

# Ambiguità delle grammatiche acontestuali

Il concetto di ambiguità è stato introdotto indipendentemente nel 1962, da David G. Cantor, e da Robert W. Floyd entrambi nel loro articolo mettono in rilievo un'ambiguità dell'ALGOL60.

L'esordio dell'articolo di D.G. Cantor recita:

[“Backus \[1\] has developed an elegant method of defining well-formed formulas for computer languages such as ALGOL”](#)

# AMBIGUITÁ - Es. 1

$E \rightarrow E + E$

$E \rightarrow E * E$

$E \rightarrow (E)$

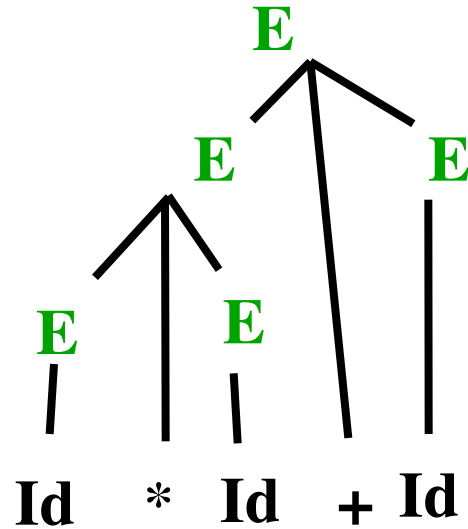
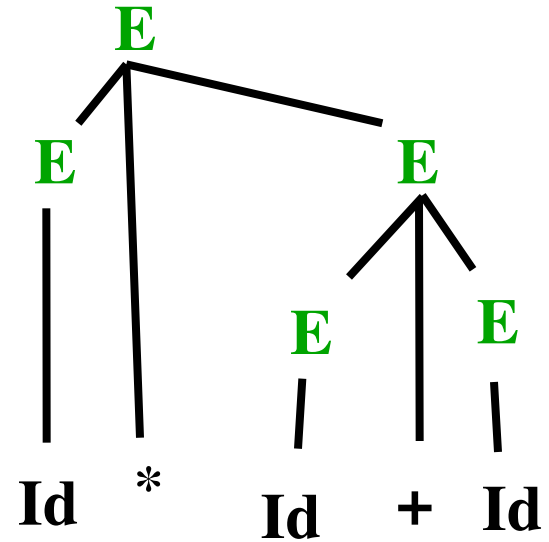
$E \rightarrow \text{Id}$

$E \Rightarrow E * E \Rightarrow \text{Id} * E$

$\Rightarrow \text{Id} * E + E \Rightarrow^* \text{Id} * \text{Id} + \text{Id}$

$E \Rightarrow E + E \Rightarrow E * E + E$

$\Rightarrow^* \text{Id} * \text{Id} + \text{Id}$



# AMBIGUITÁ - Es. 2

$E \rightarrow E + E$

$E \rightarrow E * E$

$E \rightarrow (E)$

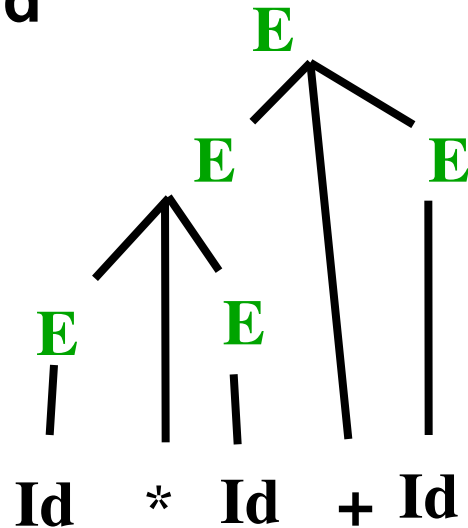
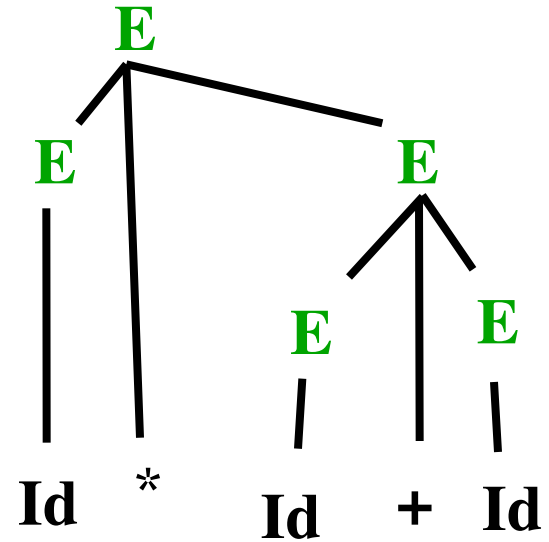
$E \rightarrow \text{Id}$

