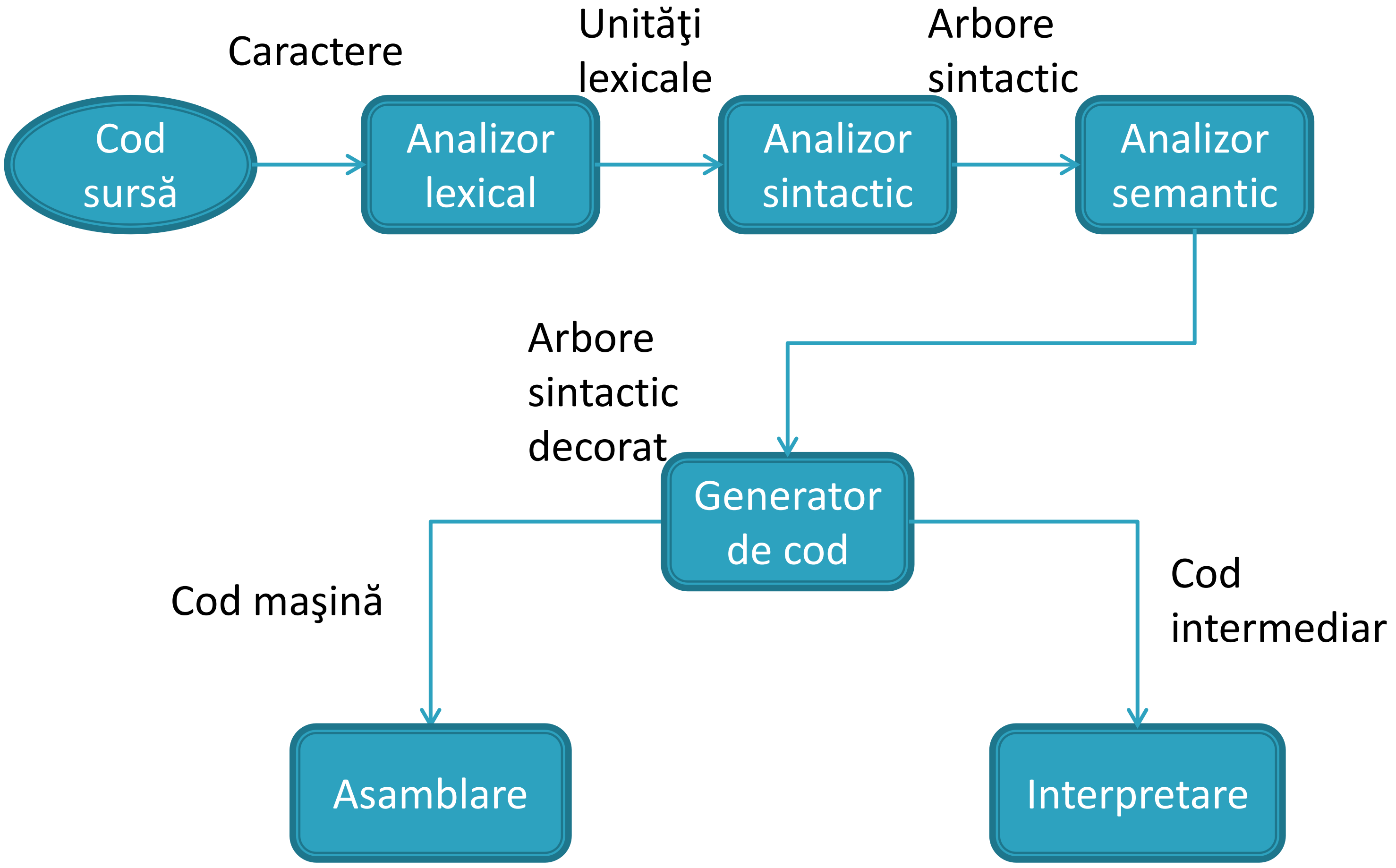
Limbaje formale, automate şi compilatoare

Curs 9

Recapitulare

- Paşii compilării
- Analiza lexicală
  - Descriere lexicală
  - Interpretare
  - Interpretare orientată dreapta
  - Descriere lexicală bine formată
- Lex
- Gramatici de tipul 2

Compilare


Caractere
Unităţi lexicale
Arbore sintactic
Cod sursă
Analizor lexical
Analizor sintactic
Analizor semantic
Arbore sintactic decorat
Generator de cod
Cod maşină
Cod intermediar
Asamblare
Interpretare

