Grammatica ad Attributi: Formalizzazione

- Formaliziamo alcune strutture importanti:
  - Alberi e alberi etichettati
  - Alberi di derivazione sintattica: T(G)
  - Alberi di sintassi astratta: T_Σ
  - Alberi Attributati: T^A
  - Grammatiche ad attributi: $L(G^A)$

Source: Marco Bellia - Dip. Informatica, Univ. Pisa

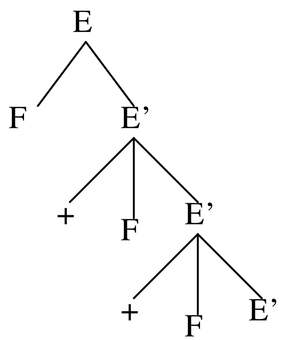
Relazione binaria: albero, albero etichettato

Alberi (e relazioni binarie):

- $T = \langle R, E \subseteq R^2 \rangle$
- $T = \langle R, E \subseteq R^2, L{:}R \to U \rangle$ -- *etichettati su U*

Dove: R: *insieme nodi* E: *insieme archi (orientati)* L: *funzione etichette su U*

Operazioni: root*: T --> R,* arity: *R --> N,* son*: T* x *N --> T*
Selettori: R(T)= R, E(T)=E, L(T)=L
Abbreviazioni: sons: *T --> T^N*, L: *T^N --> U^N*



R(T) = {0,1,2,3,4,5,6,7,8}
L(T) = {(0,E),(1,F),(2,E'),(3,+),(4,F),(5,E'),(6,+),(7,F),(8,E')}
E(T) = {(01),(0,2),(2,3),(2,4),(2,5),(5,6),(5,7),(5,8)}
T = <R(T),E(T),L(T)>

Source: Marco Bellia - Dip. Informatica, Univ. Pisa

Grammatiche e Alberi di Derivazione Sintattica

Alberi di derivazione sintattica

Definizione. Immagine I(G) della chiusura transitiva, =>*, della relazione binaria su T(G), "=>", definita dalle produzioni di una grammatica.

Dove:

$G \equiv \langle \Sigma, V, s \in V, P \equiv \{s_i ::= x_{i,1} \ldots x_{i,ji} \mid i \leq |P|\} \rangle$

$T(G) \equiv \langle R, E \subseteq R^2, L: R \rightarrow \Sigma \cup V \rangle$ *s.t.:* $|R| \geq |\Sigma \cup V|$

$E \equiv \{(l,r) \mid L(l)=s_i \;\&\; L(r)=x_{i,j}$ *per* $j \leq ji \;\&\; s_i ::= x_{i,1} \ldots x_{i,ji} \in P\}$

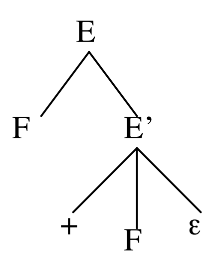
$\Rightarrow\; \equiv \{(t,u) \mid arity(root(t))=0 \& L(root(t))=s_i=L(root(u)) \& L(sons(u))=x_{i,1} \ldots x_{i,ji}$

per qualche $s_i ::= x_{i,1} \ldots x_{i,ji} \in P$, *oppure*

$son(t,j)=son(u,j)\ (\forall j \in 1..k \backslash i) \&\ son(t,i) \Rightarrow son(u,i)$

per qualche $1 \leq i \leq k]$

$I(G) \equiv \{t \mid t_s \Rightarrow^* t \;\&\; arity(root(t_s))=0 \;\&\; L(root(t_s))=s\}$



E::= F E'
E'::= + F E'
E'::= ε

Source: Marco Bellia - Dip. Informatica, Univ. Pisa

Alberi di Sintassi Astratta

Alberi di Sintassi Astratta

Definizione. Relazione binaria, antisimmetrica, “>” su Termini s.t:

$$f_k(t_1,\ldots,t_k)>t_i \qquad (\forall\, 1\leq i\leq k \;\&\; f_k(t_1,\ldots,t_k)>t_i \in Termini)$$

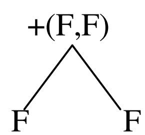
Equivalentemente: $> \equiv \{(f_k(t_1,\ldots,t_k),t_i) \mid 1\leq i\leq k,\ f_k(t_1,\ldots,t_k),t_i \in T_\Sigma\}$