$E \Rightarrow E * E \Rightarrow \text{Id} * E$

$\Rightarrow \text{Id} * E + E \Rightarrow^* \text{Id} * \text{Id} + \text{Id}$

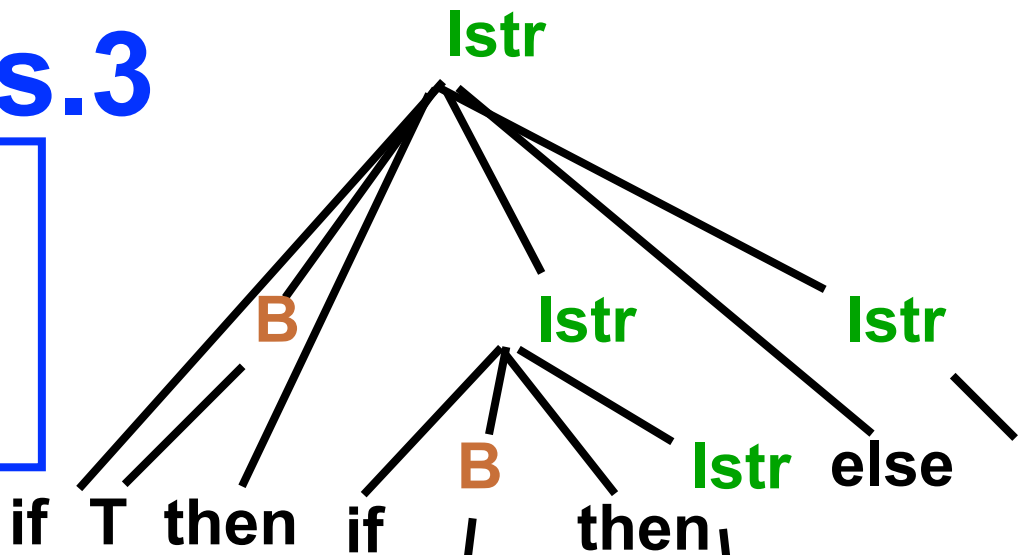
$E \Rightarrow E + E \Rightarrow E * E + E$

