Gramatici de atribute. Automate Push-Down.

LFTC Curs 12

Gramatici de atribute

- Modalitate eficienta de specificare a calculelor si a dependentelor dintre calcule dintr-un arbore de sintaxa abstracta, stabilind ordinea corecta de executie a lor

- G.i.c. sunt extinse cu doua caracteristici:
  - Fiecarui **simbol** X i se asociaza **atribute**
  - Fiecarui **productii** i se asociaza **reguli de calcul a atributelor**
- Astfel se defineste informatia din arborele de sintaxa abstracta necesara analizei semantice

Gramatici de atribute

- GA=(G, A, R) se numeste gramatica de atribute (gramatica atributiva) daca si numai daca:
  - G=(N, Σ, P, S) este o gramatica independenta de context
  - $A = \bigcup_{X \in N \cup \Sigma} A(X)$ este o multime finita de atribute: pentru fiecare simbol al gramaticii se definesc zero sau mai multe atribute, care sunt folosite pentru a memora informatii semantice atasate nodurilor din arborele de sintaxa abstracta ; toate nodurile corespunzatoare unui aceluiasi simbol vor avea aceleasi atribute, cu valori diferite;
  - $R = \bigcup_{p \epsilon P} R(p)$ este o multime finita de reguli, numite reguli de evaluare a atributelor; pentru fiecare productie $A \rightarrow X_1 X_2 \ldots Xn$ se asociaza o multime de reguli de evaluare a atributelor, continand expresii ale valorilor atributelor asociate simbolurilor A si X_i , i=1,n

Tipuri de atribute

- **Atribut sintetizat**: daca regula de evaluare pentru productia $A \rightarrow X_1 X_2 \ldots Xn$ atribuie valoare atributului lui A

- **Atribut mostenit**: daca regula de evaluare corespunzatoare productiei $A \rightarrow X_1 X_2 \ldots Xn$ atribuie valoare atributelor lui X_i, i=1,n (valoarea atributului este mostenita de la parinti).

Gramatica de atribute completa

› O gramatica de atribute este completa daca pentru orice simbol $X \in N$:

1. Pentru orice productie $X \rightarrow \alpha$ a gramaticii multimea R(p) contine o regula de evaluare a atributelor sintetizate ale lui X
2. Pentru orice productie $Y \rightarrow \alpha X \beta$, multimea R(p) contine o regula de evaluare a atributelor mostenite ale lui X.

Exemplu:

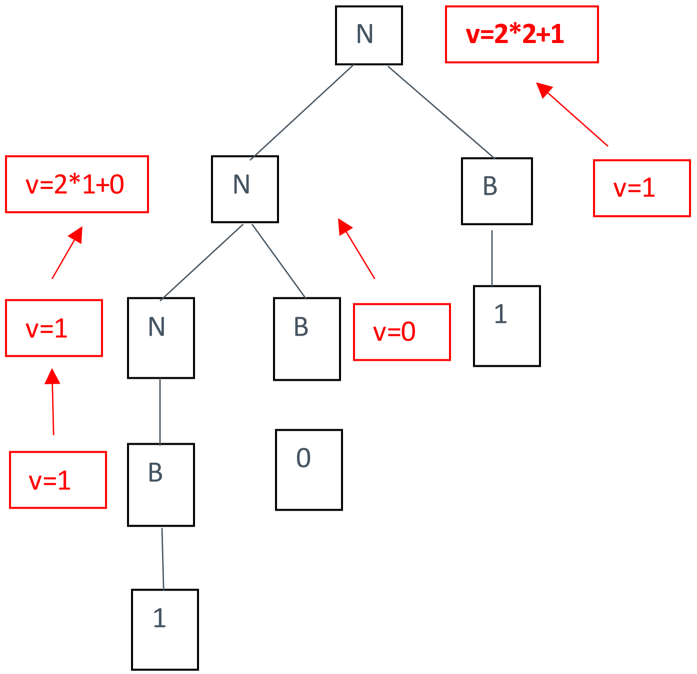
- *G=({B,N}, {0,1}, P,N)*
- $N \rightarrow NB|B$
- $B \rightarrow 0|1$

Atribute pentru determinarea valorii zecimale a unui numar binar?

