# Parsificazione
# I
## (analisi top-down)

# Analisi sintattica I

Data una grammatica G l’analizzatore sintattico o parsificatore legge la stringa sorgente e se appartiene al linguaggio L(G) ne produce una derivazione o un albero sintattico, altrimenti si ferma segnalando l’errore.

## Due classi importanti di analizzatori

Discendenti o top-down

Ascendenti o bottom-up

Come per l’analisi lessicale, anche per l’analisi sintattica sono stati sviluppati strumenti per la generazione automatica di parsificatori, sia per l’analisi bottom-up (Yacc) sia per quella top-down (Antlr).

# Analisi sintattica deterministica

Un analizzatore sintattico *deterministico* leggendo 1 (o più) caratteri in input può *eliminare le ambiguità* e scegliere sempre la strada giusta che porta al riconoscimento della stringa.
Ovviamente il modello su cui i rifanno tutti gli analizzatori sintattici (più o meno fedelmente) è l'automa a pila.

Le grammatiche che permettono **parsing predittivo discendente** sono chiamate **LL(k)**, quelle che permettono **parsing predittivo ascendente** sono chiamate **LR(k)**, dove k è il numero di simboli necessari per individuare la produzione senza ambiguità.

La famiglia **LL(k)** contiene tutti e soli i linguaggi che possono essere definiti da una grammatica LL(k) per un valore finito di k >= 1.
Non tutti i linguaggi che hanno riconoscitori deterministici sono generabili da grammatiche LL(k), cioè la famiglia dei linguaggi LL(k) è strettamente contenuta nella famiglia dei linguaggi che hanno riconoscitori deterministici.

# Parser LL(1) (iterative)

Un parser LL(1) iterativo per una grammatica LL(1) (definita esattamente in seguito) è costituito da:

• la *stringa in input* cui viene aggiunto un mark di fine stringa che denotiamo con **$**. L'input viene letto sequenzialmente come negli automi.

•una *pila* i cui elementi possono essere *terminali* o *non terminali* della grammatica

Durante l'analisi input e pila vengono modificati *esattamente* come l'input e la pila rappresentati nelle configurazioni istantanee di un *automa a pila (non deterministico)* che riconosce la grammatica G costruita nel modo standard.

Ma nell'algoritmo di parsificazione LL(1) la *decisione* su come espandere i nonterminali in cima alla pila viene presa in modo deterministico *guardando il primo simbolo dell'input (lookahead).*
Quando invece in cima alla pila si trova un terminale basterà verificare la corrispondenza di questo col simbolo in input e avanzare la testina di lettura (come nell'automa a pila).

# Parsing LL(1)    (top-down o discendente

Consideriamo per esempio la seguente grammatica (LL(1)).
Notare l'uso di "$" per definire la fine stringa:

Grammatica

1 $S \rightarrow ( S )$
2 $S \rightarrow [ S ]$
3 $S \rightarrow < S >$
4 $S \rightarrow \varepsilon$

Stringa in input: [ < > ] $

| Input | pila | | | |
|---|---|---|---|---|
| [ < > ] $ | | | | S |
| [ < > ] $ | | [ | S | ] |
| < > ] $ | | | S | ] |
| < > ] $ | < | S | > | ] |
| > ] $ | | S | > | ] |
| > ] $ | | | > | ] |
| ] $ | | | | ] |
| $ | | | | |

La decisione su come espandere S viene presa in funzione dei simboli in rosso, primi simboli dell'imput

accettazione

# FIRST

Data una grammatica G = $<V, \Sigma, P, S>$ , l'insieme FIRST di una stringa $\alpha$ di variabili e terminali, è definito formalmente come:

$$FIRST(\alpha) = \{a \mid \alpha \to^* a\beta\} \cup \{\varepsilon \mid \text{se } \alpha \to^* \varepsilon\}$$

E' l'insieme dei terminali con cui iniziano le stringhe derivabili da $\alpha$ nella grammatica G. **FIRST($\alpha$)** (abb. F($\alpha$)) soddisfa questa definizione (ricorsiva):

