2 - Grammatiche e Linguaggi

Alfabeto

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

Esempio:

$$X = 0, 1$$

Con i simboli primitivi dell'alfabeto si formano le **parole**.

Parola (Stringa)

Una sequenza finita di simboli $x_1, x_2 \dots x_n$ dove ogni x_i è preso da uno stesso alfabeto X è una **parola** (su X)
Esempio:
001110 è una parola sull'alfabeto precedente X

Una parola quindi è ottenuta concatenando simboli dell'alfabeto, se una stringa ha m simboli (non necessariamente distinti) ha lunghezza m, valida soltanto se in un certo linguaggio rispetta la grammatica.

Lunghezza di una Parola o Stringa

La lunghezza di una stringa w è denotata con $|w|$
Esempio:

$$|001110| = 6$$

Le parole di lunghezza 1 sono i simboli di X, mentre le parole vuote (o stringhe vuote), denotata con λ, hanno lunghezza 0 e sono prive di simboli

Proprietà

- **Uguaglianze di stringhe**: Due stringhe sono **uguali** se i loro caratteri, letti ordinatamente da sinistra a destra, coincidono
- **X*** (Star): L'insieme di tutte le parole di lunghezza finita sull'alfabeto X si denota con X^*
  - X^* ha un numero di elementi che è un infinito numerabile, dalla definizione segue che $\lambda \in X^*$ per ogni insieme X
    Esempio:
    Se $X = 0, 1$ allora $X^* = \lambda, 0, 1, 00, 01, 10, 11 \dots$
- **Concatenazione o prodotto**: Sia $\alpha \in X^*$ una stringa di lunghezza m e $\beta \in X^*$ una stringa di lunghezza n, la concatenazione di α e β ,denotata con $\alpha\beta$ o $\alpha \cdot \beta$, è definita come la stringa di lunghezza $m + n$, i cui primi m simboli costituiscono una stringa

uguale a α ed i cui ultimi n simboli costituiscono una stringa uguale a β.
Non è nient'altro l'operazione che combina due stringhe (o parole) per formarne una nuova. La concatenazione quindi di $\alpha = x_1, x_2, \dots x_m$(formata da x_m elementi) e $\beta = y_1, y_2 \dots y_n$ (formata da y_n elementi) è nient'altro che la stringa:

$$\alpha\beta = x_1, x_2, \dots x_m, y_1, y_2 \dots y_n$$

con una lunghezza pari a $m + n$
Esempio: Supponendo che X sia l'alfabeto latino, se $\alpha = \text{ciao}$ e $\beta = \text{mondo}$, allora la concatenazione è: ciaomondo

Operazione di concatenazione

La concatenazione di stringhe su X è un operazione binaria su X^* :

$$\cdot : X^* \times X^* \to X^*$$

Dove:

- $X^* \times X^*$ è il prodotto cartesiano di X^* con se stesso, cioè l'insieme di tutte le coppie di stringhe (α, β) con $\alpha, \beta \in X^*$.
- L'operazione $\cdot$ prende due stringhe α e β e restituisce una nuova stringa $\alpha\beta$, ottenuta concatenando α e β.

Proprietà delle operazioni di concatenazione

1. **Associatività**: La concatenazione è un'operazione associativa, l'ordine in cui si concatenano le stringhe non cambia il risultato finale. Questo significa che per tre stringhe α, β, e γ, vale:

   $$(\alpha\beta)\gamma = \alpha(\beta\gamma) = \alpha\beta\gamma$$

   Ad esempio, se $\alpha = a$, $\beta = b$, e $\gamma = c$, allora:

   $$(ab)c = a(bc) = abc$$

2. **Elemento neutro**: La stringa vuota λ è l'elemento neutro per la concatenazione, per qualsiasi stringa $\alpha \in X^*$ vale:

   $$\alpha\lambda = \lambda\alpha = \alpha$$

   Ad esempio, se $\alpha = abc$, allora:

   $$abc\lambda = \lambda abc = abc$$

3. **Non commutatività**: La concatenazione non è commutativa, cioè in generale $\alpha\beta \neq \beta\alpha$. Ad esempio, se $\alpha = ab$ e $\beta = cd$, allora:

   $$\alpha\beta = abcd \quad \text{mentre} \quad \beta\alpha = cdab$$

   e chiaramente $abcd \neq cdab$.