Pentru numarul binar $b_n \ldots b_1 b_0$ valoarea zecimala $v = bn \times 2^n + \cdots + b_1 \times 2^1 + b_0 \times 2^0$

$N_1 \rightarrow N_2 B$ $\{N_1.v = 2 \times N_2.v + B.v\}$
$N \rightarrow B$ $\{N.v = B.v\}$
$B \rightarrow 0$ $\{B.v = 0\}$
$B \rightarrow 1$ $\{B.v = 1\}$

Evaluarea atributelor



101

Gramatici L-atributate

- L –atributate (Left-attributed)
- permit evaluarea atributelor intr-o singura traversare de la stanga la dreapta a arborelui de sintaxa abstracta
- Pot fi usor incorporate intr-un analizor sintactic descendent
- Majoritatea gramaticilor limbajelor de programare indeplinesc conditia de a fi L-atributate

Gramatica L-atributata

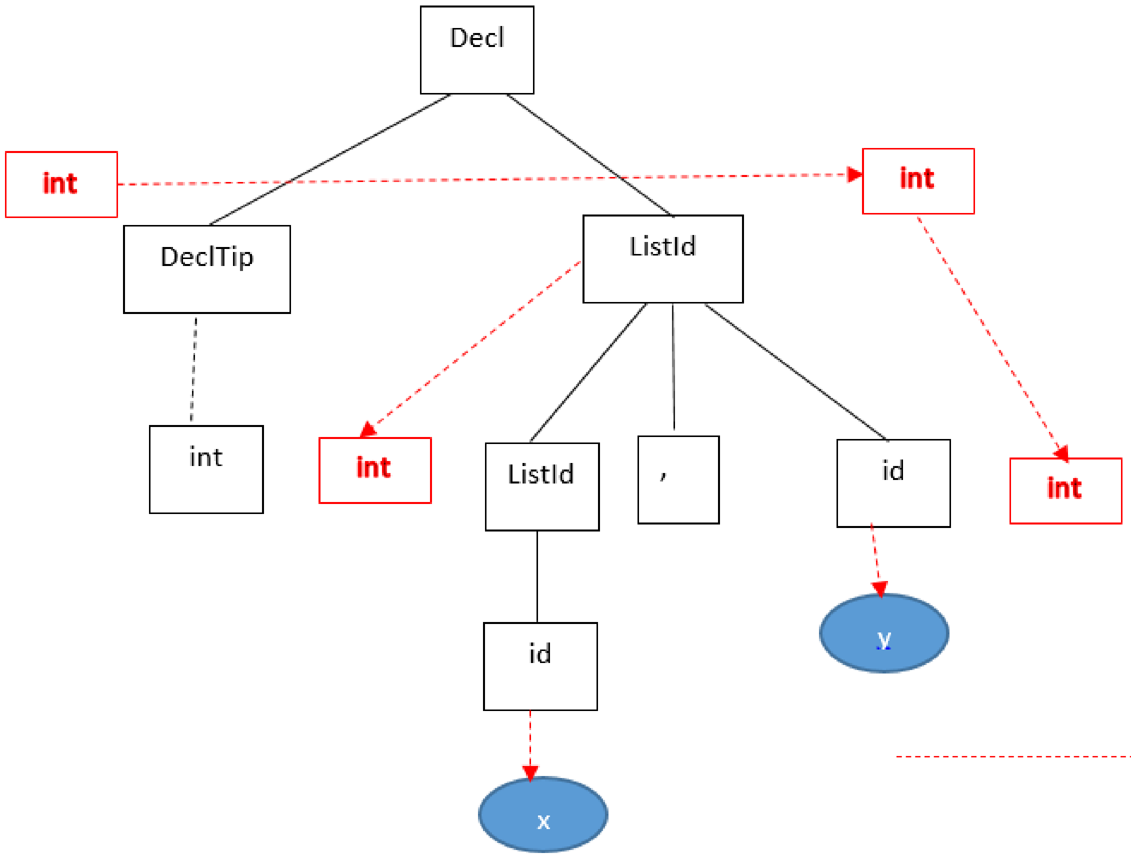
- GA=(G, A, R) se numeste gramatica L-atributata daca si numai daca GA este o gramatica de atribute si pentru orice productie $A \rightarrow X_1 X_2 \dots Xn$ regulile de calcul a atributelor respecta urmatoarele conditii:
  - Orice atribut mostenit al lui X_i depinde doar de atribute mostenite ale lui A si atribute ale simbolurilor $X_1X_2 \dots X_{i-1}$ (noduri situate la stanga in arbore)
  - Orice atribut sintetizat al lui A nu depinde de alte atribute sintetizate ale lui A.

