4 - Linguaggi liberi da contesto

(Forniti da Pice)
Alberi di derivazione $\rightarrow$ rappresentazione delle derivazioni in una grammatica libera da contesto.
La sequenza di regole usate per produrre una stringa w è chiamata *struttura di w*. (Si indica con $S \overset{*}{\Rightarrow} w$).

Albero $\rightarrow$ grafo orientato, aciclico, connesso e avente al massimo un arco entrante in ciascun nodo.
L'insieme dei nodi, presi da sinistra verso destra, è chiamata frontiera.

Data una grammatica C.F. e una parola w derivabile da tale linguaggio, un albero T rispetta le seguenti proprietà:

- la radice è etichettata con S
- ogni nodo interno è etichettato con un simbolo di V
- ogni nodo esterno (foglie) è etichettato con un simbolo di X o λ
- se un nodo N è etichettato con A, ed N ha k discendenti diretti $N_1, N_2, \ldots, N_k$ etichettati con i simboli $A_1, A_2, \ldots, A_k$, allora la produzione $A \rightarrow A_1, A_2, \ldots, A_k$ appartiene a P.
- la stringa w è rappresentata dalla frontiera dell'albero.

Lunghezza di un cammino $\rightarrow$ numero di nonterminali dalla radice ad una foglia.

Altezza o Profondità $\rightarrow$ lunghezza del cammino più lungo di un albero.

Un albero di derivazione non impone alcun ordine sull'applicazione delle produzioni in una derivazione. In altri termini, data una derivazione, esiste uno ed un solo albero di derivazione che la rappresenta, mentre un albero di derivazione rappresenta in generale più derivazioni (in base all'ordine col quale si espandono i nonterminali).

Derivazione destra

Data una grammatica G, una derivazione destra (o sinistra), è una derivazione che va a riscrivere il simbolo nonterminale più a destra (o più a sinistra).

Principio di sostituzione di sottoalberi

Data una grammatica, le cui produzioni sono:
$S \rightarrow 0B|1A$
$A \rightarrow 0|0S|1AA$

$B \to 1|1S|0BB$

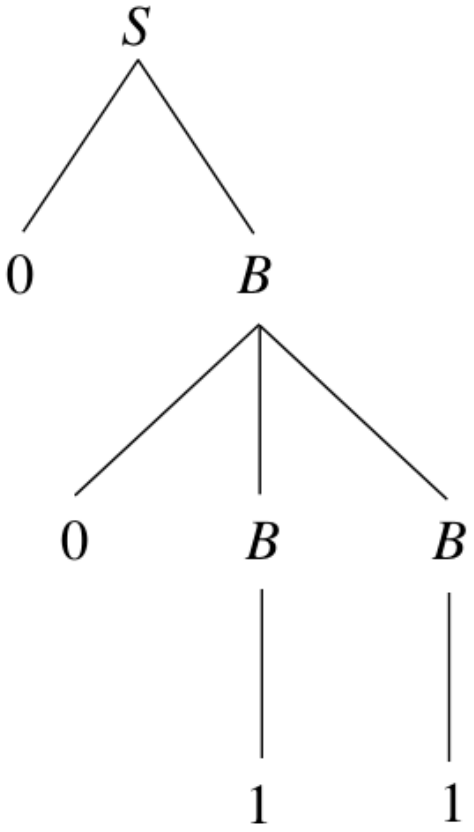
Considerando la stringa 0011, è possibile derivarla in modi diversi:

$S \Rightarrow 0B \Rightarrow 00BB \Rightarrow 001B \Rightarrow 0011$

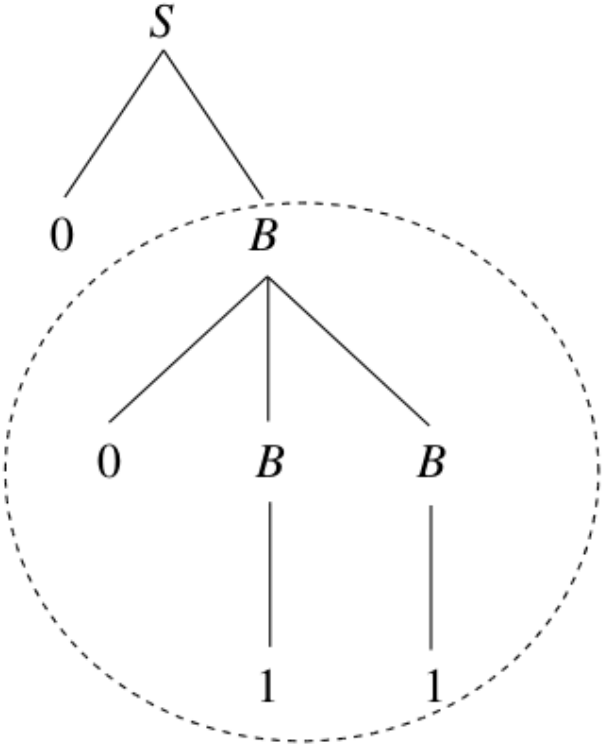
oppure

$S \Rightarrow 0B \Rightarrow 00BB \Rightarrow 00B1 \Rightarrow 0011$

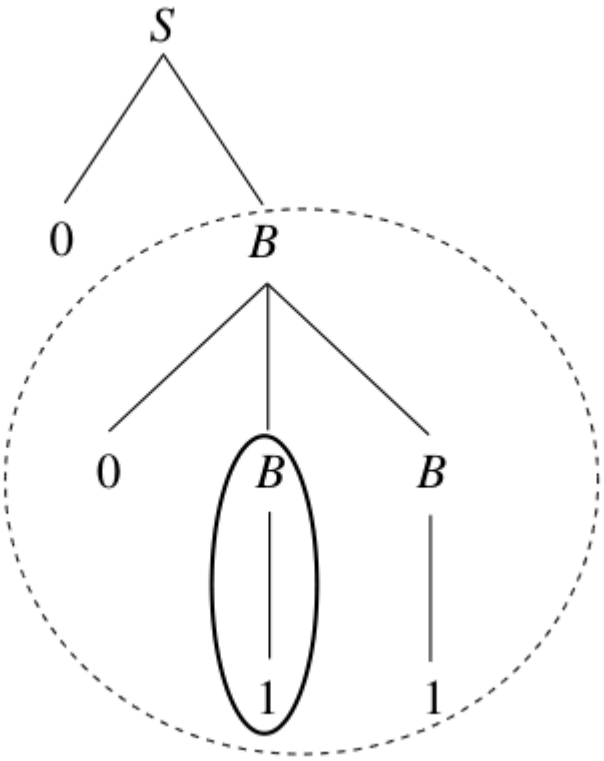
Tale diversità scompare quando se ne ricava l'albero di derivazione. Di fatti, per entrambe le derivazione l'albero rimane il seguente:



Consideriamo, di tale albero, il seguente sottoalbero:



Cosa accade se un sostituiamo un sottoalbero di radice B con il sottoalbero considerato? Il nuovo albero è ancora un albero di derivazione?
Otterremmo un albero del genere:



Il nuovo albero è anch'esso un albero di derivazione, per una parola ovviamente diversa. (In questo caso la parola sarebbe 000111).

Questo processo di sostituzione potrebbe essere effettuato all'infinito, ottenendo parole che crescono in maniera costante:

$$0011 \implies |w| = 4$$

$$000111 \implies |w| = 6$$

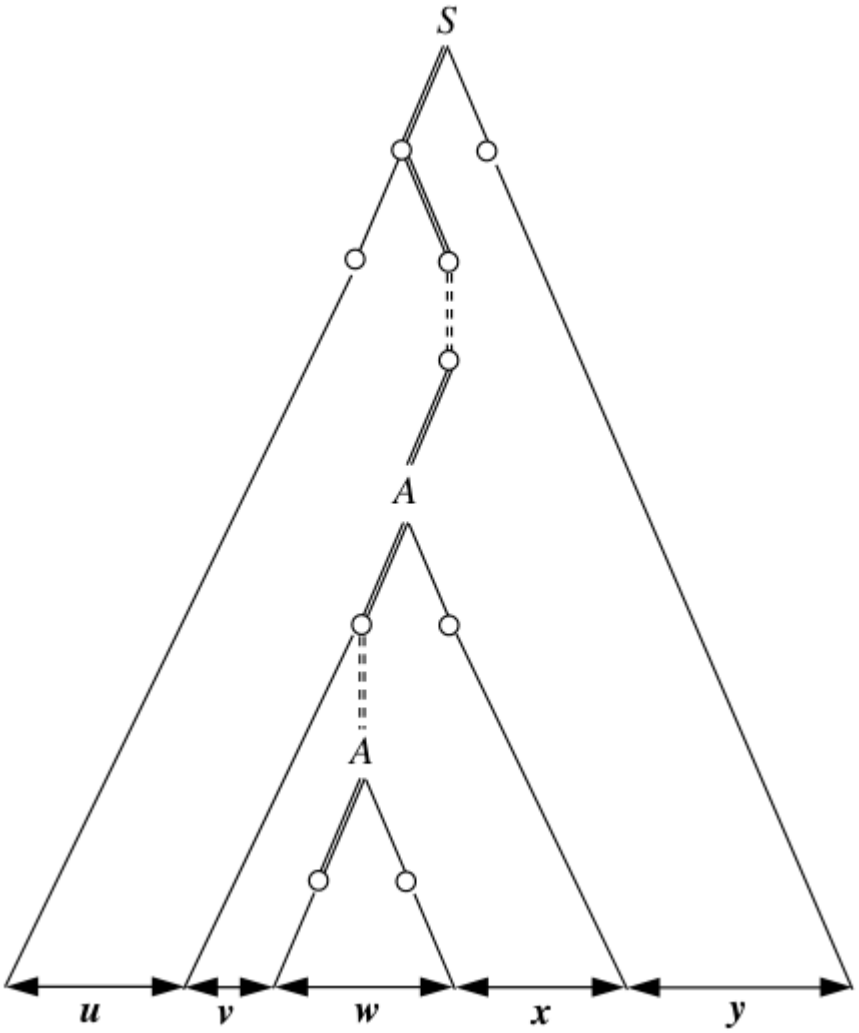
$$00001111 \implies |w| = 8$$

$$00^n11^n \implies |w| = 2n + 2 \text{ ove } n = 1, 2, K$$

ATTENZIONE: L libero $\implies$ crescita delle parole costanti. Ovvero, se il linguaggio è libero, allora la crescita delle parole è costante, il contrario non è per forza vero.
Tale crescita costante è una proprietà dei linguaggi liberi in generale. Dunque, se una grammatica genera parole la cui lunghezza cresce in maniera esponenziale, allora il linguaggio generato non è libero.

Formalizzazione:

Supponiamo di avere un albero di derivazione T_Z per una stringa z di terminali generata da una grammatica C.F. G, e supponiamo inoltre che il simbolo NTA compaia due volte su uno stesso cammino.



Il sottoalbero più in basso con radice nel nodo etichettato con una A genera la sottostringa w , mentre quello più in alto genera la sottostringa vwx. Poiché la G è C.F., sostituendo il sottoalbero più in alto con quello più in basso, si ottiene ancora una derivazione valida. Il nuovo albero genera la stringa uwy. (In pratica, siccome il processo di sostituzione lo si può effettuare fin quando abbiamo due nodi con uguale etichetta ed il linguaggio ovviamente C.F., prendendo il sottoalbero di radice A che genera w, sostituendolo al sottoalbero sempre di radice A ma che produce vwx avremmo un albero la cui frontiera diverrebbe uwy)

Se effettuiamo la sostituzione inversa (il sottoalbero più in basso viene rimpiazzato da quello più in alto), otteniamo un albero di derivazione lecito per la stringa $uvvwxxy$, ossia uv^2wx^2y. Ripetendo questa sostituzione un numero finito di volte, si ottiene l'insieme di stringhe:
$\{uv^nwx^ny|n \geq 0\}$
Dunque:
Ogni linguaggio C.F. infinito deve contenere almeno un sottoinsieme infinito di stringhe della forma:
$\{uv^nwx^ny|n \geq 0\}$

Data una grammatica G, context free, supponiamo che:
$m = max\{|v| \ t.c. \ A \rightarrow v \in P\}$

Dato l'albero T_W un albero di derivazione per una stringa w di L(G). Se l'altezza di T_W è al più uguale ad un intero j, allora: $|w| \leq m^j$

Quindi:
$height(T_W) \leq j \implies |w| \leq m^j$
Dimostrazione:
La dimostrazione la si fa tramite il principio di induzione:

- Passo Base:
  $j = 1$ quindi T_w rappresenta una unica produzione e quindi $|w| \leq m$ poiche' $1 \leq m$ sempre
- Passo induttivo:
  Supponiamo che il lemma valga per ogni albero di altezza al più uguale a j e la cui radice sia un simbolo NT e dimostriamolo per un albero di altezza j+1.
  Supponiamo che il livello più alto dell'albero rappresenti la produzione $A \to v$, dove $v = v_1 v_2 \dots v_k, |v| = k, k \leq m$ (il sottoalbero di profondità 1 ha al più m figli).
  Ogni simbolo $v_i, i = 1, 2, \dots, k$ di v può essere radice di un sottoalbero di altezza al più uguale a j, poiché T_w ha altezza uguale a $j + 1$ (un v_i potrebbe anche essere un terminale). Dunque, per ipotesi di induzione, ciascuno di questi alberi ha al più m^j foglie.
  Poiché $|v| = k, k \leq m$, la stringa w, frontiera dell'albero T_w , ha lunghezza:

$$|w| \leq \underbrace{m^j + m^j + \dots + m^j}_{|\nu| = k \text{ volte}} = |\nu| \cdot m^j = k \cdot m^j \leq m \cdot m^j = m^{j+1}$$