Dove: $T_\Sigma \equiv \{f_k(t_1,\ldots,t_k) \mid f_k \in \Sigma \;\&\; t_1,\ldots,t_k \in T_\Sigma\}$ *--- Termini su* Σ

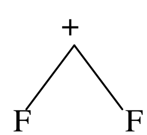
Equivalentemente: $> \equiv \langle R,\ E\subseteq R^2, L{:}R\text{->}T_\Sigma\rangle$ *s.t.:* $|R| \geq |T_\Sigma|$

Dove: $E \equiv \{(l,r) \mid L(l)=f_k(t_1,\ldots,t_k) \in T_\Sigma \;\&\; L(r)=t_i \in T_\Sigma\}$

$\Sigma = \{F_0, +_2\}$
$T_\Sigma ::= \{F\} \cup \{+(u,t) \mid u,t \in T_\Sigma\}$



oppure



con una differente L

Source: Marco Bellia - Dip. Informatica, Univ. Pisa

Grammatica ad Attributi: Formalizzazione

$T = \langle R, E \subseteq R^2, L: R \to U \rangle$ --- *etichettati su U*

$T^A = \langle T, L_{\{ai\}}: R(T) \to U_{\{ai\}} \rangle$ --- *con attributi* $\{a_i\}$ *a valori su* $U_{\{ai\}}$

Grammatica: $G = \{\Sigma, V, s, P\}$

- $P \subseteq V \times E_{\Sigma+V}$
- $L(G) = I(G)$

Grammatica Attributata: $G^A \equiv \{\Sigma, V, s, P^A, \{a_i\}\}$

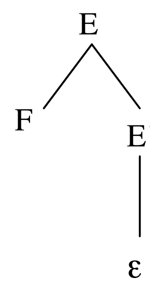
- $G^A\!\downarrow\ \equiv \{\Sigma, V, s, P\} = G$, *con* $P = P^A\!\downarrow$
- $P^A \equiv P \times (\{a_i\} \times Meta_{\{ai\}})$
- $L(G^A) \equiv \{\langle T, L_{\{ai\}} \rangle \mid T \in L(G^A\!\downarrow)\ \&\ \forall r_0 \in R(T)$:
  - $son(r_0) = r_1, \ldots, r_k$, $(\forall i \in 0..n) L(r_i) = B_i$, $B_0 ::= B_1, \ldots, B_k \{B_i.a_j = e_{i,j}\}^{i \in 0..n} \in P^A$
  - $L_{aj}(r_i) \equiv Sem_{Meta}(e_{i,j}\, [L_{ak}(r_h)/B_h.a_k]^{(h \neq i)\ \vee (k \neq j)})\}$

    - *E[V/x]* significa *E* con *x* rimpiazzata da *V*
    - $F^{c(i)}$ significa più occorrenze di *F* ciascuna con *i* rimpiazzato da un valore per cui *c(i)*

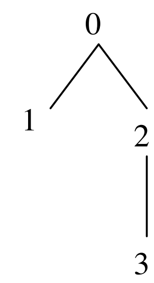
Source: Marco Bellia - Dip. Informatica, Univ. Pisa

Grammatica ad Attributi: Applichiamo la formalizzazione

$E ::= F\ E' \ \{E'.d = E.d+1\}$
$E'_1 ::= +\ F\ E'_2 \ \{E'_2.d = E'_1.d+1\}$
$E' ::= \varepsilon \ \{\varepsilon.d = E'.d\}$



parse tree



Il vero albero T

$$
\begin{aligned}
T^A = \{ &<R=\{0,1,2,3\},\ E,\ L=\{(0,E),(1,F),(2,E'),(3,\varepsilon)\}>,\\
& L_d=\{(0,Sem_{Meta}(\perp)),\\
& \quad (2,Sem_{Meta}((E.d+1)[L_d(0)/E.d])),\\
& \quad (3,Sem_{Meta}((E'.d)[L_d(2)/E'.d]))\}\}
\end{aligned}
$$

Source: Marco Bellia - Dip. Informatica, Univ. Pisa