Exemplu

- $Decl \rightarrow DeclTip\ ListId\ \{ListId.tip = DeclTip.tip\}$
- $ListId \rightarrow id\ \{id.tip = ListId.tip\}$
- $ListId_1 \rightarrow ListId_2, id\ \{ListId_2.tip = ListId_1.tip;$
  $id.tip = ListId_1.tip\}$
- $DeclTip \rightarrow int\ \{DeclTip.tip \rightarrow int\}$
- $DeclTip \rightarrow char\{DeclTip.tip \rightarrow char\}$
- $DeclTip \rightarrow bool\{DeclTip.tip \rightarrow bool\}$
- …

Gramatica L-atributata

int x,y



Ordinea de evaluare a atributului *tip*

Gramatici S-atributate

- Clasa speciala de gramatici de atribute
- Nu contin atribute mostenite
- Evaluarea atributelor sintetizate se realizeaza usor atat in analizoarele sintactice ascendente cat si in analizoarele sintactice descendente

Lista de declaratii C++ - calculul spatiului de memorare necesar

Exemplu:

- $ListDecl_1 \rightarrow ListDecl_2; Decl \; \{ListDecl_1.dim = ListDecl_2.dim + Decl.dim\}$
- $ListDecl \rightarrow Decl \; \{ListDecl.dim = Decl.dim\}$
- $Decl \rightarrow Tip \; ListId \; \{Decl.dim = Tip.dim \times ListId.nr\}$
- $Tip \rightarrow int \; \{Tip.dim = 4\}$
- $Tip \rightarrow long \; \{Tip.dim = 8\}$
- $ListId \rightarrow id \; \{ListId.nr = 1\}$
- $ListId_1 \rightarrow ListId_2, id \; \{ListId_1.nr = ListId_2.nr + 1\}$

AUTOMATE PUSH-DOWN

- Un automat **push-down (APD)** este un ansamblu $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ unde:
  - Q este o multime finita si nevida de elemente numite stari;
  - Σ este un alfabet denumit alfabetul de intrare
  - Γ – este un alfabet numit alfabetul memoriei stiva
  - $q_0 \in Q$ – starea initiala
  - $Z_0 \in \Gamma -$ este simbolul de start al memoriei stiva
  - $F \subseteq Q,$ F este multimea starilor finale
  - $\delta: Q \times \Sigma \cup \{\varepsilon\} \times \Gamma \rightarrow P_0(Q \times \Gamma^*)$ este functia de tranzitie

APD determinist

- Fie $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$. Spunem ca acest automat este **determinist** daca:

1. $\forall q \in Q$ si $Z \in \Gamma$, in cazul ca $\delta(q, \varepsilon, Z)$ este nevida, atunci $\delta(q, a, Z) = \emptyset, \forall a \in \Sigma$;
2. $|\delta(q, a, Z)| \leq 1, \forall q \in Q, \forall a \in \Sigma \cup \{\varepsilon\}, \forall Z \in \Gamma$.

In caz contrar, automatul este **nedeterminist**.

Configuratie

- $(q, w, \alpha) \in Q \times \Sigma^{*} \times \Gamma^{*}$
- Automatul este in starea q, pe banda de intrare urmeaza sa se accepte secventa w, iar in memoria stiva avem secventa α
- Configuratia initiala: (q_0, w, Z_0)
- Configuratia finala dupa **criteriul starii finale**: $(q, \varepsilon, \gamma), q \in F$
- Configuratia finala dupa **criteriul stivei vide**: $(q, \varepsilon, \varepsilon)$

Tranzitii

- Tranzitia directa $\vdash$
- $(q, ax, Z\alpha) \vdash (p, \begin{Bmatrix} x \\ ax \end{Bmatrix}, \gamma\alpha)$ daca si numai daca $\delta(q, a, Z) \ni (p, \gamma)$ sau $\delta(q, \varepsilon, Z) \ni (p,\ \gamma)$