Pumping Lemma per i linguaggi liberi da contesto (o teorema uvwxy)

Sia L un linguaggio libero da contesto.
Allora esiste una costante p, che dipende solo da L, tale che se z è una parola di L di lunghezza maggiore di p($|z| > p$), allora z può essere scritta come $uvwxy$ in modo tale che:
1. $|vwx| \leq p$
2. al più uno tra v e x (quindi o v o x) è la parola vuota ($vx \neq \lambda$)
3. $\forall i,\ i \geq 0$: $uv^iwx^iy \in L$
4. 1. **Attenzione**: v e x non devono essere entrambe nulle, ma ciò non vale per le potenze. In altre parole, nonostante $vx = \lambda$ non sia accettato, uv^iwx^iy con $i = 0$ è accettato (seppur $v^0x^0 = \lambda$).

Possiamo usare questi assiomi per dimostrare che un linguaggio non è libero.
Di fatto, se almeno una parola del linguaggio non rispecchia una delle proprietà, allora tale linguaggio non è C.F..

Di solito si usa la terza proprietà: prendendo una stringa pompata, si dimostra che tale stringa non appartiene al linguaggio.
Quindi:
Se un linguaggio infinito non obbedisce al Pumping Lemma, non può essere generato da una grammatica C.F.

```
GUARDA ESERCIZI SU PUMPING LEMMA
```

Grammatiche Ambigue

Definizione

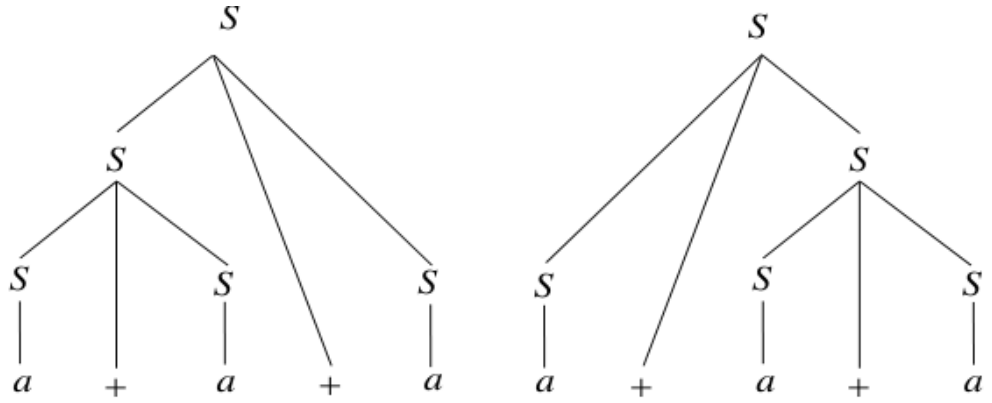
Una grammatica G libera da contesto è ambigua se esiste almeno una stringa x in $L(G)$ che ha due alberi di derivazione differenti (in alternativa, che ha due derivazioni destre o sinistre distinte)

Prendiamo in esempio una grammatica libera da contesto con queste regole:

$$G_2 = (X, V, S, P)$$

$$X = \{a, +\}, \qquad V = \{S\}, \qquad P = \{S \to S + S,\ S \to a\}$$

Questa grammatica è ambigua, la stringa $w = a + a + a$ in $L(G_2)$ ha due alberi differenti di derivazione



L'albero di derivazione è differente quando la sua struttura è differente (ci sono casi dove potrebbe essere uguale pur essendo ambigua), quindi se esiste più di un albero di derivazione per una stessa frase, essa può avere un significato non univoco.
L'ambiguità è una proprietà negativa nei linguaggi di programmazione, infatti il significato di una frase può essere definito come una funzione del suo albero di derivazione.
L'unico vantaggio delle grammatiche ambigue risulta essere, in generale, il minore numero di regole che possono avere rispetto ad una grammatica non ambigua.

