GRAMMATICA PER I COMANDI DEL LINGUAGGIO C

$$\Sigma_C = \{ a, b, c \ldots, 0, 1, 2, \ldots, =, !, <, >, \{,\}, \ldots, if, else, \ldots \}$$

Com → Ide = Exp ; | if (Exp) Com else Com
| while (Exp) Com | { SeqCom }

SeqCom → Com | Com SeqCom

Ide → Letter Seq

Letter → a | b | c | .... | z

Cif → 0 | .... | 9

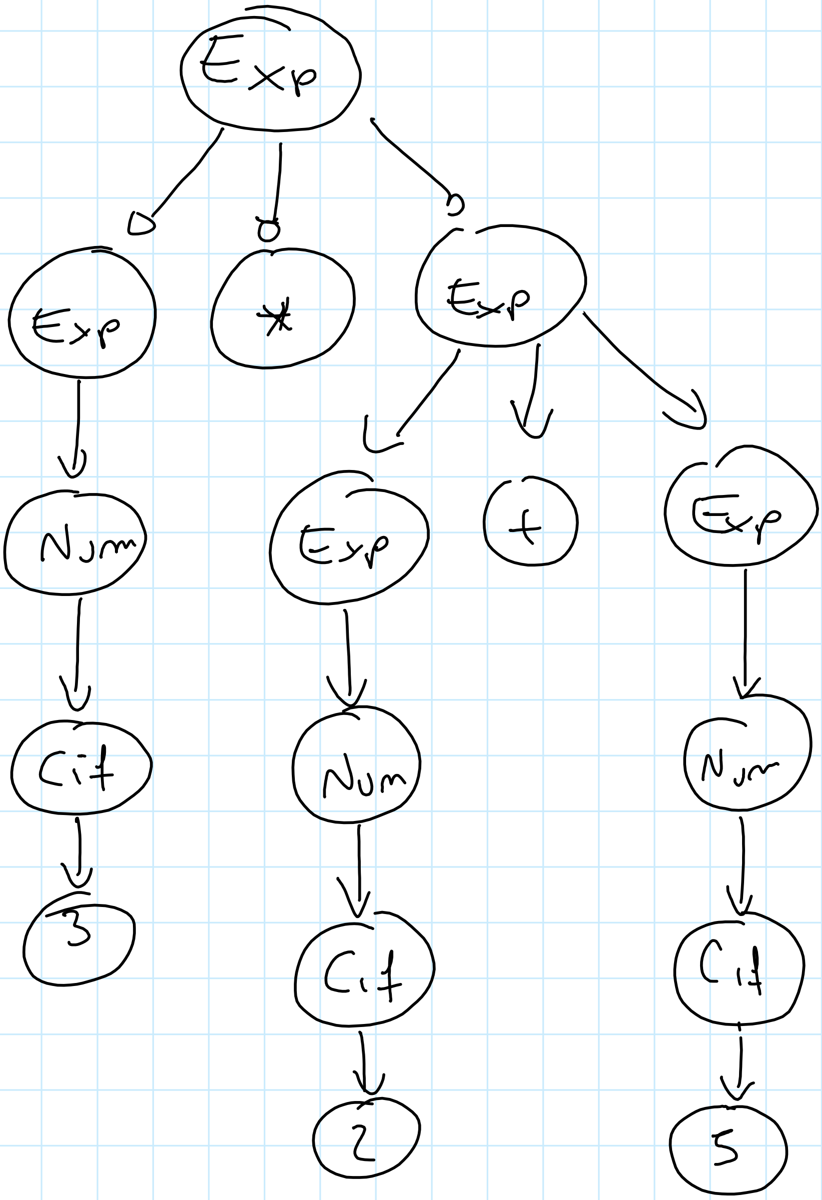
Seq → Letter | Cif | ε | Letter Seq | Cif Seq

Exp → Num | Exp + Exp | Exp * Exp | ...... | (Exp)

Num → Cif | Cif Num

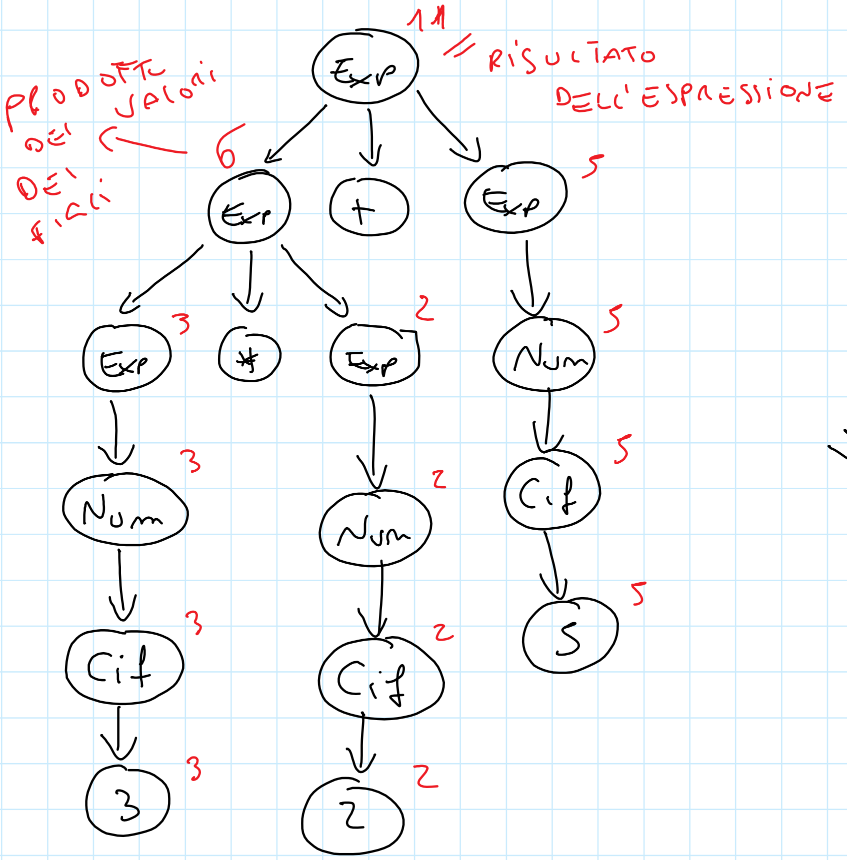
$$G_C = \{ \Sigma_C, \underbrace{V_C}_{= \{Com,\, Exp,\, \ldots\}}, Com, P_C \}$$

3 * 2 + 5 è un'espressione corretta?



3 * + 2 NON È SINTATTICAMENTE CORRETTA
(NON ESISTE UN ALBERO DI DERIVAZIONE)

$\underbrace{3*2}+\underbrace{5}$ AMMETTE UN ALTRO ALBERO DI DERIVAZIONE



⇒ LA GRAMMATICA È AMBIGUA

⇓

CONTIENE DOPPIA RICORSIONE IN Exp

SE CONSIDERO L'ALTRO ALBERO DI DERIVAZIONE MI VIENE UN RISULTATO DIVERSO (21) giovedì 19 ottobre 2017 11:38

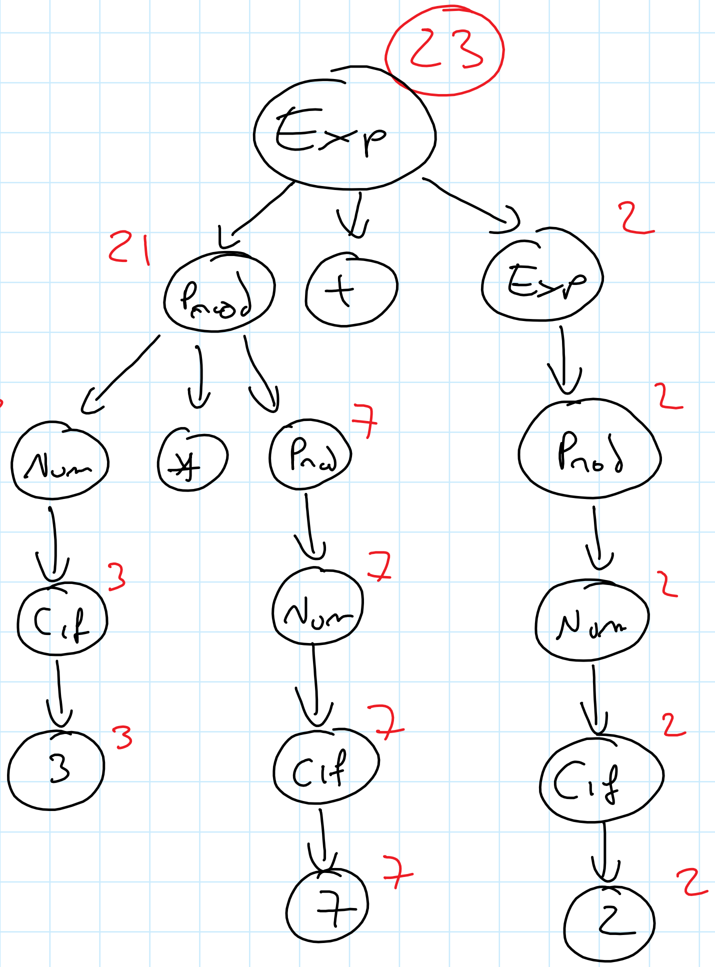
MODIFICHIAMO LA GRAMMATICA PER TOGLIERE L'AMBIGUITÀ RISPETTANDO LA PRECEDENZA DEL * SUL +

$$Exp \to Num \mid \cancel{Exp * Exp} \mid Exp + Exp$$

$$Exp \to Prod \mid Prod + Exp$$

$$Prod \to Num \mid Num * Prod$$

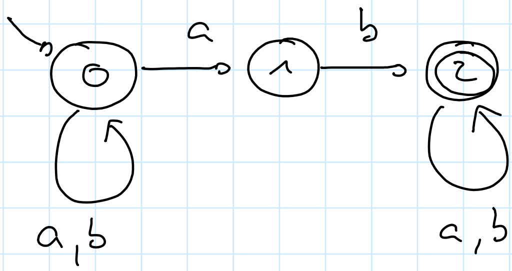
$$\underbrace{3 * 7}_{Prod} + \underbrace{2}_{Exp}$$



È L'UNICO ALBERO DI DERIVAZIONE POSSIBILE

RELAZIONE TRA AUTOMI E GRAMMATICHE

Abbiamo visto che un ASF può essere tradotto in una grammatica



$L = \{ \alpha a b \beta \mid \alpha, \beta \in \Sigma^* \}$

$\Downarrow$ USO GLI STATI COME CATEGORIE SINTATTICHE

$0 \rightarrow a0 \mid b0 \mid a1$

$1 \rightarrow b2 \mid b$

$2 \rightarrow a2 \mid a \mid b2 \mid b$

QUESTE DUE PRODUZIONI RAPPRESENTANO LA LETTURA DI b SEGUITO DA ALTRI SIMBOLI E LA LETTURA DI b COME ULTIMO SIMBOLO

FORMALMENTE

Dato un ASF

$$A = (\Sigma, Q, S, F, \delta)$$

una grammatica equivalente e'

$$G = (\Sigma, Q, S, \underline{P})$$

USO STATI
COME CATEGORIE
SINTATTICHE

dove :

→ se $\langle q, a, q' \rangle \in \delta$ $q, q' \in Q$, $a \in \Sigma$

$$q \to a q' \in P$$

→ inoltre, se $q' \in F$ allora

devo aggiungere ANCHE la seguente produzione

$$q \to a \in P$$

LO FACCIO SISTEMATICAMENTE PER TUTTE LE TRANSIZIONI IN δ E OTTENGO UNA GRAMMATICA EQUIVALENTE AD A

Le produzioni ottenute dalla traduzione di un automa hanno una forma particolare.

Sulla destra di ogni produzione abbiamo

- → un simbolo dell'alfabeto
- → oppure un simbolo dell'alfabeto seguito da una categoria sintattica

⇒ LE GRAMMATICHE DI QUESTO TIPO SONO DETTE

GRAMMATICHE REGOLARI

GRAMMATICHE LIBERE ⟷ **GRAMMATICHE REGOLARI**

$A \to \alpha$ con

$A \in V \quad \alpha \in (\Sigma \cup V)^*$

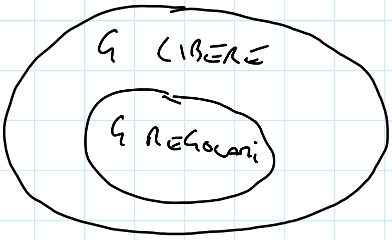
$\longleftrightarrow$

$A \to a$

$A \to aB$

$A, B \in V \quad a \in \Sigma$

Le grammatiche regolari sono un caso particolare di grammatica libera



Se Traduco un automa ottengo una grammatica regolare.

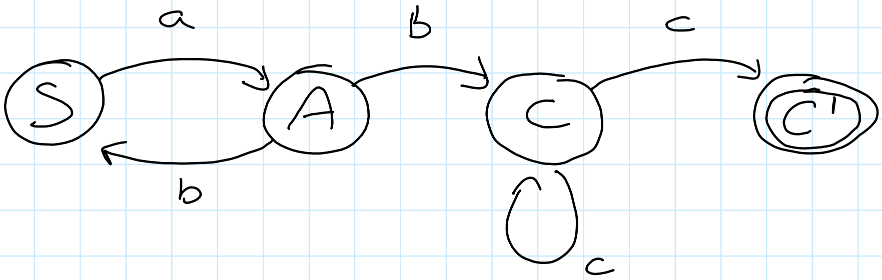
Ma, data una grammatica regolare qualunque posso sempre costruire un ASF equivalente?

SI

ESEMPI

$$S \to aA$$
$$A \to bS \mid bC$$
$$C \to cC \mid c$$

$$L = \{(ab)^n c^m \mid n, m > 0\}$$



Potendo tradurre ASF in grammatiche REGOLARI e viceversa ho dimostrato che l'insieme di tutti i linguaggi che questi due strumenti possono definire è lo stesso

ASF e GRAMMATICHE REGOLARI SONO EQUIVALENTI

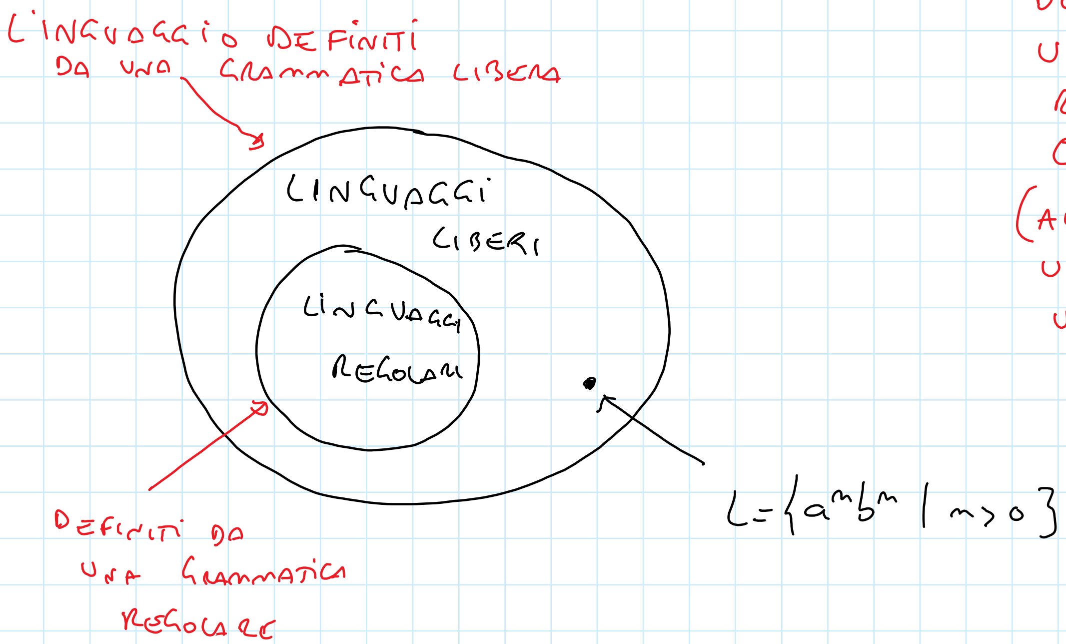
Ci sono linguaggi che posso definire con una grammatica LIBERA che non possono essere accettati da un ASF

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

NON È POSSIBILE COSTRUIRE UN AUTOMA PER QUESTO LINGUAGGIO CON UN NUMERO <u>FINITO</u> DI STATI

$$S \rightarrow ab \mid aSb$$

NON POSSO DEFINIRE UNA GRAMMATICA REGOLARE PER QUESTO LINGUAGGIO (ALTRIMENTI POTREI USARLA PER COSTRUIRE UN AUTOMA!!)



Dato un linguaggio libero, come faccio a capire se è regolare?

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

È REGOLARE? POSSO COSTRUIRE UN AUTOMA PER L?

Per rispondere a questa domanda bisogna utilizzare una proprietà dei linguaggi regolari descritta dal

PUMPING LEMMA

giovedì 19 ottobre 2017 12:40

NOTAZIONE

$|acb| = 3$

Pumping Lemma

$|w|$ indica la lunghezza della stringa w

Dato un linguaggio L, se L è regolare allora $\exists\, m \in \mathbb{N}$ tale che per tutte le stringhe $w \in L$ con $|w| \geq m$ vale la seguente proprietà:

$$w = xyz \qquad \exists\, x, y, z \in \Sigma^*$$

1) $|xy| \leq m$

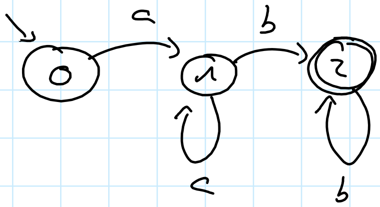
2) $y \neq \varepsilon$

3) $xy^iz \in L$ per $i \geq 0$

$\underset{i=0}{xz \in L}$ $\quad \underset{i=1}{xyz \in L}$ $\quad \underset{i=2}{xyyz \in L}$ ....

ESEMPIO:

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



$m = 3$

$\underbrace{a}_{x}\underbrace{a}_{y}\underbrace{b}_{z} \in L \qquad |aab| = 3 \geq m$

1) $|xy| = |aa| = 2 \leq m$ ✓

2) $y = a \neq \varepsilon$ ✓

3) $i = 0 \qquad xy^iz = xz = ab \in L \qquad k=1 \; m=1$

$i = 1 \qquad xy^iz = xyz = a\underline{a}b \in L$

$i = 2 \qquad xy^iz = xyyz = a\underline{a}\,\underline{a}b \in L \qquad k=3 \; m=1$

$\vdots \qquad \vdots \qquad \vdots$

PER OGNI LINGUAGGIO REGOLARE, DATA UNA STRINGA SUFFICIENTEMENTE LUNGA ($\geq m$) POSSO SICURAMENTE TROVARE UN MODO DI SUDDIVIDERLA IN x, y, z IN MODO DA SODDISFARE LE 3 PROPRIETÀ.