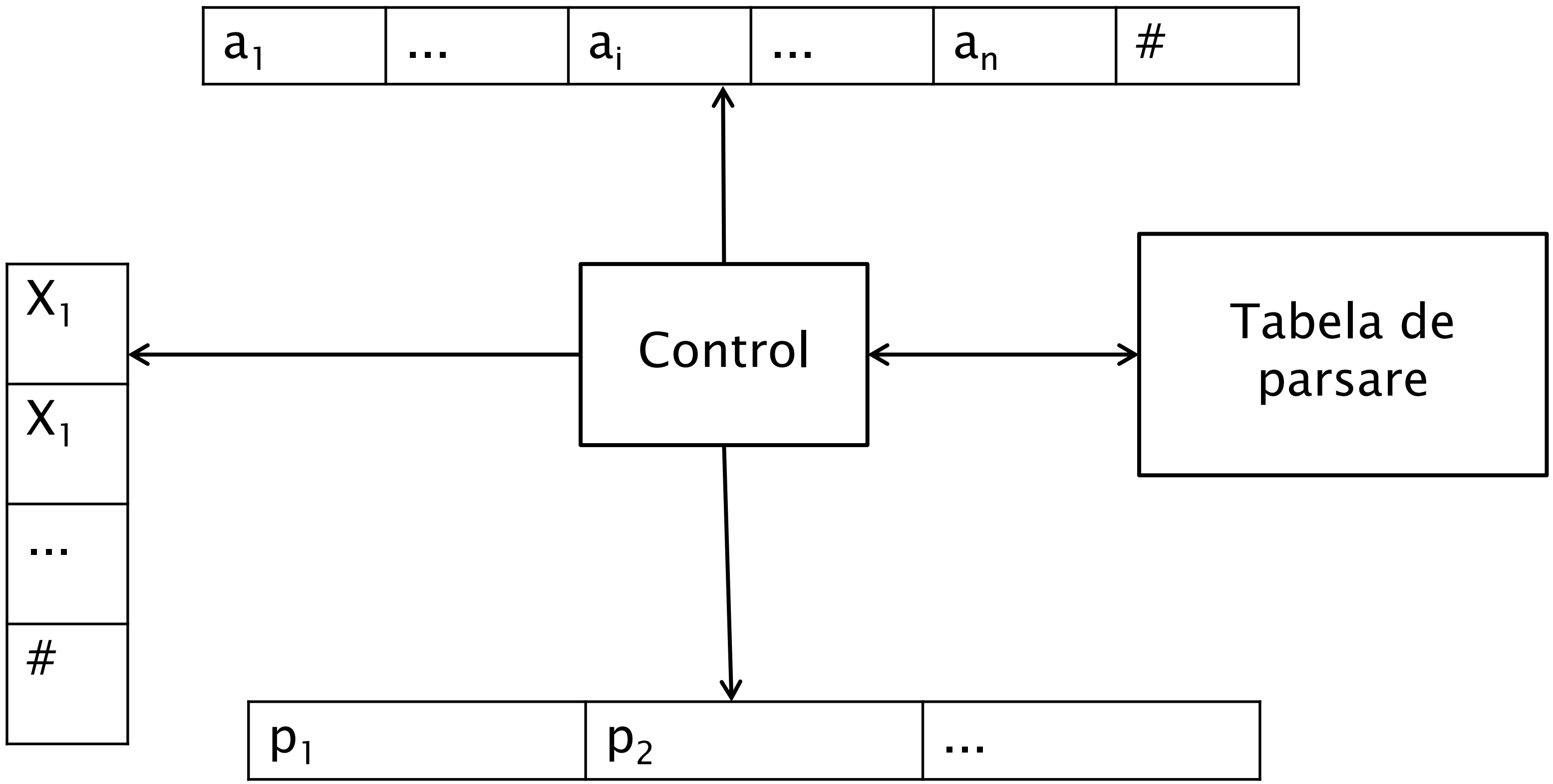
Exemplu de analizor lexical

- Fie descrierea lexicală:
  - litera → a | b |…|z
  - cifra → 0 | 1 |…| 9
  - identificator → litera (litera| cifra)*
  - semn → + | -
  - numar →(semn | ϵ) cifra+
  - operator → + | -| * | / | < | > | <= | >= | < >
  - asignare → :=
  - doua_puncte → :
  - cuvinte_rezervate → if| then|else
  - paranteze →) | (

Cuprins

- Analiza sintactică ascendentă
  - Parser ascendent general
- Gramatici LR(k)
  - Definiţie
  - Proprietăți
- Gramatici LR(0)
  - Teorema de caracterizare LR(0)
  - Automatul LR(0)

Parser ascendent general



Configuraţii

- O configuraţie (#γ, u#, π) este interpretată în felul următor:
  - –#γ este conţinutul stivei cu simbolul # la baza.
  - –u# este conţinutul intrării.
  - –π este conţinutul ieşirii.
- $C_0 = \{(\#, w\#, \varepsilon) | w \in T^*\}$ mulţimea configuraţiilor iniţiale.

Tranziţii

- Parserul ascendent ataşat gramaticii G este perechea (C_0, $\vdash$) unde C_0 este mulţimea configuraţiilor iniţiale, iar $\vdash$ este o relaţie de tranziţie definită astfel:
  - $(\#\gamma, au\#, \pi) \vdash (\#\gamma a, u\#, \pi)$ (***deplasare)*** pentru orice $\gamma \in \Sigma^*$, $a \in T$, $u \in T^*$, $\pi \in P^*$.
  - $(\#\alpha\beta, u\#, \pi) \vdash (\#\alpha A, u\#, \pi r)$ dacă $r = A \rightarrow \beta$ (***reducere).***
  - Configuraţia $(\#S, \#, \pi)$ unde $\pi \neq \varepsilon$, se numeşte ***configuraţie de acceptare.***
  - Orice configuraţie, diferită de cea de acceptare, care nu este în relaţia $\vdash$ cu nici o altă configuraţie este o configuraţie eroare.
- Parsere de deplasare/reducere.

