1. Introduzione ai Linguaggi Formali

(Fanizzi - Carofiglio)

17 marzo 2016

1. Definizioni Preliminari
   - Alfabeti e Stringhe
   - Potenze e Chiusure
   - Linguaggi
2. Grammatiche Generative
   - Definizione
   - Derivazioni
   - Linguaggio Generato da una Grammatica
   - Correttezza di una Grammatica
3. Esercizi

Alfabeti e Stringhe

- Un **alfabeto** è un insieme X finito e non vuoto di simboli

$$\text{es. } X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

- Una **parola** (o ***stringa***) w su un alfabeto X
  è una sequenza finita di simboli $x_1, x_2, \ldots, x_n$ tale che

$$\forall i = 1, \ldots, n \colon x_i \in X$$

La *lunghezza* di w è pari ad n e si denota con $|w|$.

$$\text{es. } X = \{0, 1\} \qquad w = 0010110 \; |w| = 7.$$

La *parola vuota*, denotata con λ, è la parola priva di simboli (quindi $|\lambda| = 0$)

- Si denota con X^* l'insieme di tutte le stringhe su X.

$$\text{es. } X = \{0, 1\} \Rightarrow X^* = \{\lambda, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\}$$

<u>osservazione</u>: $\forall X : \lambda \in X^*$

Concatenzaione o prodotto

- 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:

$$\gamma = \alpha\beta = x_1 \cdots x_m x_1' \cdots x_n'$$

- La concatenzazione su X é una operazione binaria su $\cdot : X^* \times X^* \to X^*$
  - ha per elemento neutro λ
  - gode della proprietà associativa: $\alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma$
  - non è commutativo
- Ogni parola w su X si puó scrivere come prodotto di parole di lunghezza unitaria

Prefisso, suffisso e sottostringa

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. $\delta = 00110$

- prefissi di δ: $\lambda, 0, 00, 001, 0011$ e δ
- suffissi di δ: $\lambda, 0, 10, 110, 0110$ e δ
- sottostringhe di δ:
  $\lambda, 0, 1, 00, 01, 10, 11, 001, 011, 110, 0011, 0110$ e δ

Potenze e Chiusure

- Data $\alpha \in X^*$, la **potenza** k**-esima** di α è definita con:

$$\alpha^k = \begin{cases} \lambda & \text{k=0} \\ \alpha\alpha^{k-1} & \text{k>0} \end{cases}$$

Dunque la potenza k-esima è un caso speciale di concatenamento

Potenze e Chiusure

- La **potenza di un alfabeto** è definita come segue:
  - $X^1 = X$,
  - $X^2 = X \cdot X = \{x_1 x_2 | x_1, x_2 \in X\}$,
  - $X^3 = X \cdot X \cdot X = \{x_1 x_2 x_3 | x_1, x_2, x_3 \in X\}$ ...etc.

$$X^k = \begin{cases} \{\lambda\} & \text{k=0} \\ X \cdot X^{k-1} & \text{k>0} \end{cases}$$

- L'insieme $X^+ = \bigcup_{k>0} X^k$ è **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

$(()() \notin L$

Linguaggi

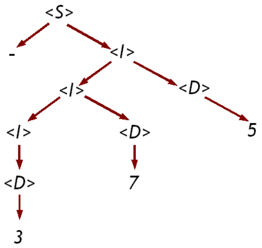
I linguaggi possono essere riguardati sotto due punti di vista:

Descrittivo-Generativo: come generare le parole w di L ?
L potrebbe essere infinito (*estensione*) ma enumerabile mediante un numero finito di regole (*intensione*).

Riconoscitivo: come decidere se $w \in L$?
E' il punto di vista dei compilatori e traduttori in fase d'analisi

Esempio

Esempio. $X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -\}$
L linguaggio dei numeri relativi
Usando il formalismo di Backus-Naur:
<S> ::= +<I> | -<I>
<I> ::= <D> | <I><D>
<D> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
$w = -375$ <S> $\Rightarrow$ -375



Grammatica Generativa

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

- 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 $\alpha \longrightarrow \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n$ riassume le produzioni:

$$\begin{gathered}\alpha \longrightarrow \beta_1 \\ \alpha \longrightarrow \beta_2 \\ \ldots \\ \alpha \longrightarrow \beta_n\end{gathered}$$

