Grammatiche di Chomsky

Created	@May 12, 2025 11:20 AM
Class	Fondamenti di informatica: linguaggi formali

Ci permettono di rappresentare in maniera finita linguaggi potenzialmente infiniti.

2.1 Grammatiche di Chomsky

In generale, con il termine di *grammatica* si intende un formalismo che permette di definire un insieme di stringhe mediante l'imposizione di un particolare metodo per la loro costruzione. Tale costruzione si effettua partendo da una stringa particolare e *riscrivendo* via via parti di essa secondo una qualche regola specificata nella grammatica stessa.

Definizione 2.1 *Una* grammatica formale $\mathcal{G}$ *è una quadrupla*

$$\mathcal{G} = \langle V_T, V_N, P, S \rangle$$

in cui

1. V_T *è un insieme finito e non vuoto di simboli detto* alfabeto terminale, *i cui elementi sono detti* caratteri terminali, *o* terminali*;*
2. V_N *è un insieme finito e non vuoto di simboli, detto* alfabeto non terminale *i cui elementi sono detti* caratteri non terminali *(o* non terminali, *o* variabili, *o* categorie sintattiche*);*
3. P *è una relazione binaria di cardinalità finita su*

$$(V_T \cup V_N)^* \circ V_N \circ (V_T \cup V_N)^* \times (V_T \cup V_N)^*.$$

   P *è detta insieme delle* regole di produzione *(o* delle produzioni, *o delle* regole sintattiche*). Una coppia $\langle \alpha, \beta \rangle \in P$, si indica generalmente con la notazione $\alpha \longrightarrow \beta$;*

4. $S \in V_N$ *è detto* assioma *ed è il simbolo non terminale di inizio, ossia la categoria sintattica più generale.*

Regole di produzione

stringa → stringa, es: S → 0A1, e quindi quando incontro S, la posso sostituire con 0A1.

Derivazione diretta

Definizione 2.3 *Sia data una grammatica* $\mathcal{G} = \langle V_T, V_N, P, S \rangle$. *La* derivazione diretta *(rispetto a* $\mathcal{G}$*) è una relazione su* $(V^* \circ V_N \circ V^*) \times V^*$, *rappresentata con il simbolo* $\underset{\mathcal{G}}{\Longrightarrow}$ *e così definita: la coppia* $\langle \phi, \psi \rangle$ *appartiene alla relazione, e scriviamo* $\phi \underset{\mathcal{G}}{\Longrightarrow} \psi$ *(*ψ *deriva direttamente da* ϕ *tramite* $\mathcal{G}$*) se esistono* $\alpha \in V^* \circ V_N \circ V^*$ *e* $\beta, \gamma, \delta \in V^*$ *tali che* $\phi = \gamma\alpha\delta$, $\psi = \gamma\beta\delta$ *e* $\alpha \longrightarrow \beta \in P$.

Derivazione non banale (+→)

... da integrare ...

Derivazione *→

Il linguaggio generato da una grammatica formale è dunque un insieme di stringhe di caratteri terminali, ognuna delle quali si può ottenere a partire dall'assioma mediante l'applicazione di un numero finito di passi di derivazione diretta. Equivalentemente, possiamo definire il linguaggio generato da una grammatica come l'insieme di tutte e sole le forme di frase composte da soli simboli terminali.

Quando risulterà chiaro dal contesto a quale grammatica ci riferiamo, anziché $\underset{\mathcal{G}}{\Longrightarrow}$ o $\overset{*}{\underset{\mathcal{G}}{\Longrightarrow}}$ scriveremo semplicemente $\Longrightarrow$ o $\overset{*}{\Longrightarrow}$.

Formalizzazione del linguaggio generato da una grammatica

Definizione 2.6 *Il* linguaggio generato da una grammatica $\mathcal{G}$ *è l'insieme*

$$L(\mathcal{G}) = \left\{ x \mid x \in V_T^* \wedge S \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} x \right\}.$$

Il linguaggio generato da una grammatica formale è dunque un insieme di stringhe di caratteri terminali, ognuna delle quali si può ottenere a partire dall'assioma mediante l'applicazione di un numero finito di passi di derivazione diretta. Equivalentemente, possiamo definire il linguaggio generato da una

2.1.1 *Grammatiche di tipo 0*

Le grammatiche di tipo 0, dette anche *non limitate*, definiscono la classe di linguaggi più ampia possibile.[1] In esse le produzioni sono del tipo più generale:

$$\alpha \longrightarrow \beta, \ \alpha \in V^* \circ V_N \circ V^*, \ \beta \in V^*.$$

Si noti che queste grammatiche ammettono anche derivazioni che "accorciano" le forme di frase , come ad esempio quelle che si ottengono applicando le ε-produzioni.

Esempio 2.6 Le produzioni

$$\begin{array}{rcl} S & \longrightarrow & aSa \mid aAb \mid aAa \mid \varepsilon \\ aAa & \longrightarrow & a \mid \varepsilon \\ aaAb & \longrightarrow & b \mid \varepsilon \end{array}$$

appartengono ad una grammatica di tipo 0. Come conseguenza immediata, la grammatica considerata nell'Esempio 2.2 risulta essere di tipo 0.