Exemplu

- Fie gramatica $S \rightarrow aSb \mid \varepsilon$. Tranziţiile sunt:
  - $(\#\gamma, u\#, \pi) \vdash (\#\gamma S, u\#, \pi 2)$
  - $(\#\gamma aSb, u\#, \pi) \vdash (\#\gamma S, u\#, \pi 1)$
  - $(\#\gamma, au\#, \pi) \vdash (\#\gamma a, u\#, \pi)$
  - $(\#\gamma, bu\#, \pi) \vdash (\#\gamma b, u\#, \pi)$
- O succesiune de tranziţii se numeşte calcul
  - $(\#, \#, \varepsilon) \vdash (\#S, \#, 2)$
  - $(\#, aabb\#, \varepsilon) \vdash (\#a, abb\#, \varepsilon) \vdash (\#aa, bb\#, \varepsilon) \vdash (\#aaS, bb\#, 2) \vdash (\#aaSb, b\#, 2) \vdash (\#aS, b\#, 21) \vdash (\#aSb, \#, 21) \vdash (\#S, \#, 211)$

Conflicte

- Parserul este nedeterminist:
  - Pentru o configuraţie de tipul $(\#\alpha\beta, au\#, \pi)$, $S\rightarrow\beta$, există două posibilităţi (conflict **deplasare/reducere**):
    - $(\#\alpha\beta, au\#, \pi) \vdash (\#\alpha S, au\#, \pi r)$ (reducere cu $S\rightarrow\beta$)
    - $(\#\alpha\beta, au\#, \pi) \vdash (\#\ \alpha\beta a, u\#, \pi)$ (deplasare)
  - Pentru o configuraţie $(\#\gamma, u\#, \pi)$ cu $\gamma=\alpha_1\beta_1=\alpha_2\beta_2$ şi $A\rightarrow\beta_1$, $B\rightarrow\beta_2$, reguli (conflict **reducere/reducere**)
    - $(\#\alpha_1\beta_1, u\#, \pi) \vdash (\#\ \alpha_1 A, au\#, \pi r_1)$
    - $(\#\alpha_2\beta_2, u\#, \pi) \vdash (\#\ \alpha_2 B, au\#, \pi r_2)$

Corectitudine

- Spunem că un cuvânt $w \in T^*$ este acceptat de un parser ascendent dacă există măcar un calcul de forma
  - $(\#, w\#, \varepsilon) \vdash^+ (\#S, \#, \pi)$
- Pentru ca parserul descris să fie corect, trebuie ca el să accepte toate cuvintele din L(G) şi numai pe acestea.
- **Teorema**
  - Parserul ascendent general ataşat unei gramatici G este corect: pentru orice $w \in T^*$, $w \in L(G)$ dacă şi numai dacă în parser are loc calculul $(\#, w\#, \varepsilon) \vdash^+ (\#S, \#, \pi)$.

Analiza sintactică LR

- Gramatici **LR(k)**:**L**eft to right scanning of the input, constructing a **R**ightmost derivation in reverse, using **k** symbols lookahead
- Definiţie
  - O gramatică G se numeşte gramatică LR(k), $k \geq 0$, dacă pentru orice două derivări de forma:
    - $S' \Rightarrow S \,_{dr}\!\Rightarrow^* \alpha A u \,_{dr}\!\Rightarrow \alpha\beta u = \delta u$
    - $S' \Rightarrow S \,_{dr}\!\Rightarrow^* \alpha' A' u' \,_{dr}\!\Rightarrow \alpha'\beta' u' = \alpha\beta v = \delta v$
  - pentru care $k{:}u = k{:}v$, are loc $\alpha=\alpha'$, $\beta=\beta'$, $A=A'$

Analiza sintactică LR

- **Teorema 1**
  - Dacă G este gramatică LR(k), $k \geq 0$, atunci G este neambiguă.
- Un limbaj L este (în clasa) $\mathcal{LR}(k)$ dacă există o gramatică LR(k) care îl generează
- **Teorema 2**
  - Orice limbaj $\mathcal{LR}(k)$ este limbaj de tip 2 determinist.
- **Teorema 3**
  - Orice limbaj de tip 2 determinist este limbaj LR(1).
- **Teorema 4**
  - Pentru orice limbaj $\mathcal{LR}(k)$, $k \geq 1$, există o gramatică LR(1) care generează acest limbaj, adică $LR(0) \subset LR(1) = LR(k)$, $k \geq 1$.