$\Rightarrow^* \text{Id} * \text{Id} + \text{Id}$

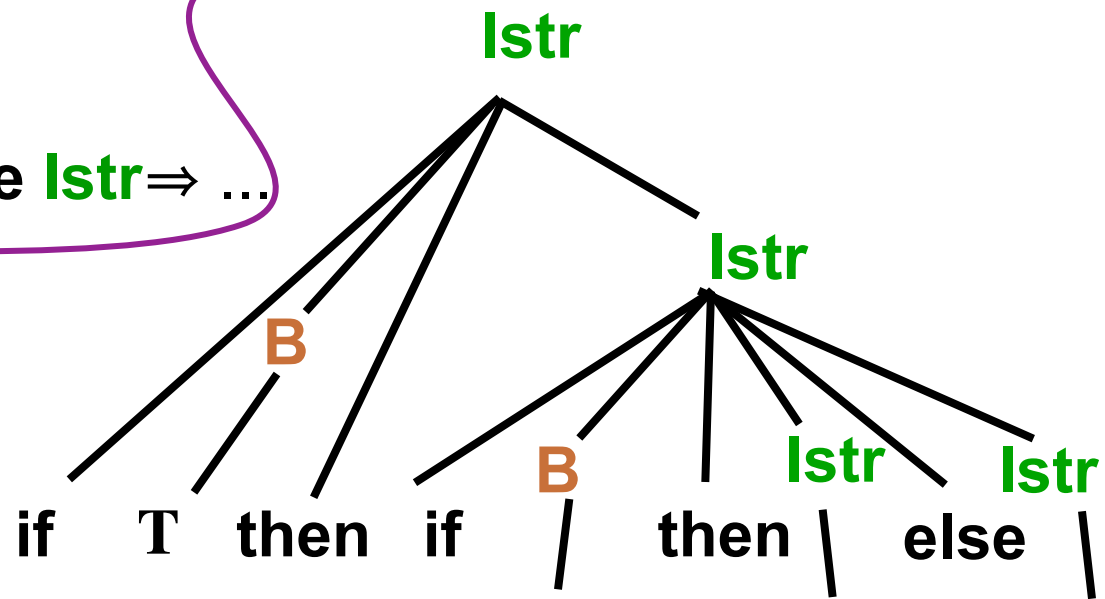


# AMBIGUITÁ - Es.3

**Istr** → if **B** then **Istr**  
 | if **B** then **Istr** else **Istr**  
 | **Altra-Istr**  
**B** → T | F | ...



**Istr** ⇒ if **B** then **Istr** else **Istr** ⇒  
 if T then **Istr** else **Istr** ⇒  
 if T then if **B** then **Istr** else **Istr** ⇒ ...