Osservazione

Ogni parola non vuota $\alpha = x_1x_2 \dots x_n$ (dove x_i sono simboli dell'alfabeto X) può essere scritta in **un unico modo** come concatenazione di simboli di lunghezza 1, cioè come prodotto di elementi di X.
Questo significa che una parola α è ottenuta concatenando i simboli $x_1, x_2, \dots, x_n$ in un ordine specifico, e questa rappresentazione è **unica**.

Il fatto che ogni parola non vuota possa essere scritta in modo unico come concatenazione di simboli di X implica che $(X^*, \cdot)$ è il **monoide libero (senza vincoli o relazioni aggiuntive tra i simboli di X, oltre a quelli imposti dalla concatenazione) generato dall'insieme X**.

Definizioni

Prefisso, Suffisso e Sottostringa

Se una parola o stringa $y \in X^*$ è della forma $y = \alpha\beta$ (dove $\alpha, \beta \in X^*$) allora α è un prefisso di y e β un suffisso di y

Una sottostringa è una **sequenza di caratteri consecutivi** estratta da una stringa più grande.
Si definisce con il simbolo δ e viene definita con:

Se $\delta, \beta \in X^*$ e δ è della forma $\delta = \alpha\beta\gamma$ (dove $\alpha, \beta \in X^*$) allora β è una sottostringa di δ

Esempio:
Sia $y = 00110$ allora:
$\{\lambda, 0, 00, 001\}$ è l'insieme dei prefissi di y
$\{\lambda, 0, 10, 110\}$ è l'insieme dei suffissi di y
$\{\lambda, 0, 1, 00, 01, 10, 11, 001, 011, 110, 0011, 0110, 00110\}$ è l'insieme di sottostringhe di y

Potenza di una stringa

La **potenza di una stringa** è un'operazione che consiste nel concatenare una stringa con se stessa un certo numero di volte.
Formalmente, data una stringa α su un alfabeto X, la potenza h-esima di α, denotata con α^h , è definita induttivamente come segue:

$$\alpha^h = \begin{cases} \lambda & \text{se } h = 0 \\ \alpha \cdot \alpha^{h-1} & \text{altrimenti} \end{cases}$$

La potenza h-esima di una stringa quindi non è altro che un caso particolare di **concatenazione ripetuta**.

Esempi
La potenza α^1 è la stringa stessa:

$$\alpha^1 = \alpha$$

La potenza α^2 è la concatenazione di α con se stessa:

$$\alpha^2 = \alpha \cdot \alpha$$

Sia $\alpha = a$ allora:

$$\alpha^0 = \lambda$$

$$\alpha^1 = a$$

$$\alpha^2 = aa$$

$$\alpha^3 = aaa$$

Potenza di un alfabeto

Sia X un alfabeto ,la **potenza i-esima** di X, denotata con X^i, è definita come l'insieme di tutte le stringhe di lunghezza i che possono essere formate concatenando i-esimi simboli di X.

In modo più formale:

- $X^1 = X$ (le stringhe di lunghezza 1 sono semplicemente i simboli dell'alfabeto).
- $X^2 = x_1x_2 \mid x_1, x_2 \in X$ (le stringhe di lunghezza 2 sono tutte le possibili coppie di simboli di X).
- $X^3 = x_1x_2x_3 \mid x_1, x_2, x_3 \in X$ (le stringhe di lunghezza 3 sono tutte le possibili triple di simboli di X).
- In generale, $X^i = x_1x_2 \ldots x_i \mid x_1, x_2, \ldots, x_i \in X$ (le stringhe di lunghezza i sono tutte le possibili sequenze di i simboli di X).

Esempio:
Supponiamo di avere un alfabeto $X = a, b$. Allora:

- $X^1 = a, b$ (stringhe di lunghezza 1).
- $X^2 = aa, ab, ba, bb$ (stringhe di lunghezza 2).
- $X^3 = aaa, aab, aba, abb, baa, bab, bba, bbb$ (stringhe di lunghezza 3).

L'insieme X^+ rappresenta tutte le stringhe di lunghezza **almeno 1** che possono essere formate a partire dall'alfabeto X. Successivamente, si estende questo concetto per includere anche la stringa vuota λ, ottenendo così l'insieme X^*, che rappresenta tutte le stringhe di lunghezza **finita** (inclusa la stringa vuota).

Se $i \geq 1$, l'insieme X^+ è definito come l'unione di tutte le potenze X^i per $i \geq 1$:

$X^+ = X^1 \cup X^2 \cup X^3 \cup \ldots X^i = \bigcup_{i=1}^{+\infty} X^i$

L'insieme X^* è definito come l'unione di X^+ con la stringa vuota λ:

$$X^* = \{\lambda\} \cup X^+$$

X^* contiene tutte le stringhe di lunghezza **finita**, inclusa la stringa vuota λ.