Gramatici LR(0)

- **Definiţie**
  - Fie G = (V, T, S, P) o gramatică independentă de context redusă. Să presupunem că simbolul • nu este în Σ. Un **articol** pentru gramatica G este o producţie $A \to \gamma$ în care s-a adăugat simbolul • într-o anume poziţie din γ. Notăm un articol prin $A \to \alpha \bullet \beta$ dacă $\gamma = \alpha\beta$. Un articol în care • este pe ultima poziţie se numeşte **articol complet**.
- **Definiţie**
  - Un **prefix viabil** pentru gramatica G este orice prefix al unui cuvânt $\alpha\beta$ dacă $S \;_{dr}\Rightarrow^* \alpha A u \;_{dr}\Rightarrow \alpha\beta u$. Dacă $\beta = \beta_1\beta_2$ şi $\varphi = \alpha\beta_1$ spunem că articolul $A \to \beta_1 \bullet \beta_2$ este **valid** pentru **prefixul viabil** φ.

Exemplu

- Exemplu $S \to A$, $A \to aAa \mid bAb \mid c \mid \varepsilon$.
  - Articole: $S\to\bullet A$, $S\to A\bullet$, $A\to\bullet aAa$, $A\to a\bullet Aa$, $A\to aA\bullet a$, $A\to aAa\bullet$, $A\to\bullet bAb$, $A\to b\bullet Ab$, $A\to bA\bullet b$, $A\to bAb\bullet$, $A\to\bullet c$, $A\to c\bullet$, $A\to \bullet$.
- Articole valide pentru prefixe viabile:

Prefixul viabil	**Articole valide**	**Derivarea corespunzătoare**
ab	$A\to b\bullet Ab$ $A\to\bullet aAa$ $A\to\bullet bAb$	$S\Rightarrow A\Rightarrow aAa\Rightarrow abAba$ $S\Rightarrow A\Rightarrow aAa\Rightarrow abAba\Rightarrow abaAaba$ $S\Rightarrow A\Rightarrow aAa\Rightarrow abAba\Rightarrow abbAbba$
ε	$S\to\bullet A$ $A\to\bullet bAb$ $A\to\bullet c$	$S\Rightarrow A$ $S\Rightarrow A\Rightarrow bAb$ $S\Rightarrow A\Rightarrow c$

Gramatici LR(0)

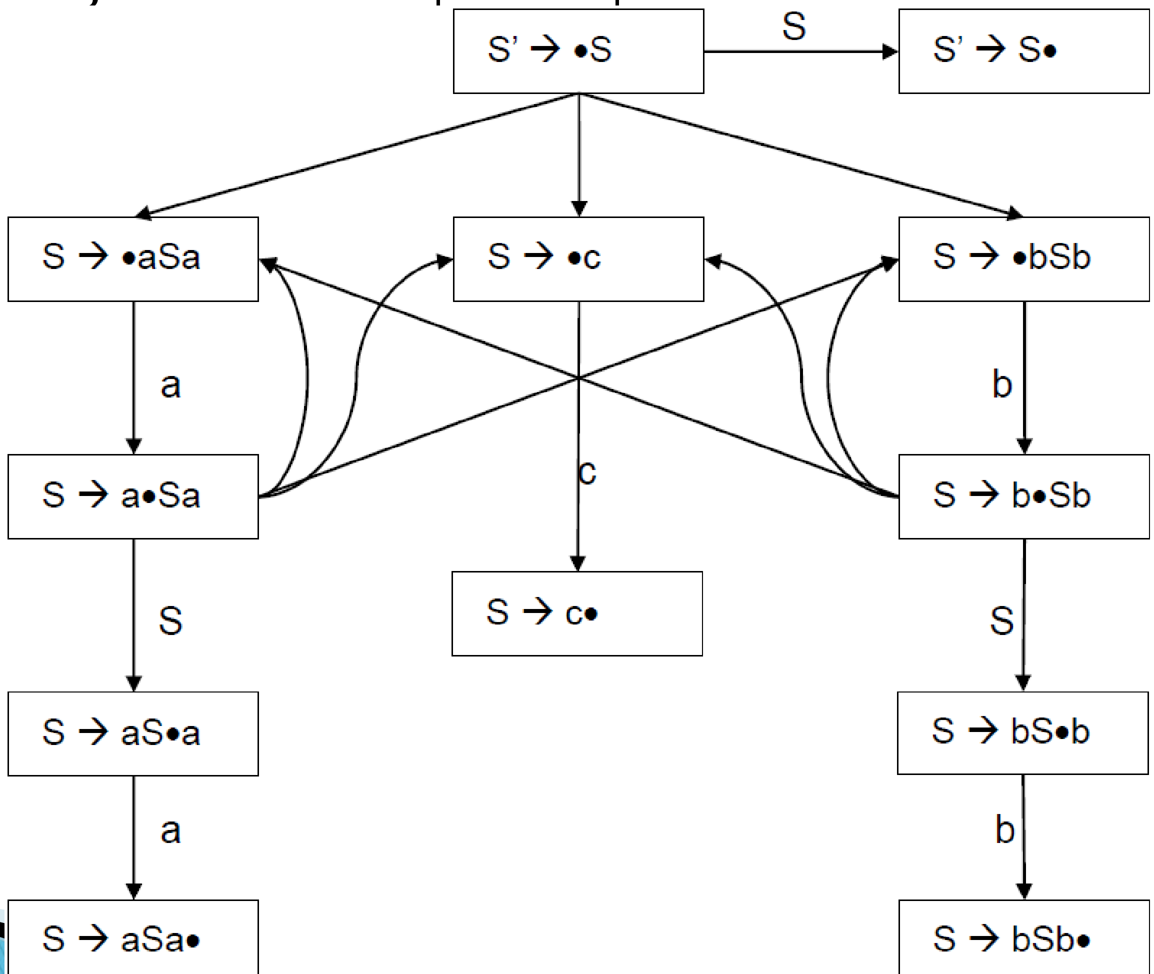
- **Lema**
  - Fie G o gramatică şi $A \to \beta_1 \bullet B\beta_2$ un articol valid pentru prefixul viabil γ. Atunci, oricare ar fi producţia $B \to \beta$, articolul $B \to \bullet\beta$ este valid pentru γ.
- **Teorema** (caracterizare LR(0))
  - Gramatica G este gramatică LR(0) dacă şi numai dacă, oricare ar fi prefixul viabil γ, sunt îndeplinite condiţiile:
    - 1.nu există două articole complete valide pentru γ.
    - 2.dacă articolul $A \to \beta\bullet$ este valid pentru γ, nu există nici un articol $B \to \beta_1 \bullet a\beta_2$, $a \in T$, valid pentru γ.