- k tranzitia
- + tranzitia
- * tranzitia

Limbajul acceptat de un APD

- Limbajul acceptat de un APD dupa criteriul starii finale este:
- $T_1(M) = \{w | w \in \Sigma^*, (q_0, w, Z_0) \overset{*}{\vdash} (q, \varepsilon, \gamma), q \in F, \gamma \in \Gamma^*\}$

- Limbajul acceptat de un APD dupa criteriul stivei vide este:
- $T_2(M) = \{w | w \in \Sigma^*, (q_0, w, Z_0) \overset{*}{\vdash} (q, \varepsilon, \varepsilon), q \in Q\}$

Reprezentarea

$\boldsymbol{Q} \times \boldsymbol{\Gamma}$		$\boldsymbol{\Sigma} \cup \{\boldsymbol{\varepsilon}\}$			
		a	b	…	$\boldsymbol{\varepsilon}$
q_0	Z_0	$\delta(q_0, a, Z_0)$	$\delta(q_0, b, Z_0)$		$\delta(q_0, \boldsymbol{\varepsilon}, Z_0)$
	…				
	Z_n	$\delta(q_0, a, Zn)$	$\delta(q_0, b, Zn)$		$\delta(q_0, \boldsymbol{\varepsilon}, Zn)$
…					
q_n	Z_0	$\delta(qn, a, Z_0)$	$\delta(qn, b, Z_0)$		$\delta(qn, \boldsymbol{\varepsilon}, Z_0)$
	Z_n	$\delta(qn, a, Z_0)$	$\delta(qn, b, Z_0)$		$\delta(qn, \boldsymbol{\varepsilon}, Z_0)$

Exemplu

- Construiti APD care accepta limbajul $L = \{0^n 1^n | n \geq 1\}$.
- Stari?

q_0 – citire simboluri 0 – la primul 1 citit se trece in starea q_1
q_1- citire simboluri 1

$M=(\{q_0,q_1\}, \{0,1\}, \{Z_0,A\},\delta,q_0,Z_0,\{q_1\})$

- Stiva?

Initial Z_0
La fiecare 0 citit adaugam un A pe stiva
La fiecare 1 citit scoatem un A de pe stiva

- Tranzitii?

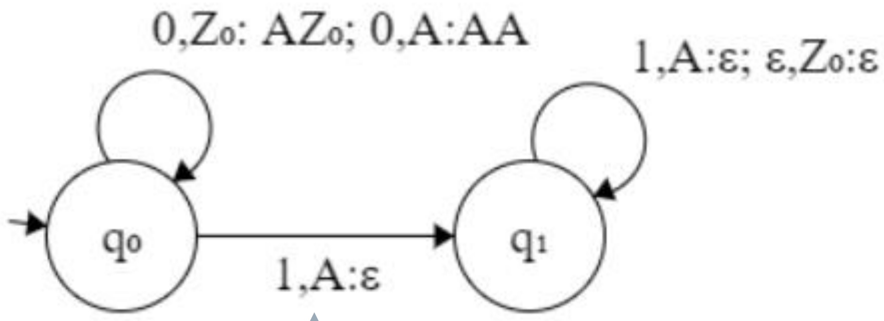
$$\delta(q_0,0,Z_0) = \{(q_0,AZ_0)\}$$
$$\delta(q_0,0,A) = \{(q_0,AA)\}$$
$$\delta(q_0,1,A) = \{(q_1,\varepsilon)\}$$
$$\delta(q_1,1,A) = \{(q_1,\varepsilon)\}$$
$$\delta(q_1,\varepsilon,Z_0) = \{(q_1,\varepsilon)\}$$

$(q_0,0011,Z_0) \vdash (q_0,011,AZ_0) \vdash (q_0,11,AAZ_0) \vdash (q_1,1,AZ_0)) \vdash (q_1,\varepsilon,Z_0)) \vdash (q_1,\varepsilon,\varepsilon)$

Reprezentare grafica

$\delta(q_0, 0, Z_0) = \{(q_0, A)\}$

$\delta(q_0, 0, A) = \{(q_0, AA)\}$



$\delta(q_0, 1, A) = \{(q_1, \varepsilon)\}$