Linguaggi inerentemente ambigui

Esistono però dei linguaggi, detti inerentemente ambigui, per i quali tutte le grammatiche che li generano risultano ambigue

Definizione

Un linguaggio L è inerentemente ambiguo se ogni grammatica che lo genera è ambigua
Definito anche con:

$$(\forall G) L = LG : G \text{ ambigua}$$

Un linguaggio inerentemente ambiguo è:

$$L = \{a^i b^j c^k | a^i b^j c^k, (i = j \quad \nu \quad j = k)\}$$

Intuitivamente, ci si rende conto di ciò pensando che in ogni grammatica che genera L devono esistere due diversi insiemi di regole per produrre le stringhe $a^i b^i c^k$ e le stringhe $a^i b^i c^i$. Le stringhe $a^i b^i c^i$ saranno necessariamente prodotte in due modi distinti, con distinti alberi di derivazione e conseguente ambiguità.

- Linguaggi di programmazione
  - Si consideri la seguente grammatica $G = (X, V, S, P)$:

$X = \{\underline{\text{if}}, \underline{\text{then}}, \underline{\text{else}}, \text{a}, \text{b}, \text{p}, \text{q}\}$

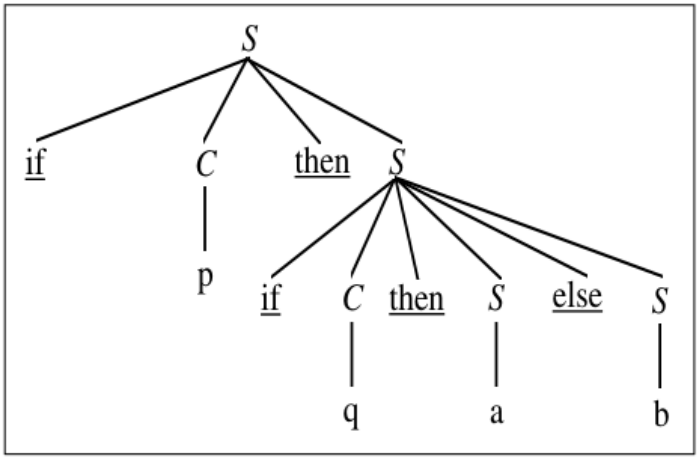
$V = \{S, C\}$

$P = \{S \rightarrow \underline{\text{if}}\ C\ \underline{\text{then}}\ S\ \underline{\text{else}}\ S \mid \underline{\text{if}}\ C\ \underline{\text{then}}\ S \mid \text{a} \mid \text{b},\ C \rightarrow \text{p} \mid \text{q}\}$

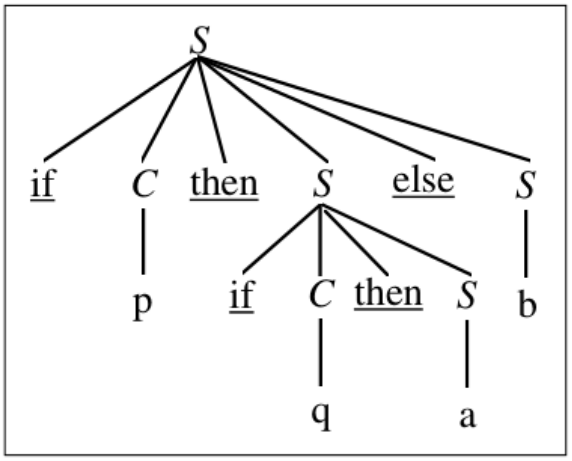
G è ambigua. Infatti la stringa:

$w =$ if p then if q then a else b

può essere generata in due modi.



if p then (if q then a else b)



if p then (if q then a) else b

Per rendere non ambigua G si usa solitamente la convenzione di associare ogni istruzione else con l'istruzione if più vicina.
La grammatica che riflette questa convenzione è la seguente:

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

$$X = \{\underline{\text{if}}, \underline{\text{then}}, \underline{\text{else}}, \text{a}, \text{b}, \text{p}, \text{q}\}$$

$$V = \{S, S1, S2, C, T\}$$

$$P = \{S \to S1 \mid S2, S1 \to \underline{\text{if}}\ C\ \underline{\text{then}}\ S1\ \underline{\text{else}}\ S2 \mid T, S2 \to \underline{\text{if}}\ C\ \underline{\text{then}}\ S \mid \underline{\text{if}}\ C\ \underline{\text{then}}\ S1\ \underline{\text{else}}\ S2 \mid T, C \to p \mid q, T \to a \mid b\}$$

In G' la stringa $w = \text{if p then if q then a else b}$ ha un unico albero di derivazione: