Chiusura rispetto a unione CFL

Sia $G_1 = (V_1, T, S_1, P_1)$ e $G_2 = (V_2, T, S_2, P_2)$, con $V_1 \cap V_2 \neq \emptyset$, allora la grammatica per l'unione è

$$G = (V, T, S, P) \text{ dove:}$$

$$S \notin V_1 \cup V_2$$

$$V = V_1 \cup V_2 \cup \{S\}$$

$$P = P_1 \cup P_2 \cup \{S \to S_1 | S \to S_2\}$$

Chiusura rispetto a intersezione CFL

La classe NON è chiusa rispetto all'intersezione.

Il linguaggio

$$L = \{a^n b^n c^n \mid n \geq 0\}$$

non è content-free, ma può essere ottenuto come intersezione di due linguaggi contesto-free:

$L_1 = \{a^n b^n c^m \mid n,m \geq 0\}$ e $L_2 = \{a^m b^n c^n \mid n,m \geq 0\}$.

Chiusura rispetto a complemento?

NO! data la chiusura rispetto all'unione.

Chiusura rispetto a prodotto CFL

Sia $G_1 = (V_1,T,S_1,P_1)$ e $G_2 = (V_2,T,S_2,P_2)$, con $V_1 \cap V_2 \neq \varnothing$, allora la grammatica per il prodotto è

$G = (V,T,S,P)$ dove:

$$S \notin V_1 \cup V_2$$

$$V = V_1 \cup V_2 \cup \{S\}$$

$$P = P_1 \cup P_2 \cup \{S \rightarrow S_1 S_2\}$$

Chiusura rispetto a stella di Kleene CFL

Sia $G_1 = (V_1,T,S_1,P_1)$, allora la grammatica per la stella di Kleene è

$G = (V,T,S,P)$ dove:

$S \notin V_1$

$V = V_1 \cup \{S\}$

$P = P_1 \cup \{S \rightarrow S_1S, S \rightarrow \varepsilon\}$

Complemento per CFL

Sappiamo che L= {ww | w in {0,1}*} non è CFL.

Faremo vedere che il suo complemento lo è.

In un primo momento faremo vedere che il linguaggio

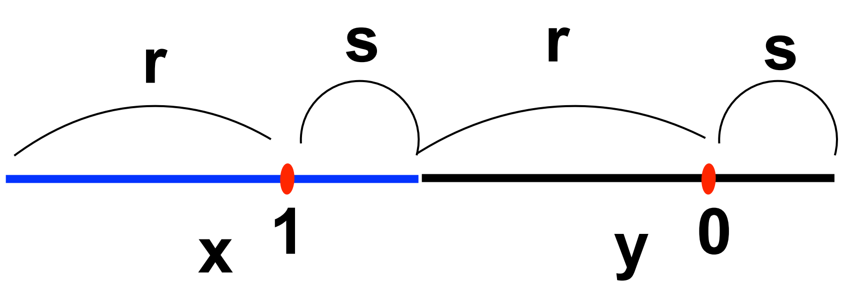
$$L_c = \{xy \mid x,y \text{ in } \{0,1\}^* \text{ e } |x|=|y| \text{ e } x \neq y\}$$

è un CFL, poi basterà aggiungere le parole di lunghezza dispari.

L_c è context-free

$L_c = \{xy \mid x,y \text{ in } \{0,1\}^* \text{ e } |x|=|y| \text{ e } x \neq y\}$

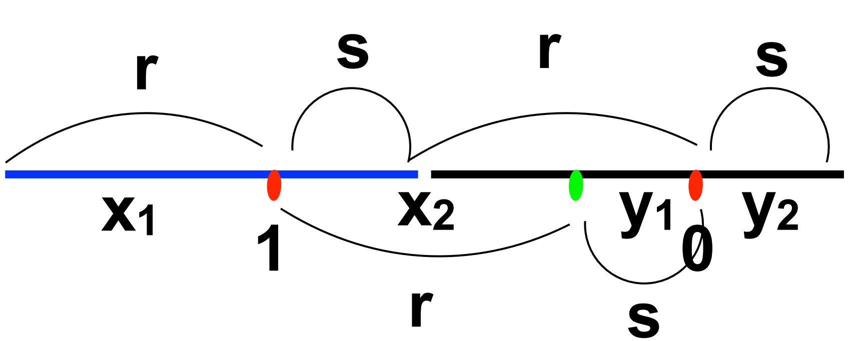
Sia $w=xy$ in L_c, poiché $x \neq y$ e $|x| = |y|$, vuol dire che esiste (almeno) una posizione in x e in y in cui differiscono:



