

Definizione di un linguaggio

Linguaggio di programmazione

- Un linguaggio di programmazione è una notazione formale che può essere usata per descrivere algoritmi.
- Due aspetti del linguaggio:
 - sintassi
 - semantica

Sintassi e semantica

- Sintassi: l'insieme di regole formali per la scrittura di programmi in un linguaggio, che dettano le modalità per costruire frasi corrette nel linguaggio stesso
- Semantica: l'insieme dei significati da attribuire alle frasi (sintatticamente corrette) costruite nel linguaggio
- NB: una frase può essere sintatticamente corretta e tuttavia non avere significato!

Sintassi di un linguaggio

- Vocabolario V : insieme di simboli
- Insieme di tutte le sequenze finite di simboli in $V:V^*$ (universo linguistico)
- Linguaggio L sul vocabolario $V: L \subseteq V^*$
 - Es. (λ parola nulla), $V=\{p,c\}$, imponendo p solo in posizione pari
 $L=\{\lambda,c,cp,cc,ccc,ccccp\} \subseteq V^*$

Grammatica

- $G = \langle V, N, S, P \rangle$ con
- V vocabolario, composto da “simboli terminali”
- N insieme composto da “categorie sintattiche”.
 $N \cap V = \emptyset$
- S simbolo iniziale. $S \in N$
- P è relazione di N su $(N \cup V)^*$, ovvero sottoinsieme di $N \times (N \cup V)^*$.
- $P = \{ \langle n, \eta \rangle \}$, $n \in N$, η sequenza di elementi $\in N \cup V$
 - $\langle n, \eta \rangle$ è chiamata produzione (scritta: $n \rightarrow \eta$)

Grammatica e linguaggio

- Data una grammatica G , si dice

Linguaggio L_G generato da G

l'insieme delle frasi di V

- derivabili dal simbolo iniziale S
- applicando le produzioni P

- Le frasi di un linguaggio di programmazione vengono dette *programmi* di tale linguaggio

Derivazione diretta

- $pre, post \in (N \cup V)^*, a_0 \in N, \alpha_1 \in (N \cup V)^*$
- $pre a_0 post \xrightarrow{l} pre \alpha_1 post$
se
 $a_0 \rightarrow \alpha_1$

Derivazione indiretta

- $\alpha_0, \alpha_N \in (N \cup V)^*$
- $\alpha_0 \rightarrow \alpha_N$
se
 \exists sequenza di $\alpha_i \in (N \cup V)^* \mid \forall i \in [1, N] \alpha_{i-1} \rightarrow^l \alpha_i$

Esercizio

- Creare il linguaggio delle espressioni aritmetiche
 - Deve includere $a + b * (c - d)$
 - Deve escludere $a + b - * c$
- Prima si definiscono V ed S , quindi definendo P si determina anche N
- Creare l'albero sintattico delle due espressioni

EBNF

- $\eta ::= n \quad \eta \rightarrow n \in P$
- $\eta ::= n_1 | n_2 \quad \eta \rightarrow n_1, \eta \rightarrow n_2 \in P$
- $\eta ::= n_{pre}[n]n_{post} \quad n_{pre}n_{post}, n_{pre} n n_{post} \in P$
- $\eta ::= n_{pre} \{n\}_{min}^{max} n_{post}$ P contiene le produzioni che riducono η in una sequenza aperta da n_{pre} e chiusa da n_{post} con n ripetuta da *min* a *max* volte.
- $\langle \rangle$ sono usati per racchiudere $n \in N$

Esercizio

- Scrivere in EBNF il comando copy del DOS
- nel DOS i nomi dei file e directory erano composti da 8 caratteri con eventuale punto e altri 3 caratteri di estensione

Semantica e sintassi

- La definizione della semantica può impiegare le produzioni di una grammatica per ridurre il significato di un'espressione alla composizione del significato delle sue parti.
- Induzione strutturale

Esercizio

- Consideriamo espressioni di logica Booleana
- la semantica di $\langle \text{expr} \rangle$ è il valore Γ che essa restituisce
 - es. $\Gamma(\langle \text{expr} \rangle)$ è il valore True/False dell'espressione
- Valutare l'albero sintattico di $a \wedge (b \vee \neg c)$ per $a = \text{true}, b = c = \text{false}$

Esempio

- XML è definito da una grammatica EBNF di circa 80 regole, tra cui:
 - $\text{document} ::= \text{prolog element Misc}^*$
 - $\text{element} ::= \text{EmptyElemTag} \mid \text{STag content ETag}$
 - $\text{STag} ::= '<' \text{ Name } (\text{S Attribute})^* \text{ S? } '>'$
 - $\text{ETag} ::= '</' \text{ Name } \text{ S? } '>'$
 - $\text{EmptyElemTag} ::= '<' \text{ Name } (\text{S Attribute})^* \text{ S? } '/>'$
 - $\text{Attribute} ::= \text{ Name Eq AttValue}$