La potenza h-esima di X, X^h, può essere definita in modo induttivo come segue:

$$X^h = \begin{cases} \lambda & \text{se } h = 0 \\ X \cdot X^{h-1} & \text{altrimenti} \end{cases}$$

Linguaggio formale

Un linguaggio formale L su un alfabeto X è un sottoinsieme di X^*

$$L \subseteq X^*$$

Un esempio classico di linguaggio formale è il **linguaggio delle parentesi ben formate**., supponiamo di avere l'alfabeto $X = (,)$,il linguaggio delle parentesi ben formate è un sottoinsieme di X^*, cioè l'insieme di tutte le stringhe di parentesi che sono bilanciate correttamente.

- La stringa $()$ appartiene al linguaggio.
- La stringa $(())$ appartiene al linguaggio.
- La stringa $)($ **non** appartiene al linguaggio.

Si può dire quindi, denotando il linguaggio delle parentesi ben formate con M che:

$$M \subset \{(,)\}^*$$

Un linguaggio di programmazione quindi può essere costruito a partire dall'alfabeto X dei simboli sulla tastiera, quindi **l'insieme, finito o infinito, dei programmi ben costruiti sintatticamente costituisce un linguaggio**

Generazione e riconoscimento dei linguaggi formali

I linguaggi formali possono essere essere visti da due punti:

- Dal punto di vista **Descrittivo/Generativo**:
  Un linguaggio finito può essere **descritto/generato** per elencazione degli elementi (se il numero non è troppo grande). Un linguaggio infinito non è elencabile. Questi sono i più interessanti perché devono essere specificati necessariamente attraverso una proprietà che ne caratterizza gli elementi, che ne definisce l'intensione. Tale proprietà può essere vista come una regola da seguire per generare gli elementi del linguaggio. Il vero problema è trovare la(e) regola(e) generativa(e) (di produzione) di un linguaggio. È quello che accade quando si impara un linguaggio: non è possibile memorizzare tutte le frasi del linguaggio, per esempio, non è possibile elencare tutti i teoremi della teoria dei gruppi, perché i teoremi realizzabili da quelli noti sono infiniti, un esempio con un linguaggio:
  Sia L un linguaggio su $X = \{0\}$, costituito da sole stringhe con numero pari di 0 (

$L = \lambda, 00, 0000, \ldots$), la sua regola di produzione viene espressa con:

$$L = \{\lambda\} \cup \{w^n | w = 00, n = 1, 2 \ldots\}$$

- Dal punto di vista **Riconoscitivo:** Questo secondo punto di vista ha come obiettivo la costruzione di “macchine” in grado di decidere/stabilire se una stringa è un elemento di L oppure no. Si intende costruire una “macchinetta” cui dare in ingresso una particolare parola e che produce una tra due possibili risposte:

$$sì \equiv '\in L' \quad \text{e} \quad no \equiv '\notin L'$$

Analizziamo il problema della generazione di L (quindi dal punto di vista generativo)

Esempio 2.11

Sia dato l'alfabeto:

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

Voglio generare il linguaggio L dei numeri interi relativi. Ovviamente

$$L \subseteq X^*$$

Più precisamente, $L \subset X^*$ poiché, ad esempio, $1++-5 \notin L$.

Non possiamo elencare gli elementi di L. Cerchiamo dunque una serie di regole mediante le quali è possibile produrre tutti e soli gli elementi di L.

Assumiamo, per semplicità, che un numero relativo sia costituito da una serie di cifre precedute da + o −.

Adottiamo la BNF per descrivere le produzioni:

$$< S > ::= + < I > | - < I >$$

$$< I > ::= < D > | < I > < D >$$

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

Queste regole generano tutti gli interi relativi purché partiamo dal simbolo nonterminale S.

I è il simbolo nonterminale (da ora in poi, talvolta abbreviato in NT), anche detto *categoria sintattica*, che sta ad indicare (e da cui si genera) la classe dei numeri interi.

I è definito ricorsivamente o come una cifra oppure come un intero seguito da una cifra.

Ogni intero relativo è generato da queste regole e niente che non sia un intero relativo può essere generato da queste regole.