Esempio: $w=xy$

$x= 0^r10^s$ e $y= 0^r00^s$

Posso scomporre la parola in modo diverso

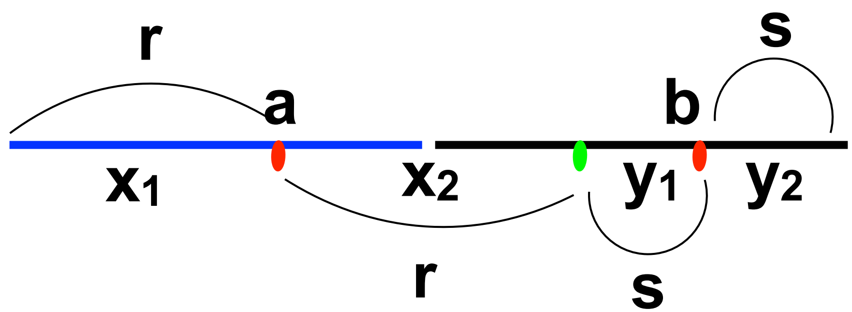


Esempio: $w=x_1 1 x_2 y_1 0 y_2$

$x_1 = 0^r$, $x_2 = 0^r$, $y_1 = y_2 = 0^s$

L_c è context-free

$L_c = \{xy \mid x,y \text{ in } \{0,1\}^* \text{ e } |x|=|y| \text{ e } x \neq y\}$



Allora devo considerare parole del tipo $w=x_1ax_2y_1by_2$, dove x_1,x_2,y_1,y_2 sono in $\{0,1\}^*$, $a \neq b$ in $\{0,1\}$, $|x_1| = |x_2|$ e $|y_1| = |y_2|$.

Le parole x_1ax_2 e y_1by_2 sono dello stesso tipo e sono semplicemente parole di lunghezza dispari.

Quindi le parole di L_c sono la concatenazione di parole di lunghezza dispari di centro diverso

Le grammatiche per le parole dispari di centro diverso

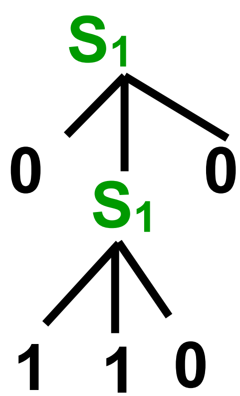
G1: $S_0 \rightarrow 0 \mid 0S_00 \mid 0S_01 \mid 1S_00 \mid 1S_01$
G2: $S_1 \rightarrow 1 \mid 0S_10 \mid 0S_11 \mid 1S_10 \mid 1S_11$

L(G1) = {x0y | x,y in {0,1}* e |x|=|y|}

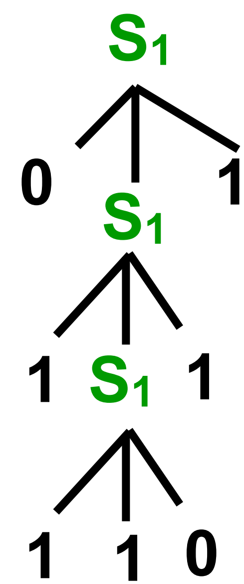
L(G2) = {x1y | x,y in {0,1}* e |x|=|y|}

Per esempio vediamo come derivare una qualsiasi parola dispari con centro 1

01 1 00



011 1 011



La grammatica per L_c

$L_c = \{xy \mid x,y \text{ in } \{0,1\}^* \text{ e } |x|=|y| \text{ e } x \neq y\}$

Abbiamo visto che queste parole sono del tipo $w=x_1ax_2y_1by_2$, dove x_1,x_2,y_1,y_2 sono in $\{0,1\}^*$, $a \neq b$ in $\{0,1\}$ e $|x_1|=|x_2|$ e $|y_1| =|y_2|$

Ora costruire la grammatica per L_c è facile:

G3: $S \rightarrow S_0S_1 \mid S_1S_0$

$S_0 \rightarrow 0 \mid 0S_00 \mid 0S_01 \mid 1S_00 \mid 1S_01$

$S_1 \rightarrow 1 \mid 0S_00 \mid 0S_01 \mid 1S_00 \mid 1S_01$

Esempio di derivazione

Ora costruire la grammatica per L_c è facile:

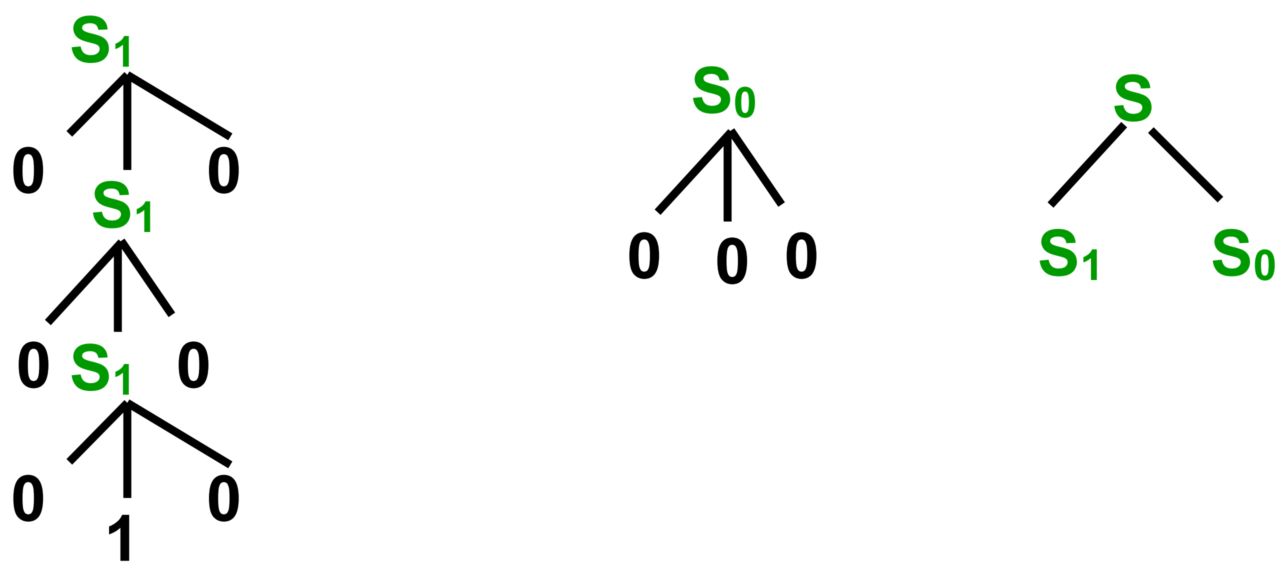
G3: $S \rightarrow S_0S_1 \mid S_1S_0$

$S_0 \rightarrow 0 \mid 0S_00 \mid 0S_01 \mid 1S_00 \mid 1S_01$

$S_1 \rightarrow 1 \mid 0S_00 \mid 0S_01 \mid 1S_00 \mid 1S_01$

w = xy = 0^3100^300 con x=0^310 e y =0^300, che vediamo come

zt con z = 0^310^3 e t = 000



La grammatica per il complemento

L= {ww | w in {0,1}*} non è CFL.

$\neg L = L_c \cup \{x \mid x \text{ in } \{0,1\}^* \text{ tali che } |x| \text{ dispari}\}$

Ora costruire la grammatica per ¬L è facile:

G4: $S \rightarrow S_0 \mid S_1 \mid S_0 S_1 \mid S_1 S_0$

$S_0 \rightarrow 0 \mid 0S_00 \mid 0S_01 \mid 1S_00 \mid 1S_01$

$S_1 \rightarrow 1 \mid 0S_00 \mid 0S_01 \mid 1S_00 \mid 1S_01$

Decidere se L = $\emptyset$, quando L è CFL

Algoritmo per decidere se il linguaggio generato da una CFG è vuoto.

Input: una CFG =(V,T,S,P)

Marca i terminali

1. **Ripeti fino a quando non ci sono variabili che si possono marcare**

   marca ogni variabile parte sinistra di una regola in cui la parte destra è marcata

3. **Se S non è marcato rispondi sì, altrimenti no.**

Esempio:
G: $S \rightarrow AB \mid AC$
$C \rightarrow SB$
$A \rightarrow a$
$B \rightarrow b$

Alla prima iterazione si marcano A e B, alla seconda S, alla terza C e fine, perchè sono tutte marcate.

Esercizi

Si consideri il linguaggio $L = \{a^n b^{n+m} c^m \mid n,m \geq 1\}$, si costruisca una CFG che lo genera e un PDA che lo accetta. Si utilizzino le proprietà di chiusura viste per ottenere una grammatica leggibile facilmente.