

Linguaggi di Programmazione Corso C

Parte n.2

Introduzione ai Linguaggi Formali

Nicola Fanizzi (*fanizzi@di.uniba.it*)

**Dipartimento di Informatica
Università degli Studi di Bari**

Definizioni Preliminari

- Un **alfabeto** è un insieme finito e non vuoto
es. $X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Una **parola** (o stringa) w su un alfabeto X
è una sequenza finita di simboli x_1, x_2, \dots, x_n tale che $\forall i = 1, \dots, n: x_i \in X$.
La lunghezza di w è pari ad n e si denota con $|w|$.
es. $X = \{0, 1\}$ $w = 0010110$.
La parola vuota, denotata con λ , è la parola priva di simboli ($|\lambda| = 0$)
- X^* è l'insieme di tutte le stringhe su X .
es. $X = \{0, 1\} \Rightarrow X^* = \{0, 1, 00, 01, 10, 11, 000, 001, \dots\}$
osservazione: $\forall X: \lambda \in X^*$

Il Monoide (X, \cdot)

- Date $\alpha, \beta \in X^*$ tali che $\alpha = x_1 \cdots x_m$ e $\beta = x'_1 \cdots x'_n$ la **concatenazione** (o **prodotto**) di α e β è data dalla stringa $\alpha\beta$ (denotata anche $\alpha \cdot \beta$) di lunghezza $m + n$ con i primi m simboli uguali a quelli di α e gli ultimi n uguali a quelli di β :

$$\gamma = \alpha\beta = x_1 \cdots x_m x'_1 \cdots x'_n$$

- Il prodotto $\cdot : X^* \times X^* \rightarrow X^*$
 - ha per elemento neutro λ
 - gode della proprietà associativa: $\alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma$
 - non è commutativo

pertanto l'insieme (X, \cdot) è un monoide (non commutativo)

- Ogni parola w su X si può scrivere come prodotto di parole di lunghezza unitaria, si dice pertanto che (X, \cdot) è un monoide libero generato da X

Potenze e Chiusure

- Data la stringa $\delta = \alpha\beta\gamma$, tale che $\alpha, \beta, \gamma \in X^*$, α è un **prefisso** di δ , γ è un **suffisso** di δ e β è una **sottostringa** di δ

es. $\delta = 00110$

$\lambda, 0, 00, 001, 0011$ e δ sono prefissi di δ

$\lambda, 0, 10, 110, 0110$ e δ sono suffissi di δ

$\lambda, 0, 1, 00, 01, 10, 11, 001, 011, 110, 0011, 0110$ e δ sono sottostringhe di δ .

- Data $\alpha \in X^*$, la **potenza k -esima** di α è definita con: $\alpha^k = \begin{cases} \lambda & k=0 \\ \alpha\alpha^{k-1} & k>0 \end{cases}$
- La **potenza di un alfabeto** è definita come segue: $X^k = \begin{cases} \{\lambda\} & k=0 \\ X \cdot X^{k-1} & k>0 \end{cases}$
 - $X^1 = X, X^2 = X \cdot X = \{x_1x_2 | x_1, x_2 \in X\} \dots$ etc.
- L'insieme $X^+ = \bigcup_{k \geq 0} X^k$ dicesi **chiusura transitiva** di X .
- Si osservi che $X^* = X^+ \cup \{\lambda\}$ è la **chiusura riflessiva e transitiva** di X

Linguaggi

- Un **linguaggio** L su un alfabeto X è un sottinsieme di X^* : $L \subseteq X^*$

es. Linguaggio delle parentesi ben formate: $L \subseteq \{(\,)\}^*$:

$((\)) \in L$ e $(\ (\ \)) \in L$ mentre $(\ (\)) \notin L$

I linguaggi possono essere riguardati sotto due punti di vista:

1. **Descrittivo-Generativo**: come generare le parole w di L ?
 L potrebbe essere infinito (estensione) ma enumerabile mediante un numero finito di regole (intensione).
2. **Riconoscitivo**: come decidere se $w \in L$?
E' il punto di vista dei compilatori e traduttori in fase d'analisi