Gramatici LR(0)

- **Teorema**
  - Fie G = (V, T, S, P) o gramatică independentă de context. Mulţimea prefixelor viabile pentru gramatica G este limbaj regulat.
- **Demonstraţie**
  - G' este G la care se adaugă $S' \to S$.
  - $M = (Q, \Sigma, \delta, q_0, Q)$, unde:
    - Q este mulţimea articolelor gramaticii G',
    - $\Sigma = V \cup T$, $q_0 = S' \to \bullet S$
    - $\delta: Q x(\Sigma \cup \{\varepsilon\}) \to 2^Q$ definită astfel:
      - $\delta(A \to \alpha \bullet B\beta, \varepsilon) = \{B \to \bullet \alpha \mid B \to \gamma \in P\}$.
      - $\delta(A \to \alpha \bullet X\beta, X) = \{ A \to \alpha X \bullet \beta\}$, $X \in \Sigma$.
      - $\delta(A \to \alpha \bullet a\beta, \varepsilon) = \varnothing$, $\forall a \in T$.
      - $\delta(A \to \alpha \bullet X\beta, Y) = \varnothing$, $\forall$ $X, Y \in \Sigma$ cu $X \neq Y$.
- Se arată că are loc:
  - $(A \to \alpha \bullet \beta \in \delta$ ^$(q_0, \gamma) \Leftrightarrow \gamma$ este prefix viabil şi $A \to \alpha \bullet \beta$ este valid pentru γ.

Exemplu

- S’ → S, S → aSa |bSb | c

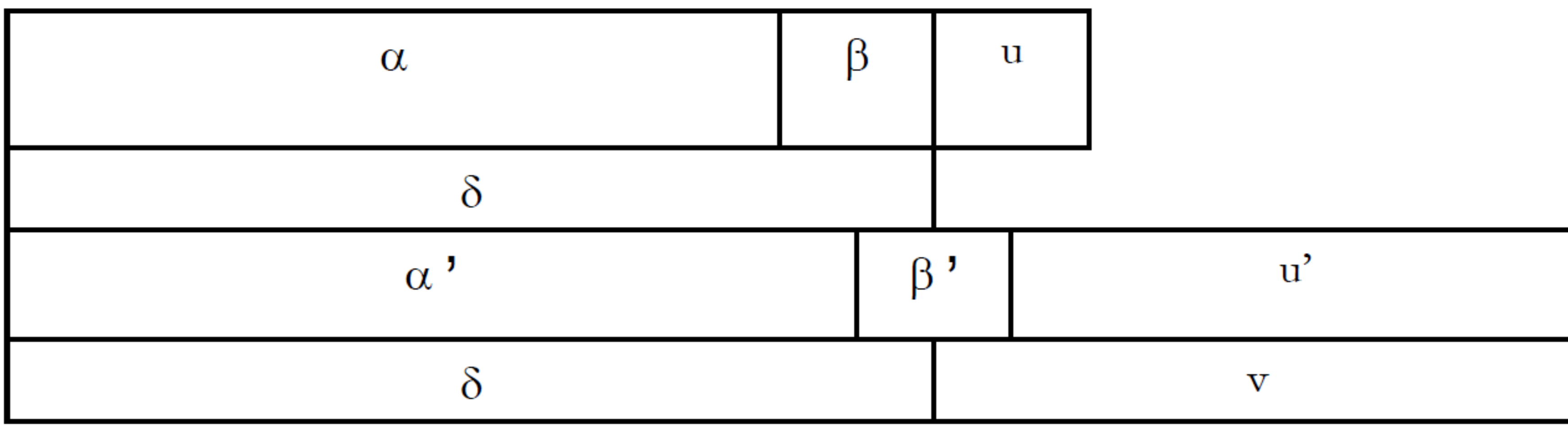


Gramatici LR(0)

