

I metodi formali dell'Analisi Lessicale: Le grammatiche Lineari Destre

Nicola Fanizzi - Valeria Carofiglio

31 marzo 2016

- 1 Analizzatore Lessicale
 - Funzionalità
- 2 La Gerarchia di Chomsky
- 3 Grammatiche lineari destre

Analisi Lessicale

L'analisi lessicale è l'attività del compilatore tesa ad aggregare i caratteri di un programma sorgente per *riconoscere e classificare*

le stringhe appartenenti al *vocabolario* (o *lessico*) del linguaggio di programmazione

- Un analizzatore lessicale (o ***Scanner***) svolge questa attività
- é spesso realizzato con un apposito sottoprogramma del compilatore
- agisce su chiamata dell'*analizzatore sintattico*
- legge un carattere alla volta finchè non identifica un *simbolo* di una *categoria sintattica* del linguaggio di programmazione

Funzionalità

Funzioni dello scanner:

- Riconoscimento delle categorie
 - parole chiave
 - identificatori
 - numeri
 - separatori ed operatori
 - stringhe
- eliminazione spazi superflui, commenti e caratteri di controllo
- produzione *listati* del sorgente con gli eventuali errori
- espansione delle *macro*

Definizione delle categorie del linguaggio di programmazione (Java-like):

- parole chiave: `class`, `try`, `new`, ...
- identificatori: `somma_parziale`, `h1`, ...
(lunghezza max: 32 caratteri)
- separatori ed operatori a singolo carattere: `,` `+` `-` `;` `=`, ...
- operatori a due caratteri: `[` `]`, `<=` ...
- numeri interi positivi: es. `223472`, ...
- stringhe: es. `"ciao mondo!"`, ...
- commenti: `/* questo e' un commento */`, ...

Spazi e terminatori di linea servono (in genere) a separare i vari simboli del linguaggio

Quali i METODI FORMALI?

Gerarchia delle Grammatiche

Secondo la forma delle regole di produzione si distinguono le seguenti classi di grammatiche:

- Tipo 0** nessuna limitazione alle produzioni: $yAz \rightarrow w$ dove $y, z, w \in (X \cup V)^*$ e $A \in V$;
- Tipo 1** **dipendenti da contesto** (o *context-sensitive*) con produzioni
- $yAz \rightarrow ywz$
con $A \in V$, $y, z \in (X \cup V)^*$ e $w \in (X \cup V)^+$;
 - $S \rightarrow \lambda$
purché S non compaia a destra di alcuna produzione;
- Tipo 2** **libere da contesto** (o *context-free*) con produzioni $v \rightarrow w$, $v \in V$;
- Tipo 3** **lineari (destre)** con produzioni
- $A \rightarrow bC$ con $A, C \in V$, $b \in X$
 - $A \rightarrow d$ con $A \in V$, $d \in X \cup \{\lambda\}$.

I linguaggi generati dalle grammatiche ne mutuano i tipi.

Per descrivere gli elementi del lessico
basta una sottoclasse delle grammatiche libere detta delle
lineari destre

Tipo 3 lineari (destre) con produzioni

- $A \longrightarrow bC$ con $A, C \in V$, $b \in X$
- $A \longrightarrow d$ con $A \in V$, $d \in X \cup \{\lambda\}$.

Esempi

$G_1 = (\{a, b\}, \{S\}, S, P_1)$ con produzioni

$$P_1 = \{S \rightarrow aS \mid a\}$$

é una grammatica lineare destra

$G_2 = (\{a, b\}, \{S, S_1, S_2\}, S, P_2)$ con produzioni

$$P_2 = \{S \rightarrow S_1ab \\ S_1 \rightarrow S_1ab \mid S_2 \\ S_2 \rightarrow a\}$$

NON é una grammatica lineare destra

Esempi

$G_3 = (\{S, A, B\}, \{a, b\}, S, P_3)$ con produzioni

$$P_3 = \{S \rightarrow A \\ A \rightarrow Ab \mid \lambda \\ B \rightarrow aB\}$$

NON é una né lineare destra, né lineare sinistra

Esempio.

Gli identificatori in C-like

$G_1 = (X, V, S, P)$ con

- $X = \{a, b, c, \dots, z, A, B, C, \dots, Z, _, \$, 0, \dots, 9\}$
- $V = \{< lettera >, < cifra >, < simbolo >, < resto >, < identificatore >\}$
- $P = \{< lettera > \rightarrow a | \dots | z | A | \dots | Z | _ | \$$
 $< cifra > \rightarrow 0 | \dots | 9$
 $< simbolo > \rightarrow < lettera > | < cifra >$
 $< resto > \rightarrow < simbolo > | < simbolo > < resto >$
 $< identificatore > \rightarrow < lettera > |$
 $< lettera > < resto > \}$
- $S = < identificatore >$

G_1 non é lineare destra

Esempi.

Gli identificatori in C-like

Cerchiamo la grammatica lineare destra corrispondente a G_1

$$G_2 = (X, V, S, P) \text{ con}$$

- $X = \{a, b, c, \dots, z, A, B, C, \dots, Z, _, \$, 0, \dots, 9\}$
- $V = \{ \langle \textit{identificatore} \rangle, \langle \textit{resto} \rangle \}$
- $P = \{ \langle \textit{Identificatore} \rangle \rightarrow a | \dots | z | A | \dots | Z | _ | \$ |$
 $\quad \quad \quad a < \textit{resto} \rangle |$
 $\quad \quad \quad b < \textit{resto} \rangle |$
 $\quad \quad \quad \dots |$
 $\quad \quad \quad \$ < \textit{resto} \rangle |$
 $\quad \langle \textit{resto} \rangle \rightarrow a | \dots | z | A | \dots | Z | _ | \$ |$
 $\quad \quad \quad 0 | \dots | 9 |$
 $\quad \quad \quad a < \textit{resto} \rangle |$
 $\quad \quad \quad b < \textit{resto} \rangle |$
 $\quad \quad \quad \dots |$
 $\quad \quad \quad \$ < \textit{resto} \rangle |$
 $\quad \quad \quad 0 < \textit{resto} \rangle |$
 $\quad \quad \quad \dots |$
 $\quad \quad \quad 9 < \textit{resto} \rangle$
- $S = \langle \textit{identificatore} \rangle$

G_2 é lineare destra

Esercizi.

- Costruire la grammatica lineare destra per il linguaggio

$$L = \{a^n b^m \mid n \geq 2, m \geq 3\}$$

- Trovare la grammatica lineare destra su $X = \{a, b\}$ consistente di tutte le stringhe con non più di 3 "a".
- Costruire la grammatica lineare destra per il linguaggio

$$L = \{a^n b^m \mid n + m \text{ un numero pari}\}$$

- Costruire la grammatica lineare destra per i linguaggi seguenti su $X = \{a, b\}$

$$L = \{w : n_a(w) - n_b(w) \bmod 3 = 1\}$$

$$L = \{w : n_a(w) - n_b(w) \bmod 3 \neq 1\}$$