Esempio $X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -\}$

L linguaggio dei numeri relativi

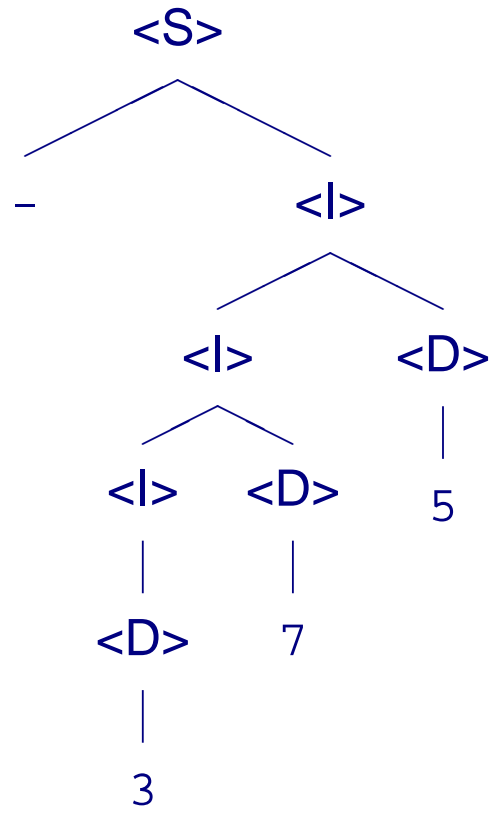
Usando il formalismo di Backus-Naur:

$\langle S \rangle ::= +\langle l \rangle \mid -\langle l \rangle$

$\langle l \rangle ::= \langle D \rangle \mid \langle l \rangle \langle D \rangle$

$\langle D \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$w = -375 \quad \langle S \rangle \Rightarrow -375$



Grammatiche Generative

Una **grammatica generativa** è una quadrupla $G = (X, V, S, P)$ dove:

X alfabeto terminale;

V alfabeto non terminale (NT), tale che $X \cap V = \emptyset$

$S \in V$ simbolo di partenza o distintivo

P insieme delle produzioni (α, β) denotate anche $\alpha \longrightarrow \beta$ dove

$\alpha \in (X \cup V)^+$ contiene almeno un non terminale e

$\beta \in (X \cup V)^*$ (puó essere anche λ)

La notazione $\alpha \longrightarrow \beta_1 | \beta_2 | \dots | \beta_n$ riassume le produzioni:

$$\alpha \longrightarrow \beta_1$$

$$\alpha \longrightarrow \beta_2$$

...

$$\alpha \longrightarrow \beta_n$$

Derivazioni

- Data la grammatica $G = (X, V, S, P)$ e due stringhe y e z su $X \cup V$ tali che $y = \gamma\alpha\delta \in (X \cup V)^+$ e $z = \gamma\beta\delta \in (X \cup V)^*$ con $\alpha, \beta, \gamma, \delta \in (X \cup V)^*$, y **produce direttamente** z (ovvero z è direttamente derivabile da y) sse $a \rightarrow \beta \in P$.
Ciò è denotato con $y \Rightarrow z$.
- y **produce** z (o che z è derivabile da y), denotato con $y \xRightarrow{*} z$, sse
 - $y = z$ oppure
 - $\exists w_1 = y, w_2, \dots, w_{n-1} \in (X \cup V)^+ \text{ e } w_n = z \in (X \cup V)^*$ tali che:
 $w_i \Rightarrow w_{i+1} \quad \forall i = 1, \dots, n - 1$
- Denoteremo con \xRightarrow{n} una derivazione in n passi (lunghezza della derivazione) e con $\xRightarrow{+}$ la chiusura transitiva della relazione \Rightarrow
- Dato un ordinamento su P denoteremo con \Rightarrow_i una derivazione diretta usando la produzione i -esima

Linguaggio Generato da una Grammatica

- Data la grammatica $G = (X, V, S, P)$, il **linguaggio generato dalla grammatica** G , denotato con $L(G)$ è l'insieme delle stringhe di simboli terminali derivabili da S :

$$L(G) = \{w \in X^* \mid S \xRightarrow{*} w\}$$

- $w \in (X \cup V)^*$ è una **forma di frase** di G sse: $S \xRightarrow{*} w$

Alle forme di frase si applicano gli stessi operatori usati fin qui per le stringhe.

- Due **grammatiche** G e G' sono **equivalenti** sse $L(G) = L(G')$

Linguaggio Generato da una Grammatica II

- In generale, dati un linguaggio L ed una grammatica G , non esiste un algoritmo in grado di dimostrare che $L = L(G)$.