- **Teorema** (caracterizare LR(0)). **Demonstraţie**
  - $\Rightarrow$ G este LR(0) şi, prin reducere la absurd 1 sau 2 nu are loc
  - $\Leftarrow$ 1, 2 au loc si prin reducere la absurd, G nu este LR(0): există
    - $S' \Rightarrow S \;_{dr}\Rightarrow^* \alpha A u \;_{dr}\Rightarrow \alpha\beta u = \delta u$
    - $S' \Rightarrow S \;_{dr}\Rightarrow^* \alpha' A' u' \;_{dr}\Rightarrow \alpha'\beta' u' = \alpha\beta v = \delta v$
    - Nu au loc $\alpha=\alpha'$, $\beta=\beta'$, $A=A'$
  - Fără a restrânge generalitatea, presupunem că
    - $|\delta| = |\alpha\beta| \leq |\alpha'\beta'|$

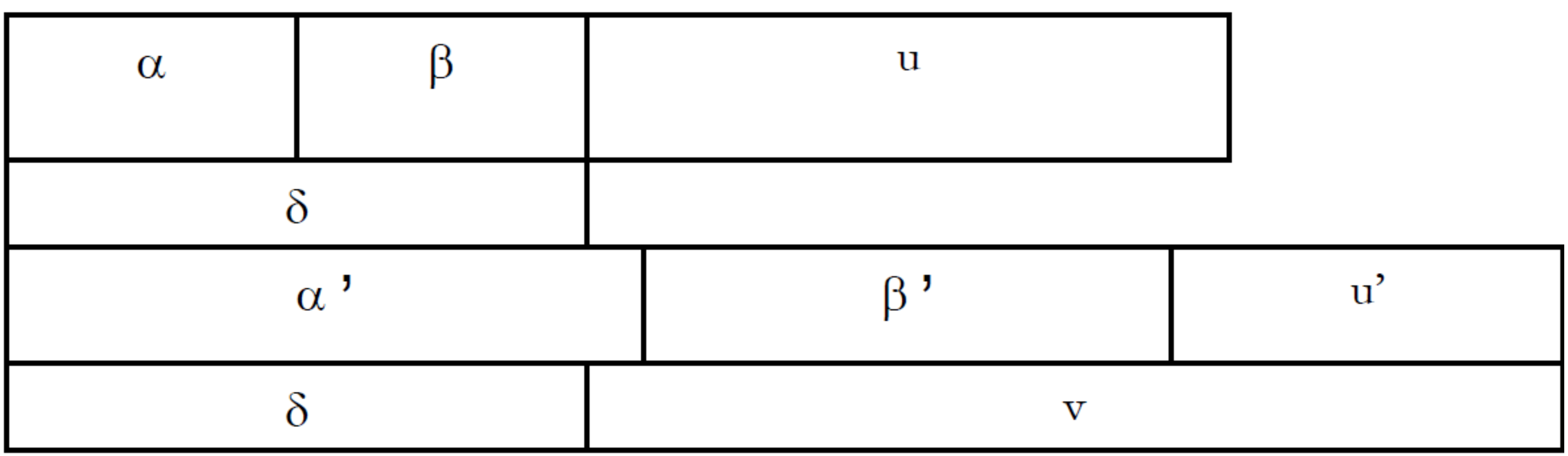
Gramatici LR(0)

- Cazul 1: $|\alpha'| \leq |\delta|$
  - $\beta'=\beta_1'\beta_2'$, $v=\beta_2'u'$, $\beta_2' \in T^*$. Din (1) avem $A \rightarrow \beta \bullet$ articol valid pentru δ şi din (2) avem $A' \rightarrow \beta_1' \bullet \beta_2'$ articol valid pentru δ. Dacă $\beta_2' = \varepsilon$ atunci contrazic (1) iar altfel contrazic (2).



Gramatici LR(0)

- Cazul 2: $|\alpha'| > |\delta|$
  - $\alpha'=\delta u_1$, $v=u_1\beta' u' \in T^*$ ($|u_1|\geq 1$) În derivarea (2), punem în evidenţă prima formă propoziţională care are prefixul δ.
    - $S' \Rightarrow S \underset{dr}{\Rightarrow} \alpha_1 A_1 u_1 \underset{dr}{\Rightarrow} \alpha_1\beta_1 u_1 \underset{dr}{\Rightarrow} \alpha_1\beta'_1\ \beta''_1 u_1 = \delta\beta''_1 u_1 \underset{dr}{\Rightarrow} \delta v$
    - Avem $A\rightarrow\beta\bullet$ şi $A_1\rightarrow\beta'_1\bullet\beta''_1$ articole valid pentru δ



Automatul LR(0)

- Algoritmul 1(procedura închidere(t))
- Intrare:
  - Gramatica G = (V, T, S, P);
  - Mulţimea t de articole din gramatica G;
- Ieşire: $t' = \text{închidere}(t) = \{q \in Q \mid \exists p \in t,\ q \in \delta(p,\varepsilon)\} = \delta(t,\varepsilon)$