**Istr** ⇒ if **B** then **Istr** ⇒  
 if T then **Istr** ⇒  
 if T then if **B** then **Istr** else **Istr** ⇒ ...

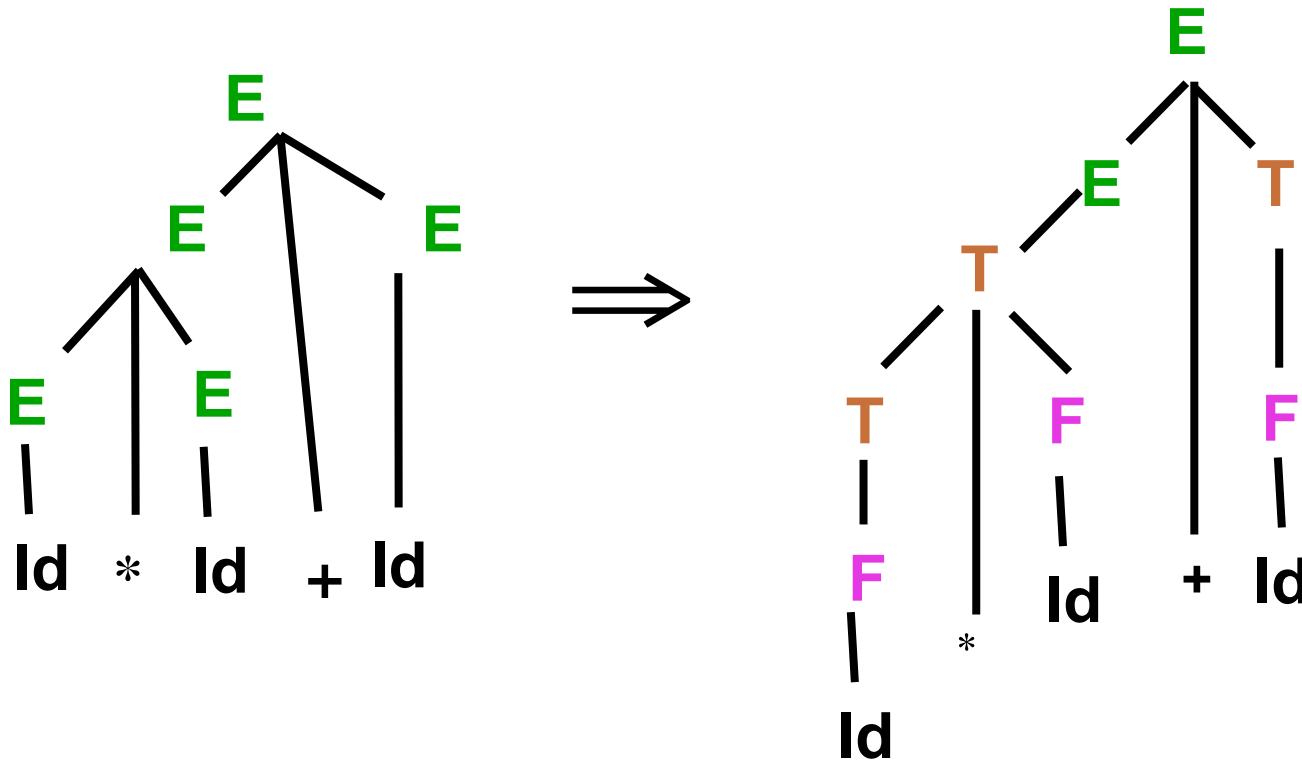
# AMBIGUITÀ: eliminazione ad hoc

$$E \rightarrow E + E \mid E * E \mid E \rightarrow (E) \mid Id$$



$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow (E) \mid Id \end{aligned}$$

Modifichiamo le regole in modo da attribuire una precedenza più alta al prodotto rispetto alla somma.



# AMBIGUITÁ: eliminazione ad hoc

**Istr** → if **B** then **Istr**  
| if **B** then **Istr** else **Istr**  
| **Altra-Istr**  
**B** → **T** | **F** | ...



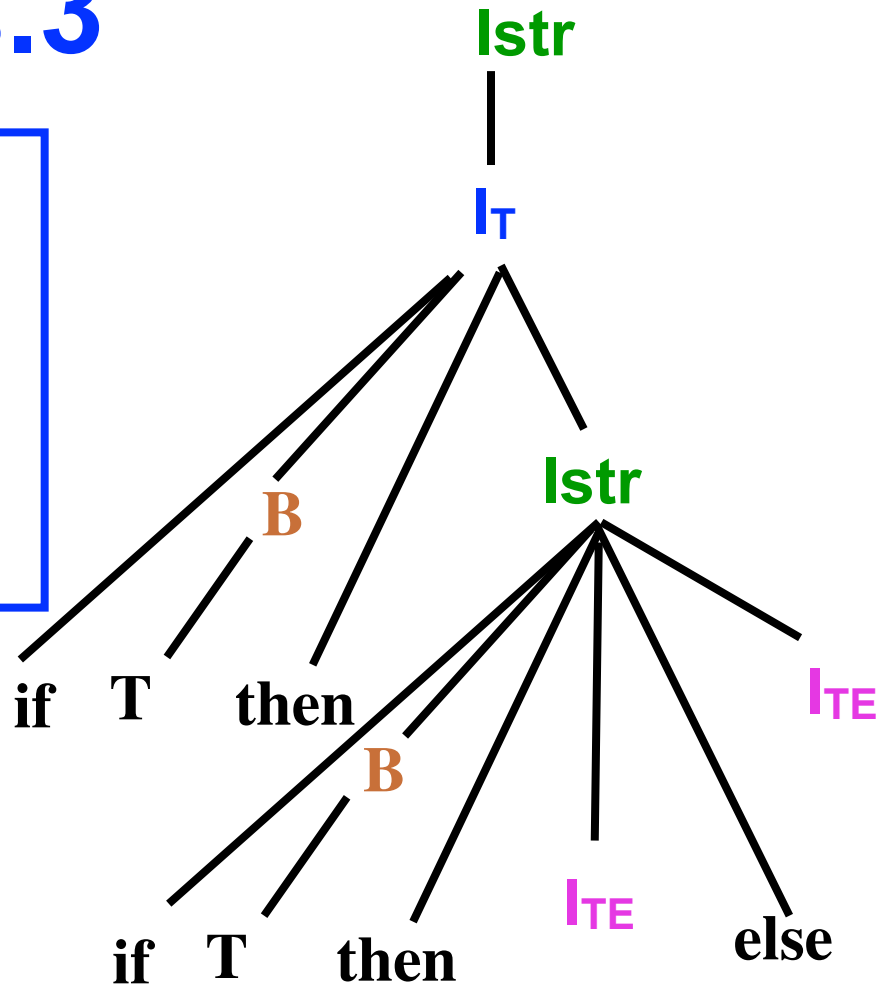
**Istr** → **I<sub>TE</sub>** | **I<sub>T</sub>** | ...  
**I<sub>T</sub>** → if **B** then **Istr**  
| if **B** then **I<sub>TE</sub>** else **I<sub>T</sub>**  
**I<sub>TE</sub>** → if **B** then **I<sub>TE</sub>** else **I<sub>TE</sub>**  
| **Altra-Istr**  
**B** → **T** | **F** | ...

Modifichiamo le regole in modo da incorporare la regola dell'accoppiamento dell'else: l'*else* va accoppiato al più vicino *then* non ancora accoppiato.

# AMBIGUITÁ - Es.3

$l_{str} \rightarrow l_{TE} \mid l_T \mid \dots$   
 $l_T \rightarrow \text{if } B \text{ then } l_{str}$   
 $\quad \mid \text{if } B \text{ then } l_{TE} \text{ else } l_T$   
 $l_{TE} \rightarrow \text{if } B \text{ then } l_{TE} \text{ else } l_{TE}$   
 $\quad \mid \text{Altra-}l_{str}$   
 $B \rightarrow T \mid F \mid \dots$

$l_{str} \Rightarrow l_T \Rightarrow \text{if } B \text{ then } l_{str} \Rightarrow$   
 $\text{if } B \text{ then } l_{TE} \Rightarrow$   
 $\text{if } B \text{ then } l_{TE} \text{ else } l_{TE}$



# AMBIGUITÁ: eliminazione ?

Disponiamo di un algoritmo che ricevendo in input una CFG, ne produce una equivalente non ambigua?

**NO!**

Anzi possiamo dimostrare che un tale algoritmo **NON** esiste.

Inoltre ci sono linguaggi che sono **inerentemente ambigui**, cioè tali da non ammettere una grammatica non ambigua. Per esempio:

$$L = \{a^i b^j c^k \mid i = j \text{ o } j = k \text{ e } i, j, k \geq 0\}$$



# Ambiguità facilmente eliminabile

$$L = \{a^m b^n \mid m > n > 0\}$$

$$G : S \rightarrow aSb \mid aS \mid aab$$

$$S \Rightarrow^* aSb \Rightarrow aaSb \Rightarrow a^3Sb \Rightarrow a^3aabb$$

$$S \Rightarrow aS \Rightarrow aaS \Rightarrow a^3Sb \Rightarrow a^3aabb$$

$$G' : S \rightarrow AB$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow aBb \mid ab$$

In  $G'$  in ogni derivazione leftmost prima si usano le regole per  $A$  e poi quelle di  $B$ , quindi l'albero di derivazione di ogni parola è unico.

# AMBIGUITÀ e DETERMINISMO

Se  $L$  è in  $L(\text{DPDA})$  allora **NON** è **inerentemente ambiguo**.

La costruzione standard della grammatica da un PDA fornisce una grammatica non ambigua, se il PDA di partenza è deterministico.