Teorema. Il problema generale di dimostrare la correttezza di una grammatica è irresolubile per via algoritmica

- In molti casi specifici, questo si può dimostrare per induzione
 - $L \subseteq L(G)$, i.e. G genera solo stringhe di L
 - $L(G) \subseteq L$, i.e. L contiene solo stringhe generabili da G

Esercizi

1. Determinare la grammatica che genera il linguaggio $L = \{a^n b^n \mid n > 0\}$.
2. Data la grammatica $G = (X, V, S, P)$
con $X = \{0, 1\}$, $V = \{S, A, B\}$ e
 $P = \{S \rightarrow 0B \mid 1A, A \rightarrow 0 \mid 0S \mid 1AA, B \rightarrow 1 \mid 1S \mid 0BB\}$
determinare il linguaggio $L(G)$.
3. Determinare la grammatica che genera il linguaggio $L = \{a^n b^{2n} \mid n > 0\}$.
4. Determinare la grammatica che genera il linguaggio $L = \{a^k b^n c^{2k} \mid n, k > 0\}$.
5. Dimostrare induttivamente che è vuoto il linguaggio $L(G)$ generato dalla grammatica
 $G = (X, V, S, P)$, con $X = \{a, b, c\}$, $V = \{S, A, B\}$
 $P = \{S \rightarrow aBS \mid bA, aB \rightarrow Ac \mid a, bA \rightarrow S \mid Ba\}$

Esercizio Determinare la grammatica che genera $L = \{a^n b^{2n} \mid n > 0\}$.

$$G = (X, V, S, P)$$

$$X = \{a, b\} \quad V = \{S\} \quad P = \{S \xrightarrow{1} aSbb, S \xrightarrow{2} abb\}$$

Dobbiamo dimostrare: $L = L(G)$ cioè: $L \subseteq L(G)$ and $L(G) \subseteq L$

$L(G) \subseteq L$ Sia w una parola derivabile da S in G : $w \in L(G)$ cioè $S \xRightarrow{*} w$

Procediamo per indizione sulla lunghezza della derivazione n di w da S .

base $n = 1$ $S \xRightarrow{*} abb$ e $abb \in L$

passo Dimostriamo $\forall n > 1$: Se $(w' \in L(G) \wedge S \xRightarrow{n-1} w')$ implica $w' \in L$ allora

$(w \in L(G) \wedge S \xRightarrow{n} w)$ implica $w \in L$

Consideriamo: $w \in L(G)$ con $S \xRightarrow{n} w$

e cioè: $\exists w_1, w_2, \dots, w_n : S \Longrightarrow w_1 \Longrightarrow w_2 \Longrightarrow \dots \Longrightarrow w_n = w$

Necessariamente si ha: $w_1 = aSbb \xRightarrow{n-1} w_n = w$.

Per ipotesi di induzione ogni stringa w' derivabile da S in $n - 1$ passi è in L e quindi è del tipo $w' = a^k b^{2k}$, $k > 0$ e precisamente $w' = a^{n-1} b^{2(n-1)}$ essendo $S \xRightarrow{k} a^k S b^{2k}$, $k > 0$.

Quindi $aw'bb = aa^{n-1}b^{2(n-1)}bb = a^n b^{2n} \in L$ è derivabile da S in n passi mediante la derivazione: $S \Longrightarrow_1 aSbb \xRightarrow{n-1} aw'bb = a^n b^{2n} = w$

$L \subseteq L(G)$ Sia $w \in L$. Procediamo per induzione sulla lunghezza della parola w .

base $n = 1$ nella def. di L allora $w = abb$ con $|w| = 3$.

È facile vedere che: $S \implies_2 abb \in L(G)$.

passo Tesi: Se $w' \in L$, $|w'| = n - 1 + 2(n - 1) = 3(n - 1)$ implica $S \xRightarrow{*} w'$
allora $w \in L$, $|w| = 3n$ implica che $S \xRightarrow{*} w$.

Sia w una parola su $X = \{a, b\}$ tale che $w \in L$, $|w| = 3n$, $n > 1$.

L'unica parola di L di tale lunghezza è: $w = a^n b^{2n}$.

Nella derivazione dovremo necessariamente applicare la prima produzione di G come primo passo: $S \implies_1 aSbb$.

Per ipotesi di induzione ogni $w' \in L$, $|w'| = 3(n - 1)$ è derivabile da S in G .

Quindi anche $w' = a^{n-1}b^{2(n-1)}$: $S \xRightarrow{*} a^{n-1}b^{2(n-1)}$.

Unendo i due risultati si ottiene:

$$S \implies_1 aSbb \xRightarrow{*} aw'bb = aa^{n-1}b^{2(n-1)}bb = a^n b^{2n} = w$$

Quindi la tesi è dimostrata.