Automatul LR(0)

- t' = t ; flag = true;
- while(flag) {
  - flag = false;
  - for ($A \rightarrow \alpha \bullet B\beta \in t'$) {
    - for ($B \rightarrow \gamma \in P$)
      - if ($B \rightarrow \bullet\gamma \notin t'$) {
      - $t' = t' \cup \{B \rightarrow \bullet\gamma\}$;
      - flag = true;
      - }//endif
    - }//endforB
  - }//endforA
- }//endwhile
- return t';

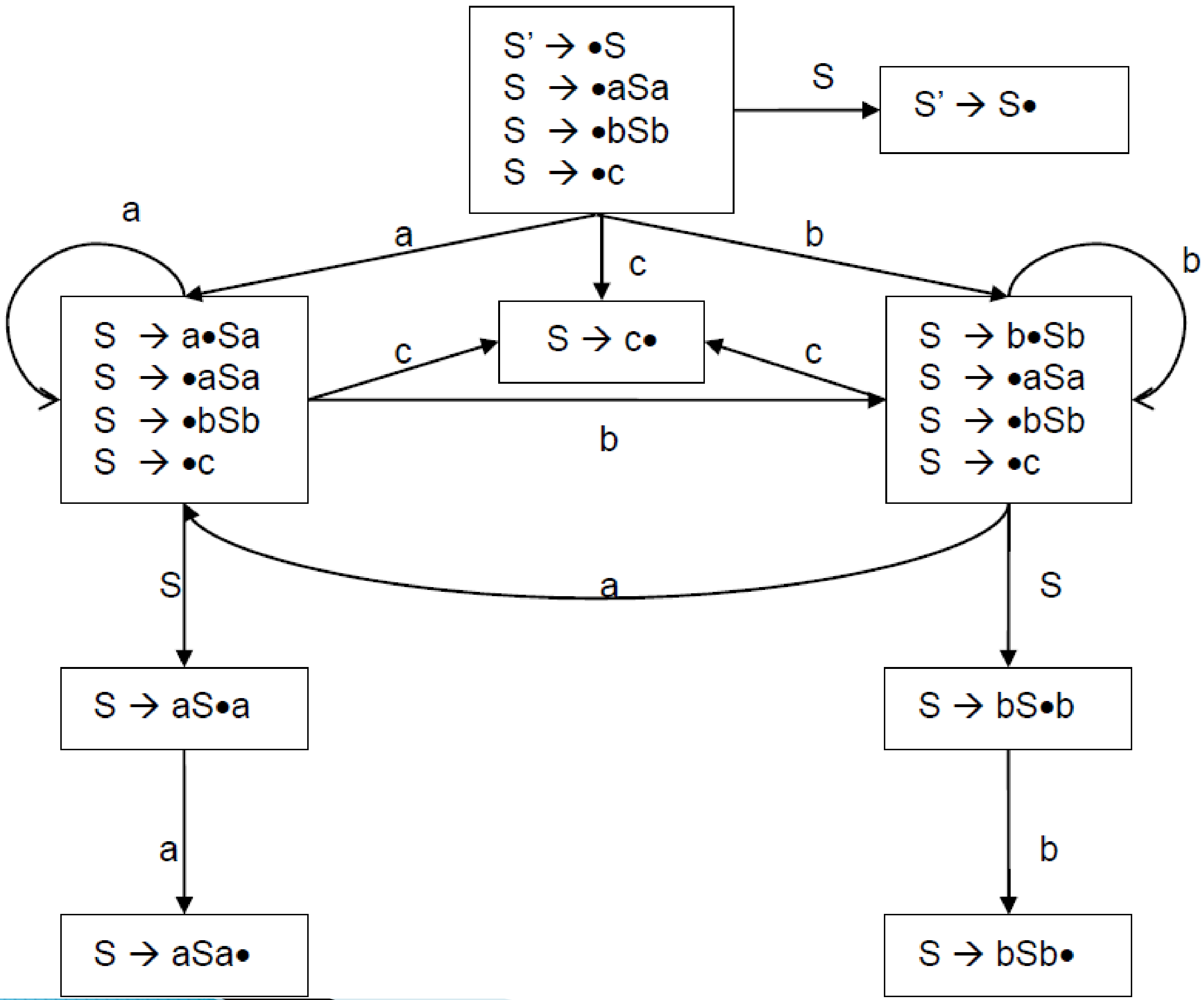
Automatul LR(0)

- Algoritmul 2 Automatul LR(0)
  - Intrare:Gramatica G = (N, T, S, P) la care s-a adăugat S’ → S;
  - Ieşire:Automatul determinist M = (T, Σ, g, t_0, T) echivalent cu M.