Generazione ad albero

Proviamo a generare l'intero relativo –375:

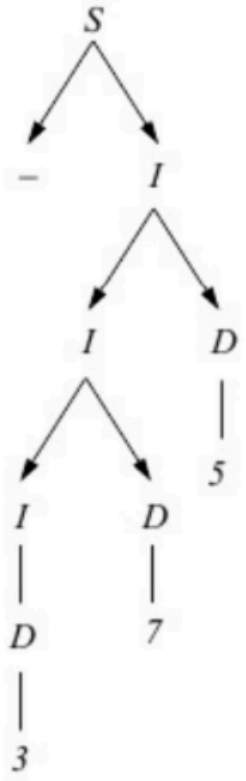

Figura 2.2

Tale albero prende il nome di ***albero di derivazione***.

$$S \overset{*}{\Rightarrow} -375 \quad \Leftrightarrow \quad -375 \in L$$

Grammatiche generative

Per generare un linguaggio quindi sono necessari:

- Un insieme X di simboli primitivi con cui si formano le parole del linguaggio, detto **alfabeto dei simboli terminali**

- Un insieme V di simboli ausiliari (o variabili) con cui si identificano le categorie sintattiche del linguaggio, detto **alfabeto dei simboli non terminali/alfabeto nonterminale/alfabeto delle variabili**
- Un simboli speciale S, scelto tra i nonterminali, da cui far partire la generazione delle parole del linguaggio chiamato **assioma/scopo/simbolo distintivo**
- Un insieme P di **produzioni**, espresse in un formalismo quali regole di scrittura, BNF, carte sintattiche etc..

Definizione di Grammatica

Una grammatica generativa (o struttura di frase G) è una quadrupla

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

dove:

- X è l'alfabeto terminale per la grammatica
- V è l'alfabeto non terminale per la grammatica
- S è il simbolo di partenza per la grammatica
- P è l'insieme di produzioni della grammatica con le seguenti condizioni:$X \cap V = \oslash$ (non hanno elementi comuni tra loro) e $S \in V$ (esiste un simbolo di partenza nell'alfabeto non terminale)

Definizione di produzione

Una produzione è una coppia di parole (v, w),
dove $v \in (X \cup V)^+$ (si può avere tutto quello che si vuole, con vincolo di avere almeno un simbolo non terminale)
e dove $w \in (X \cup V)^*$ (w è l'unione di tutti i simboli terminali e non terminali)
Un elemento (v, w) di P viene comunemente scritto nella forma

$$v \rightarrow w$$

Quindi una produzione deve, in qualche modo, riscrivere un simbolo non terminale NT

Per convenzione, gli elementi di X sono rappresentati di solito con lettere minuscole(con o senza pedici, come prime lettere dell'alfabeto) o cifre ed operatori (connettivi), mentre gli elementi di V sono rappresentati con lettere maiuscole (con o senza pedici) o con stringhe delimitate dalle parentesi angolari <"e">.

Definizione di derivazione o produzione diretta

La **produzione diretta** è il passo più semplice nel processo di derivazione, si tratta di applicare una **regola di produzione** per sostituire un simbolo non terminale con una sequenza di simboli (terminali o non terminali).
Una produzione diretta avviene quando, dove data una grammatica $G = (X, V, S, P)$, abbiamo due stringhe y e z (composte da simboli terminali e non terminali con pezzi in

comune) tali che:

$$y \Rightarrow z$$

Questo significa che z è ottenuta da y applicando una **regola di produzione** della grammatica G.

Esempio:
Consideriamo una grammatica semplice con le seguenti produzioni:

1. $S \to aSb$
2. $S \to ab$
   Supponiamo di voler applicare la prima produzione alla stringa S. La produzione diretta è:
   $S \Rightarrow aSb$
   Abbiamo quindi sostituito S con aSb utilizzando la prima regola di produzione

---

La **derivazione** è un processo più generale che consiste nell'applicare **una o più produzioni dirette** in sequenza per trasformare il simbolo di partenza S in una stringa di simboli terminali.
Una **derivazione** è una sequenza di produzioni dirette:
$y_1 \Rightarrow y_2 \Rightarrow y_3 \cdots \Rightarrow y_n$
dove:

- y_1 è il simbolo di partenza S
- y_n è una stringa di soli simboli terminali (parola del linguaggio)
- ogni passo $y_i \Rightarrow y_{i+1}$ è una produzione diretta

Esempio:
Consideriamo una grammatica semplice con le seguenti produzioni:

1. $S \to aSb$
2. $S \to ab$

Per derivare una stringa come $aabb$ eseguiamo i seguenti passaggi:

1. Partiamo da S
2. Applichiamo la prima regola di produzione
   $S \Rightarrow aSb$
3. Applichiamola di nuovo
   $aSb \Rightarrow aaSbb$
4. Applichiamo la seconda regola di produzione
   $aaSbb \Rightarrow aabb$

La derivazione completa quindi è

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$$

Definizione di linguaggio generato da una grammatica

Sia $G = (X, V, S, P)$ una grammatica, il **linguaggio generato da G**, denotato con $L(G)$, è l'insieme delle stringhe di terminali derivabili dal simbolo di partenza S

$$L(G) = (w \in X^* | S \Rightarrow w)$$

Sono stringhe di $L(G)$ le stringhe che:

- Consistono di solo terminali
- Possono essere derivate da S in G

Definizione di forma di frase

Sia $G = (X, V, S, P)$ una grammatica, una stringa w dove $w \in (X \cup V)^*$ è una forma di frase di G se:

$$S \Rightarrow w$$

Alle forme di frase si applicano le stesse definizioni (come la potenza) e gli stessi operatori (come la concatenazione) dati per le stringhe

Definizione di grammatiche equivalenti

Due grammatiche G e G' si dicono **equivalenti** se generano lo stesso linguaggio, ossia se

$$L(G) = L(G')$$

Esempio

- Sia $G = (X, V, S, P)$, ove

$$X = \{a,b\}, \qquad V = \{S\}, \qquad P = \left\{ S \overset{(1)}{\rightarrow} aSb, S \overset{(2)}{\rightarrow} ab \right\}$$

Determiniamo $L(G)$.

$ab \in L(G)$ poiché $S \underset{(2)}{\Rightarrow} ab$

Se numeriamo le produzioni, possiamo indicare la produzione usata immediatamente al di sotto del simbolo $\Rightarrow$.

$\underset{(n)}{\Rightarrow}$ ≡ ho applicato la produzione n

$y \overset{k}{\Rightarrow} z$ ≡ y produce z in k passi, dove k=lunghezza della derivazione

Esempio

- $a^2b^2 \in L(G)$ poiché $S \underset{(1)}{\Rightarrow} aSb \underset{(2)}{\Rightarrow} a^2b^2$
- $a^3b^3 \in L(G)$ poiché $S \overset{3}{\Rightarrow} a^3b^3$

.....

$$\{a^n b^n \mid n > 0\} \subseteq L(G)$$

- Inoltre, qualsiasi derivazione da S in G produce frasi del tipo a^nb^n.
  - Dunque $L(G) \subseteq \{a^n b^n \mid n > 0\}$ e quindi

$$L(G) = \{a^n b^n \mid n > 0\}$$

Osservazione

Dunque, una grammatica è uno strumento generativo di un linguaggio perché, data una qualsiasi parola di quel linguaggio, possiamo risalire mediante le produzioni al simbolo di partenza della grammatica.

Viceversa, dato il simbolo di partenza di una grammatica, seguendo uno qualsiasi dei cammini dell'albero di derivazione, si produce una parola "valida" del linguaggio.

In generale, dato un linguaggio L ed una grammatica G, non esiste un algoritmo in grado di dimostrare che la grammatica genera il linguaggio, ossia che $L = L(G)$.
Più specificamente, non esiste un algoritmo che stabilisce se una data stringa è generata o no dalla grammatica presa in considerazione.

In molti casi importanti, però, è possibile dimostrare per induzione che una particolare grammatica genera proprio un particolare linguaggio. Queste dimostrazioni ci consentono di stabilire se, data una grammatica G ed un linguaggio L, risulta:

- $w \in L(G) \Rightarrow w \in L$ cioè $L(G) \subseteq L$ (La grammatica G genera solo stringhe appartenenti al linguaggio L)
- $w \in L \Rightarrow w \in L(G)$ cioè $L \subseteq L(G)$ (Il linguaggio L comprende solo parole generabili dalla grammatica G)

Principio di induzione

Sia n_0 un intero e sia $P = P(n)$ un enunciato che ha senso per ogni intero maggiore o uguale a n_0, se:

- $P(n_0)$ è vero
- Per ogni $n > n_0$, $P(n-1)$ vero implica $P(n)$ vero
  allora $P(n)$ è vero per tutti gli n maggiori o uguali ad n_0

L'**induzione** quindi è un procedimento che cerca di stabilire una legge universale partendo da singoli casi particolari