1. $F(\varepsilon) = \{\varepsilon\}$

$$F(A) = \bigcup_{A \to \gamma_i \in P} F(\gamma_i)$$

2. $F(a\beta) = \{a\}$

3. $F(A\beta) = \begin{cases} F(A) & \text{se A non è annullabile} \\ (F(A) - \{\varepsilon\}) \cup F(\beta) & \text{se A è annullabile} \end{cases}$

N.B. A è annullabile se $A \to^* \varepsilon$

N.B. ($A \to \gamma_1 \mid \gamma_2 \mid \ldots \mid \gamma_k$ è l'insieme delle produzioni di A in G)

# Esempio

$G = < \{X, Y, Z\}, \{a, c, d\}, P, Z >$

P: $Z \rightarrow d \mid XYZ$

$Y \rightarrow c \mid \varepsilon$

$X \rightarrow Y \mid a$

FIRST(d) = {d}

FIRST(XYZ) = {a, c, d}

Infatti $Z \rightarrow XYZ \rightarrow aYZ$

$Z \rightarrow XYZ \rightarrow YZ \rightarrow cZ$

$Z \rightarrow XYZ \rightarrow YZ \rightarrow Z \rightarrow d$

FIRST{X} = {a, c, ε}

Infatti $X \rightarrow a$

$X \rightarrow Y \rightarrow c$

$X \rightarrow Y \rightarrow \varepsilon$

# FOLLOW

Data una grammatica G = <V, Σ, P, S> , l'insieme **FOLLOW** (insieme dei <u>seguiti</u>) di una variabile A è l'insieme dei terminali con cui iniziano le stringhe che seguono A nelle derivazioni della grammatica G (assumendo $ in fine stringa). Formalmente:

$$\mathrm{FOLLOW}(A) = \{a \mid S \to^* \alpha A a \beta\} \cup \{\$ \mid \text{se } S \to^* \alpha A\}$$

Notare che $ appartiene <u>sempre</u> al Fw dell'assioma.

**FOLLOW(A)** (abb. Fw(A)) soddisfa la seguente equazione:

$$Fw(A) = \left[ \bigcup_{B \to \alpha A \beta \in P} (F(\beta) - \{\varepsilon\}) \right] \cup \left[ \bigcup_{\substack{B \to \alpha A \beta \in P \text{ tali che} \\ \beta \text{ annullabile e } B \neq A}} Fw(B) \right] \cup$$

$$\cup \{\$\} \text{ se A è lo start symbol di G}$$

# Esempio

$Z \rightarrow d \mid XYZ$
$Y \rightarrow c \mid \varepsilon$
$X \rightarrow Y \mid a$

FOLLOW(Y) = {a, d, c}

Infatti: $Z \rightarrow XYZ \rightarrow XYXYZ$ -> $XYaYZ$

$Z \rightarrow XYZ \rightarrow XYd$

$Z \rightarrow XYZ \rightarrow YYZ \rightarrow YcZ$

FOLLOW(X) = {c, d, a}

Infatti: $Z \rightarrow XYZ \rightarrow XcZ$

$Z \rightarrow XYZ \rightarrow XYd \rightarrow Xd$

$Z \rightarrow XYZ \rightarrow XYXYZ \rightarrow XXYZ \rightarrow XaYZ$

FOLLOW(Z) = {\$} perchè Z è l'assioma.