- t0=închidere($S' \rightarrow \bullet\ S$); T={$t_0$}; marcat($t_0$)=false;
- while($\exists$ t ϵ T && !marcat(t)) { // marcat(t) = false
  - for(X ϵ Σ) {// $\Sigma = N \cup T$
    - t' = $\emptyset$;
    - for($A \rightarrow \alpha \bullet X\beta$ ϵt)
      - t' = t' $\cup$ {$B \rightarrow \alpha X \bullet \beta$ | $A \rightarrow \alpha \bullet X\beta$ ϵt};
      - if(t'$\neq \emptyset$){
        - t' = închidere(t');
        - if(t'$\notin$T) {
          - T = T$\cup${ t' };
          - marcat(t') = false;
        - }//endif
      - g(t, X) = t';
      - }//endif
    - }//endfor
  - }//endfor
  - marcat(t) = true;
- }// endwhile

Automatul LR(0) – Exemplu

- S’ → S, S → aSa | bSb | c



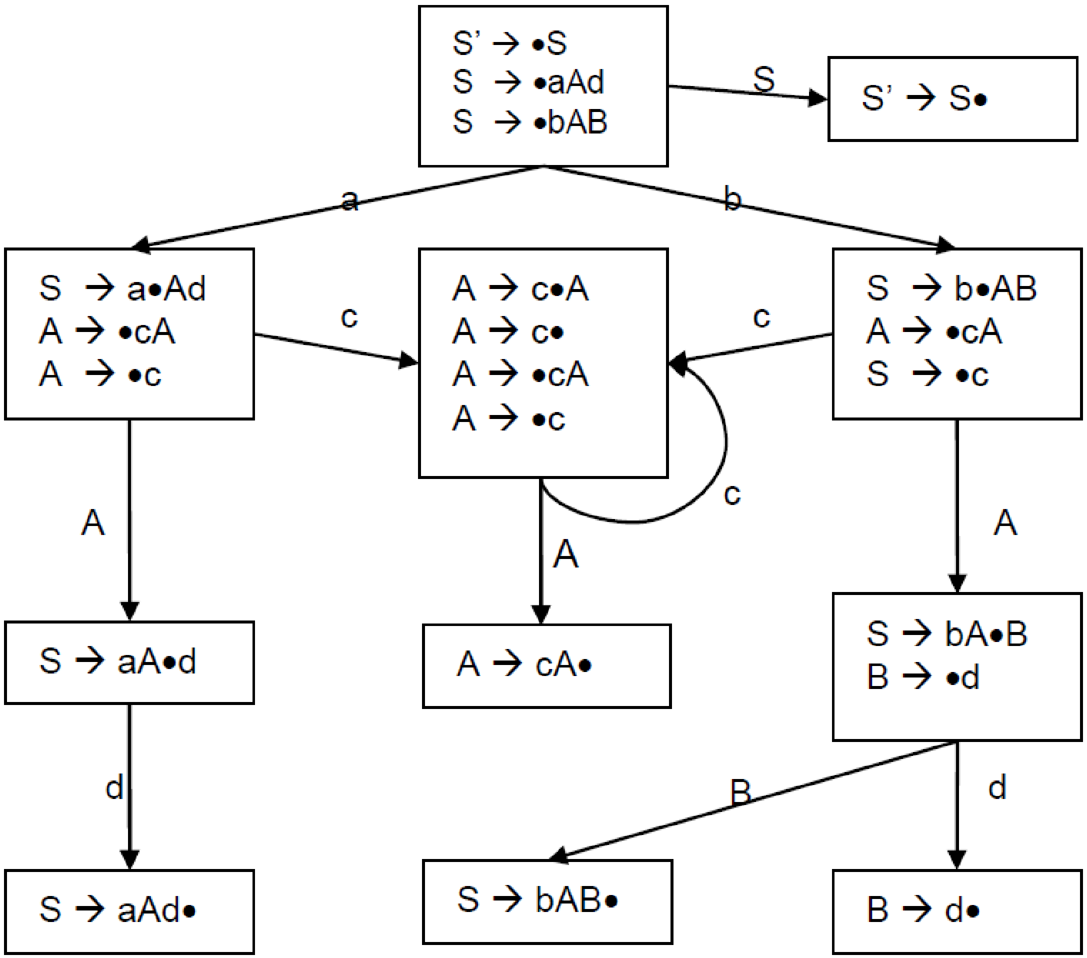
Test LR(0)

- **Definiţie** Fie G o gramatică şi M automatul LR(0) ataşat lui G.
  - Spunem că o stare a lui M are un conflict **reducere/reducere** dacă ea conţine două articole complete distincte $A \to \alpha \bullet$, $B \to \beta \bullet$.
  - Spunem că o stare a lui M are un conflict **deplasare/reducere** dacă ea conţine un articol complet $A \to \alpha \bullet$ şi un articol cu terminal după punct de forma $B \to \beta \bullet a\gamma$.
  - Spunem că o stare este **consistentă** dacă ea nu conţine conflicte şi este **inconsistentă** în caz contrar.

- **Teorema** Fie G o gramatică şi M automatul său LR(0). Gramatica G este LR(0) dacă şi numai dacă automatul M nu conţine stări inconsistente

Exemplu

- $S \to aAd \mid bAB, \; A \to cA \mid c, \; B \to d$



Bibliografie

- Grigoraş Gh., *Construcţia compilatoarelor. Algoritmi fundamentali*, Editura Universităţii "Alexandru Ioan Cuza", Iaşi, 2005