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 **deriva direttamente** z sse $a \longrightarrow \beta \in P$. Ció è denotato con $y \Longrightarrow z$
- y **deriva** z, denotato con $y \overset{*}{\Longrightarrow} z$, sse
  - $y = z$ oppure
  - $\exists w_1 = y, w_2, \ldots, w_{n-1} \in (X \cup V)^+$ e $w_n = z \in (X \cup V)^*$ tali che: $w_i \Longrightarrow w_{i+1} \quad \forall i = 1, \ldots, n-1$
- $\overset{n}{\Longrightarrow}$ denota una derivazione in n passi (n *lunghezza della derivazione*)
- Dato un ordinamento su P, $\Longrightarrow_i$ denota 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:

$$L(G) = \{w \in X^* | S \stackrel{*}{\Longrightarrow} w\}$$

- $w \in (X \cup V)^*$ è una **forma di frase** di G sse: $S \stackrel{*}{\Longrightarrow} w$
  Alle forme di frase si applicano gli stessi operatori usati fin qui per le stringhe.
- Due **grammatiche** G e G' sono **equivalenti** sse $L(G) = L(G')$

Esempio

Data la grammatica $G = (X, V, S, P)$,
dove $X = \{a, b\}$ $V = \{S\}$ $P = \{S \to aSb, S \to ab\}$

Determiniamo $L(G)$

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Esempio

- la stringa $ab \in L(G)$ (dato che esiste la produzione $S \rightarrow ab$ $(S \Rightarrow ab)$)
- la stringa $a^2b^2 \in L(G)$ (poichè $S \Rightarrow_1 aSb \Rightarrow_2 aabb = a^2b^2$)
- la stringa $a^3b^3 \in L(G)$ (poichè $S \Rightarrow_1 aSb \Rightarrow_1 aaSbb \Rightarrow_2 aaSbb = a^3b^3$)
- ...

$\{a^nb^n | n > 0\} \subseteq L(G)$
Inoltre tutte le stringhe generate da S in G sono del tipo a^nb^n, ovvero
$L(G) \subseteq \{a^nb^n | n > 0\}$

$$\Downarrow$$

$$L(G) = \{a^nb^n | n > 0\}$$

Correttezza di una grammatica

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 puó 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 $L = \{a^n b^n | n > 0\}$.
2. Determinare la grammatica che genera il linguaggio $L = \{a^n b^{n+1} | n > 0\}$.
3. 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 $L(G)$.
4. Determinare la grammatica che genera il linguaggio $L = \{a^n b^{2n} \mid n > 0\}$.

Esercizi

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

Esercizio 1

Esercizio 1. Determinare la grammatica per $L = \{a^n b^n \mid n > 0\}$

$G = (\{a, b\}, \{S\}, \{S \xrightarrow{1} aSb,\ S \xrightarrow{2} ab\})$
Occorre dimostrare: $L \subseteq L(G)$ and $L(G) \subseteq L$

$L(G) \subseteq L$

Sia $w \in \{a, b\}^*$ tale che: $S \stackrel{*}{\Longrightarrow} w$
Per induzione sulla <u>lunghezza n della derivazione</u> di w da S.

base $n = 1$ $\quad S \stackrel{*}{\Longrightarrow} ab$ e $ab \in L$

passo Dimostriamo che:

$\forall n > 1$
SE $(w' \in L(G) \land S \stackrel{n-1}{\Longrightarrow} w')$ implica $w' \in L$
ALLORA *da* $(w \in L(G) \land S \stackrel{n}{\Longrightarrow} w)$ *consegue* $w \in L$

$L(G) \subseteq L$

- Sia $w \in L(G)$ e $S \xrightarrow{n} w$, cioè:
  $\exists w_1, w_2, \ldots, w_l : S \Longrightarrow w_1 \Longrightarrow w_2 \Longrightarrow \cdots \Longrightarrow w_n = w$
- Necessariamente: $w_1 = aSb \stackrel{n-1}{\Longrightarrow} w_n = w$
- Per ipotesi di induzione:
  É vero che $\forall n \; (w' \in L(G) \wedge S \stackrel{n-1}{\Longrightarrow} w')$ implica $w' \in L$.
  Dunque $S \stackrel{n-1}{\Longrightarrow} w' = a^k b^k$ con $k > 0$.
- ma $S \stackrel{k}{\Longrightarrow} w' = a^k S b^k$ con $k > 0$
  Dunque $w' = a^{n-1} b^{n-1}$
- allora la stringa:
  $aw'b = aa^{n-1}b^{n-1}b{=}a^n b^n$
  é ancora una stringa di L ed é inoltre derivabile da S in n passi
  infatti:$S \stackrel{1}{\Longrightarrow} aSb \stackrel{n-1}{\Longrightarrow} aw'b = a^n b^n = w$

