Nella precedente lezione abbiamo incontrato le grammatiche, dandone una definizione formale, e la nozione di derivazione, in uno e più passi.
Le grammatiche che abbiamo visto consistevano di più produzioni con lo stesso driver:

$$G: \begin{array}{l} S \to aSb \\ S \to ab \end{array} \qquad \text{Possiamo condensare in: } S \to aSb \mid ab$$

La derivazione in più passi era di questo tipo:

$$G: \begin{array}{l} S \to aAb \\ aA \to aaAb \\ A \to \varepsilon \end{array}$$

$$S \overset{①}{\Rightarrow} aAb \overset{③}{\Rightarrow} ab$$
$$aAb \overset{②}{\Rightarrow} aaAbb \overset{③}{\Rightarrow} aabb$$
$$aaAbb \overset{②}{\Rightarrow} aaaAbbb \overset{③}{\Rightarrow} aaabbb$$
$$aaaAbbb \overset{②}{\Rightarrow} \dots$$

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

Oggi proviamo esercizi del tipo “dato il linguaggio, determina la grammatica”.

$$L_1 = \{a^n b^j \mid n \geq 0, j > 0\}$$

Tentativo 1: errato

$$G_1: \begin{cases} S \to aSb \\ S \to b \end{cases}$$

Abbiamo: $S \overset{①}{\Rightarrow} a^k S b^k, k \geq 0 \overset{②}{\Rightarrow} a^k b b^k$, e cioè a G_1 corrisponde $\{a^n b^{n+1} \mid n \geq 0\}$

Tentativo 2: errato

$$G_2: \begin{cases} S \to aS \\ S \to b \end{cases}$$

No, perché produce $L(G_2) = \{a^n b \mid n \geq 0\}$ e $a^2b^2 \in L_1$ & $a^2b^2 \notin L(G_2)$

Tentativo 3: errato

$$G_3: \begin{cases} S \to ab \mid bB \\ A \to a \mid \varepsilon \\ B \to b \mid \varepsilon \end{cases}$$

No, non c'è ricorsione sulla A

La mia proposta: CORRETTA!

$$S \to A \mid aS$$
$$A \to bA \mid b$$

Tentativo 4:

$$G_4: \begin{cases} S \to AB \\ A \to aA \mid a \mid \varepsilon \\ B \to bB \mid b \end{cases}$$

superflua

È corretto.

Abbiamo poi studiato i vari modi di ottenere ‘a’ da $A \to aA \mid a \mid \varepsilon$

1 $A \Rightarrow aA \Rightarrow a$

2 $A \Rightarrow a$

È un esempio di grammatica ambigua: esiste almeno una parola appartenente al linguaggio per la quale si possono proporre due derivazioni “canoniche”. Altamente negative, perché non si analizzano (non si riesce).
L’analisi sintattica si pone come principale obiettivo di capire se il programma è derivabile dal linguaggio in cui lo ho scritto. Esistono varie tecniche per derivare le frasi, con l’albero, ad esempio; ma si vuole che questo albero per derivarle sia unico.

Facciamo un esempio: 1 + 2 x 3 = 7… ma perché non 9? Perché ci hanno insegnato le regole di precedenza.
Proviamo a descrivere tale frase con la seguente grammatica:

$$S \to S + S \mid S \cdot S \mid n$$

$$S \overset{?}{\Rightarrow}^* num + num \cdot num?$$

Scegliamo la convenzione rightmost

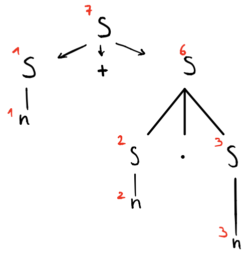
Individuiamo 2 derivazioni:

D1. $S \overset{1}{\Rightarrow} S+S \overset{3}{\Rightarrow} num+S \overset{2}{\Rightarrow} num + S \cdot S \overset{3}{\Rightarrow} num + num \cdot S \overset{3}{\Rightarrow} num + num \cdot num$

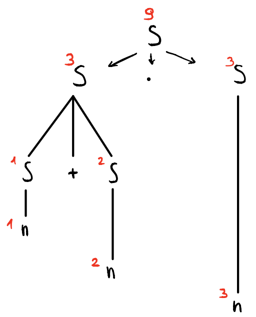
D2. $S \overset{2}{\Rightarrow} S \cdot S \overset{1}{\Rightarrow} S+S \cdot S \overset{3}{\Rightarrow} num+S \cdot S \overset{3}{\Rightarrow} num + num \cdot S \overset{3}{\Rightarrow} num + num \cdot num$

Esaminiamone gli alberi di derivazione:

D1



S
$S+S$
$n+S$
$n+S \cdot S$
$n+n \cdot S$
$n+n \cdot n$



S
$S \cdot S$
$S+S \cdot S$
$n+S \cdot S$
$n+n \cdot S$
$n+n \cdot n$

Eravamo partiti da 1 + 2 x 3. L'analisi lessicale aveva tradotto tale stringa in <n,1> + <n,2> * <n,3>.

Abbiamo visto che è una grammatica ambigua; l'ambiguità non ci piace.

Definizioni:
- Una derivazione si dice **leftmost** (**rightmost**) se ad ogni passo viene sostituito il non-terminale più a sinistra (destra) della stringa corrente.
- Una grammatica G è ambigua ⇔ ∃ w ∈ L(G) esistono 2 distinte derivazioni, entrambe leftmost oppure entrambe rightmost.

$S \rightarrow$ if b then S | if b then S else S | a

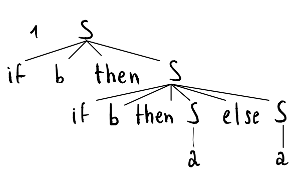
E' ambigua? Fornite un esempio.

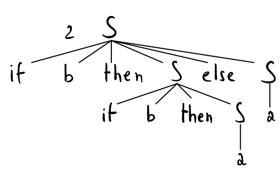
FGAZ: if b then if b then a else a

1 - if b then (if b then a else a)

2 - if b then (if b then a) else a

Alberi:





Riflettiamo su $L: \{a^n b^n c^n \mid n > 0\}$

$S \to aSBc \mid abc$

$cB \to Bc$

$bB \to bb$

Proviamo: $S \overset{1}{\Rightarrow} aSBc \overset{1}{\Rightarrow} aaSBcBc \overset{3}{\Rightarrow} aaSBBcc$