# Insiemi guida

Data una grammatica G, **l'insieme guida** di una produzione della grammatica $A \rightarrow \alpha$ - **Gui ($A \rightarrow \alpha$)** - è l'insieme dei terminali (o $\varepsilon$ se ci si trova a fine parola) con cui *iniziano* le stringhe generabili a partire dalla produzione stessa :

**Gui ($A \rightarrow \alpha$)** $= \{a \mid S \rightarrow_{\underline{c}}^{*} wA\beta \rightarrow w\alpha\beta \rightarrow_{\underline{lm}} wa\gamma\}$

L'insieme Gui ($A \rightarrow \alpha$) si può esprimere usando F() e FW():

$$\mathrm{Gui}(A \rightarrow \alpha) = \begin{cases} F(\alpha) & \text{se } \alpha \text{ non è annullabile} \\ (F(\alpha) - \{\varepsilon\}) \cup FW(A) & \text{se } \alpha \text{ è annullabile} \end{cases}$$

**Esempio:**

Gui ($Z \rightarrow d$) = {d}
Gui ($Z \rightarrow XYZ$) = {a, c, d}
Gui ($Y \rightarrow c$) = {c}

Gui ($Y \rightarrow \varepsilon$) = {a, c, d}
Gui ($X \rightarrow Y$) = {a, c, d}
Gui ($X \rightarrow a$) = {a}

# Grammatiche LL(1)

Una **grammatica** è **LL(1)** se *per ogni non terminale* A e *per ogni coppia* di produzioni $A \longrightarrow \alpha$ e $A \longrightarrow \beta$, gli insiemi guida sono disgiunti:

$$\mathbf{Gui(A \longrightarrow \alpha) \cap Gui(A \longrightarrow \beta) = \Phi}$$

**Esempio**:

La grammatica $<\{S\}, \{(, [, <, ), ], >\}, P = \{S \longrightarrow (S) \mid [S] \mid <S> \mid \varepsilon\}, S>$ è LL(1).

$F(S) = \{(, [, <\}$

$FW(S) = \{\$, ), ], >\}$

$Gui(S \longrightarrow (S)) = \{ ( \}$

$Gui(S \longrightarrow [S]) = \{ [ \}$

$Gui(S \longrightarrow <S>) = \{ < \}$

$Gui(S \longrightarrow \varepsilon) = \{\$, ), ], >\}$

# Insiemi GUIDA: esempio di calcolo

$Z \rightarrow d \mid XYZ$

$Y \rightarrow c \mid \varepsilon$

$X \rightarrow Y \mid a$

$F(Z) = \{a, c, d\}$ $\quad F(X) = \{a, c, \varepsilon\}$ $\quad F(Y) = \{c, \varepsilon\}$

$Fw(Z) = \{\$\}$, $\quad Fw(X) = \{a, c, d\}$, $\quad Fw(Y) = \{a, c, d\}$

$\text{Gui}\ (Z \rightarrow d) = F(d) = \{d\}$

$\text{Gui}\ (Z \rightarrow XYZ) = F(XYZ) = (F(X) - \{\varepsilon\}) \cup F(YZ) =$

$= (F(X) - \{\varepsilon\}) \cup (F(Y) - \{\varepsilon\}) \cup F(Z) = \{a, c, d\}$

$\text{Gui}\ (Y \rightarrow c) = \{c\}$

$\text{Gui}\ (Y \rightarrow \varepsilon) = (F(\varepsilon) - \{\varepsilon\}) \cup FW(Y) = \{a, c, d\}$

$\text{Gui}\ (X \rightarrow Y) = \{a, c, d\}$

$\text{Gui}\ (X \rightarrow a) = \{a\}$

**Data la seguente Grammatica:**

| Produzione | Insieme guida |
|---|---|
| 1. $S \rightarrow PQ$ | {a, b, c} |
| 2. $Q \rightarrow \&PQ$ | {&} |
| 3. $Q \rightarrow \varepsilon$ | {$} |
| 4. $P \rightarrow aPb$ | {a} |
| 5. $P \rightarrow bPa$ | {b} |
| 6. $P \rightarrow c$ | {c} |

Verificare gli insiemi guida calcolando I FIRST e i FOLLOW usando direttamente le definizioni.

# Calcolo di FIRST(X) (abbreviato F(X))

Un metodo per calcolare gli F(X) per una grammatica $G = \langle V, \Sigma, P, S \rangle$ dove $X \in V \cup \Sigma$:

1. Si pone $F(a) = \{a\}$ per ogni $a \in \Sigma$.
2. Si pone inizialmente

   $F(A) = \{a \mid A \rightarrow a\,\alpha \in P\} \cup \{\varepsilon \mid \text{se } A \rightarrow \varepsilon \in P\}$.

   Altrimenti si inizializza $F(A) = \{\}$ (insieme vuoto)
3. *per ogni* produzione $A \rightarrow Y_1..Y_k$ :
   1. si *aggiunge* $F(Y_1)-\{\varepsilon\}$ a $F(A)$.
   2. se $\varepsilon \in F(Y_1)$ (tipicamente se $Y_1 \rightarrow \varepsilon \in P$) si *aggiunge* $F(Y_2)-\{\varepsilon\}$ a $F(A)$.
   3. se $\varepsilon \in F(Y_1)$ e $\varepsilon \in F(Y_2)$ si *aggiunge* $F(Y_3)-\{\varepsilon\}$ a $F(A)$.
   4. …….
   5. se $\varepsilon \in F(Y_1),..,F(Y_k)$ si *aggiunge* $\varepsilon$ a $F(A)$.
4. si ripete il passo 3, fino a che gli insiemi F(A) non cambiano più

# Esempio

$G = < \{X, Y, Z\}, \{a, c, d\}, P, Z >$

P: $Z \rightarrow d \mid XYZ$

$Y \rightarrow c \mid \varepsilon$

$X \rightarrow Y \mid a$

Valori iniziali: F(a) = {a}, F(c)= {c}, F{d} = {d}   (non cambiano più)

F(Z) ={ d}, F(Y) = {ε, c}, F(X)={a}

Esaminiamo le produzioni nell'ordine in cui sono scritte:

Dopo una *prima* passata: F(Z) ={d, c,a}, F(Y) = {ε,c}, F(X} = {ε,c,a}

Questi sono i valori *definitivi*.

Per esempio $c \in F(Z)$. Infatti $Z \rightarrow XYZ \rightarrow YYZ \rightarrow YZ \rightarrow cZ$

Esempio 2 : $F(XYZ) = \{c, a\} \cup \{c\} \cup \{d,c,a\} = \{d, c, a\}$

# Calcolo dei Follow(A) (abbreviato Fw(A))

Sia data una grammatica G = <V, Σ, P, S> .Si eseguono i seguenti passi

1. Per ogni $A \in V$ Si calcolano F(A). Si suppone inizialmente Fw(S) = {$} e Fw(A) vuoto per ogni altra $A \in V$.
2. per ogni produzione $A \rightarrow \alpha B \beta \in P$, e per ogni $\beta$ si *aggiunge* $F(\beta) - \{\varepsilon\}$ a Fw (B)
3. *per ogni* produzione $A \rightarrow aB\beta \in P$ tale che $\varepsilon \in F(\beta)$ (ovvero $\beta \in V^*$ è annullabile) si *aggiunge* Fw(A) a Fw(B) ;
4. si ripete il passo 3. fino a che gli insiemi non si cambiano più.

Note: conviene esaminare le produzioni in un ordine fisso.

# Esempio

$Z \rightarrow d \mid XYZ$
$Y \rightarrow c \mid \varepsilon$
$X \rightarrow Y \mid a$

Ricordiamo che F(Z) ={d,c,a}, F(Y) = {ε,c}, F(X} = {ε,c,a}

Valori *iniziali*: Fw(Z)={$}, Fw(Y)={ }, Fw(X)={ }

Dopo il *passo 2.*: Fw(Z)={$}, Fw(Y)={d,c,a}, Fw(X)={d, a, c}.

Non ci sono le condizioni per eseguire il *passo 3.* quindi si salta.

Questi sono i valori definitivi.

Per esempio $a \in Fw(Y)$.
Infatti $Z \rightarrow XYZ \rightarrow XYXYZ \rightarrow XYaYZ$