Esempio 2.7 La grammatica

$$\mathcal{G} = \langle \{a, b\}, \{S, A\}, P, S \rangle,$$

in cui P è

$$\begin{array}{rcl} S & \longrightarrow & aAb \\ aA & \longrightarrow & aaAb \\ A & \longrightarrow & \varepsilon, \end{array}$$

è anch'essa una grammatica di tipo 0 e genera il linguaggio $L = \{a^n b^n \mid n \geq 1\}$.

I linguaggi generabili da grammatiche di tipo 0 si dicono *linguaggi di tipo 0*.

2.1.2 Grammatiche di tipo 1

Queste grammatiche, dette anche *contestuali* o *context sensitive* (CS), ammettono qualunque regola di produzione che non riduca la lunghezza delle stringhe, cioè produzioni del tipo:

$$\alpha \longrightarrow \gamma, \alpha \in V^* \circ V_N \circ V^*, \gamma \in V^+, |\,\alpha\,| \leq |\,\gamma\,|\ .$$

Esempio 2.8 Le produzioni

$$\begin{array}{rcl} S & \longrightarrow & aSa \mid aAb \mid aAa \\ aA & \longrightarrow & aa \\ Ab & \longrightarrow & aab \end{array}$$

appartengono ad una grammatica di tipo 1.

I linguaggi generabili da grammatiche di tipo 1 si dicono *linguaggi di tipo 1*, o *contestuali*, o *context sensitive* (CS).

Il termine "linguaggio contestuale", deriva dal fatto che, storicamente, questi linguaggi sono stati definiti da Chomsky come la classe dei linguaggi generabili da grammatiche aventi produzioni "contestuali" del tipo

$$\beta_1 A \beta_2 \longrightarrow \beta_1 \gamma \beta_2,\ A \in V_N,\ \beta_1, \beta_2 \in V^*,\ \gamma \in V^+,$$

in cui si esprime il fatto che la produzione $A \longrightarrow \gamma$ può essere applicata solo se A si trova nel contesto $\langle \beta_1, \beta_2 \rangle$. Ad esempio, nella grammatica per il linguaggio $\{a^n b^n c^n \mid n \geq 1\}$ riportata nell'Esempio 2.9, la produzione $Bb \longrightarrow bb$ ha la struttura suddetta, cioè il carattere B può essere rimpiazzato dal carattere b solo se alla sua destra è presente il contesto b.

2.1.3 Grammatiche di tipo 2

Queste grammatiche, dette anche *non contestuali* o *context free* (CF), ammettono produzioni del tipo:

$$A \longrightarrow \beta, \quad A \in V_N, \ \beta \in V^+$$

cioè produzioni in cui ogni non terminale A può essere riscritto in una stringa β indipendentemente dal contesto in cui esso si trova.

Esempio 2.10 Il linguaggio generato dalla grammatica dell'Esempio 2.7 può essere generato anche dalla grammatica equivalente di tipo 2 con produzioni:

$$S \longrightarrow aSb \mid ab.$$

Esempio 2.11 Un esempio di grammatica non contestuale è costituito dalla grammatica che, a partire dall'assioma E, genera espressioni aritmetiche di somme e moltiplicazioni in una variabile i, con le seguenti produzioni:

$$\begin{array}{lcl} E & \longrightarrow & E + T \mid T \\ T & \longrightarrow & T * F \mid F \\ F & \longrightarrow & i \mid (E). \end{array}$$

I linguaggi generabili da grammatiche di tipo 2 vengono detti *linguaggi di tipo 2* o *non contestuali* o *context free* (CF).

2.1.4 Grammatiche di tipo 3

Queste grammatiche, dette anche *lineari destre* o *regolari*, ammettc produzioni del tipo:

$$A \longrightarrow \delta, \quad A \in V_N, \ \delta \in (V_T \circ V_N) \cup V_T.$$

Il termine "regolare" deriva dal fatto che, come vedremo più avanti (di Sezione3.6, tali linguaggi sono rappresentabili per mezzo di espressioni re lari. Il termine "lineare" deriva dal fatto che al lato destro di ogni produzic compare al più un simbolo non terminale.

I linguaggi generabili da grammatiche di tipo 3 vengono detti *linguaggi tipo 3* o *regolari*.

Esempio 2.13 La grammatica definita da

$$\mathcal{G} = \langle \{a, b\}, \{S\}, P, S \rangle$$

in cui P contiene le produzioni

$$\begin{array}{lcl} S & \longrightarrow & aS \\ S & \longrightarrow & b \end{array}$$

è una grammatica di tipo 3 e genera il linguaggio regolare $L = \{a^n b \mid n \geq 0\}$.

In modo del tutto analogo, i linguaggi regolari si possono definire anche mediante grammatiche *lineari sinistre* caratterizzate da regole del tipo:

$$A \longrightarrow \delta, \quad A \in V_N,\, \delta \in (V_N \circ V_T) \cup V_T.$$

Esempio 2.14 Come si è visto nell'Esercizio 2.3, il linguaggio $\{a^n b \mid n \geq 0\}$ può anche essere generato attraverso le produzioni

$$\begin{array}{lcl} S & \longrightarrow & Ab \mid b \\ A & \longrightarrow & Aa \mid a. \end{array}$$

Vedremo nel seguito (Esercizio 3.11) che per ogni grammatica lineare destra ne esiste una lineare sinistra equivalente e viceversa.