formei normale $x_0 \to \lambda$ iar $x_0 \notin dr$

__EXEMPLU__ : Considerăm gramatica

$$S \to aSb \mid ab \in \mathcal{G}_2.$$

Forma normală Greibach este ușor de găsit

$$\begin{cases} S \to aSX \mid aX \\ X \to b \end{cases}$$

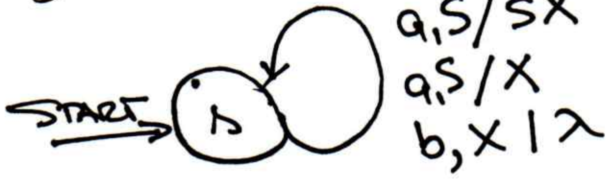
Construim funcția de evoluție pt APD echivalent

$S \to aSX \quad \Rightarrow \quad f(s, a, S) \ni (s, SX)$ 1)

$S \to aX \quad \Rightarrow \quad f(s, a, S) \ni (s, X)$ 2)

$X \to b \quad \Rightarrow \quad f(s, b, X) \ni (s, \lambda)$ 3)

Diagrama de stări



Verificăm recunoașterea / generarea lui a^3b^3

__Derivarea__ (notăm numărul regulii aplicate!)

$$S \overset{1)}{\Rightarrow} aSX \overset{1)}{\Rightarrow} aaSXX \overset{2)}{\Rightarrow} aaaXXX \Rightarrow$$
$$\overset{3)}{\Rightarrow} aaabXX \overset{3)}{\Rightarrow} aaabbX \overset{3)}{\Rightarrow} aaabbb$$

__Recunoașterea__ (pașii corespund aplicării regulilor!)

$$(s, aaabbb, S) \overset{1)}{\longmapsto} (s, aabbb, SX) \overset{1)}{\longmapsto}$$
$$(s, abbb, SXX) \overset{2)}{\longmapsto} (s, bbb, XXX) \overset{3)}{\longmapsto}$$
$$(s, bb, XX) \overset{3)}{\longmapsto} (s, b, X) \overset{3)}{\longmapsto} (s, \lambda, \lambda)$$