$C.V.D \; L(G) \subseteq L$

$L \subseteq L(G)$

Sia $w \in L$

Per induzione sulla lunghezza $|w|$ della parola $w \in L$

base $|w| = 2$

$\exists\ S \Longrightarrow ab = w$ e $w \in L$

oss: $|W|$ minima è 2 perché devo applicare almeno una regola di derivazione.

passo Dimostriamo che:

$\forall n$

SE $w' \in L, |w'| = 2(n-1)$ implica $S \stackrel{*}{\Longrightarrow} w'$

ALLORA $w \in L, |w| = 2n$ implica $S \stackrel{*}{\Longrightarrow} w$.

$L \subseteq L(G)$

- Sia $w \in L,\ |w| = 2n,\ n > 1$:
  l'unica parola di L di tale lunghezza è $w = a^n b^n$
  Nella derivazione dovremo necessariamente applicare la produzione (1) come primo passo: $S \Longrightarrow_1 aSb$
- Per ipotesi di induzione:
  É vero che $\forall w' \in L,\ |w'| = 2(n-1) : S \stackrel{*}{\Longrightarrow} w'$
  Quindi anche $S \stackrel{*}{\Longrightarrow} a^{n-1}b^{n-1} = w'$
- Unendo i due risultati si ottiene:
  $S \Longrightarrow_1 aSb \stackrel{*}{\Longrightarrow} aw'b = aa^{n-1}b^{n-1}b = a^n b^n = w$

$C.V.D.\ L \subseteq L(G)$

Esercizio 2

Determinare la grammatica per $L = \{a^n b^{n+1} \mid n > 0\}$

$G = (\{a, b\}, \{S, A\}, \{S \xrightarrow{1} Ab, A \xrightarrow{2} aAb,\ A \xrightarrow{3} ab\})$
Occorre dimostrare: $L \subseteq L(G)$ and $L(G) \subseteq L$
... la dimostrazione del tutto uguale alla precedente è lasciata come esercizio.

Esercizio 2, seconda pagina

Dimostrare induttivamente che è vuoto il linguaggio $L(G)$ generato dalla grammatica
$G = (X, V, S, P)$, con $X = \{a, b, c\}$, $V = \{S, A, B\}$
$P = \{S \to aBS \mid bA, aB \to Ac \mid a, bA \to S \mid Ba\}$

Dovremmo provare che:

- $L(G) \subseteq \emptyset$
- $\emptyset \subseteq L(G)$

Ovviamente è sufficiente provare che $L(G) \subseteq \emptyset$.

$L(G) \subseteq \emptyset$

Sia $w \in L(G)$.
se $\forall n \; S \overset{n}{\Longrightarrow} w$ allora $w = \alpha N \beta$, con
$\alpha, \beta \in (X \cup V)^*$ e $N \in V$

Per induzione sulla lunghezza n della derivazione

base $n = 1$
le uniche derivazioni possibili sono:

a. $S \Longrightarrow aBS$
b. $S \Longrightarrow bA$

entrambe presentano almeno un non terminale

$L(G) \subseteq \emptyset$: cenni

passo Dimostriamo che:

$$\forall n > 1$$

SE $S \overset{n-1}{\Longrightarrow} w'$ implica $\exists N \in V : w' = yNz$, con $y,\ z \in (V \cup X)^*$

ALLORA $S \overset{n}{\Longrightarrow} w'$ implica $\exists N \in V : w' = yNz$, con $y,\ z \in (V \cup X)^*$

$L(G) \subseteq \emptyset$: cenni

- Consideriamo una qualunque derivazione in G di n passi: $S \overset{n}{\Longrightarrow} w$
- Per definizione: $\exists w_1, .., w_n = w$ tali che: $S \overset{n-1}{\Longrightarrow} w_{n-1} \Longrightarrow w_n = w$
- Per ipotesi di induzione: ogni stringa derivabile da S in $n-1$ passi presenta un non terminale Dunque anche w_{n-1} presenta un non terminale.
- Si hanno le seguenti possibilità:
  - in w_{n-1} compare il non terminale S: allora....
  - in w_{n-1} compare il non terminale A: allora.....
  - in w_{n-1} compare il non terminale B: allora.....