Linguaggi Acontestuali ad estensione dei regolari

Dato l'alfabeto Σ, definizione come ***classe dei linguaggi acontestuali*** Σ il seguente insieme:

$$\text{CFL} = \{L \subseteq \Sigma^* \mid \exists CFG\ G\ t.c\ L = L(G)\}$$

Conversione da DFA a CFG

Date due classi dei linguaggi REG e CFL, si ha che:

$$\text{REG} \subseteq \text{CFL}$$

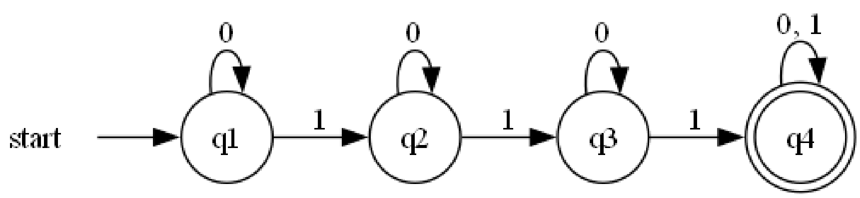
Dimostrazione:

- Dato $L \in$ REG, sia $D = (Q, \Sigma, \delta, q_0, F)$ il DFA tale che $L = L(D)$.
- Consideriamo quindi la **CFG** $G = (V, \Sigma, R, S)$ tale che:
  - Esiste una funzione biettiva: $\varphi : Q \to V : q_i \mapsto V_i$
  - S = $\varphi(q_0) = V_0$k
  - Dati $q_i, q_j \in Q$ e $a \in \Sigma$, si ha che: $\delta(q_i, a) = q_j \implies \varphi(q_i) \to a\varphi(q_j) \implies V_i \to aVj$
  - $q_f \in F \implies \varphi(q_f) \to \varepsilon \implies V_f \to \varepsilon$
  - A questo punto, per costruzione stessa di G si ha che: $w \in L(D) \iff w \in L(G)$ implicando dunque che $L(D) \in$ CFL e di conseguenza che: REG $\subseteq$ CFL

$\square$

Esempio

Consideriamo il seguente DFA



- Una CFG G = (V, Σ, R, S) equivalente è costituita da:
  - $V = V_1, V_2, V_3, V_4$
  - $S = V_1 – R$ definito come:
  - $V_1 \to 0V_1 \mid 1V_2$
  - $V_2 \to 0V_2 \mid 1V_3$
  - $V_3 \to 0V_3 \mid 1V_4$
  - $V_4 \to 0V_4 \mid 1V_4 \mid \varepsilon$

  Infatti, sia il **DFA** sia la **CFG** descrivono il seguente linguaggio: $L = \{w \in \Sigma^* \mid |w|_1 \geq 3\}$

Linguaggi acontestuali estensione dei linguaggi regolari

> Teorema
>
> Date le due classi dei linguaggi **REG** e **CFL**, si ha che: REG $\subsetneq$ CFL

Forma normale di Chomsky

Chomsky's Normal Form ***(CNF)***

Una $CFG\ G = (V, \Sigma, R, S)$ viene detta in ***Chomsky's Normal Form (CNF)*** se tutte le regole in R assumono una delle seguenti tre forme:

$$A \to BC \quad A \to a \quad S \to \varepsilon$$

dove $A \in V, a \in \Sigma$ e $B, C \in V - S$.

Teorema: Conversione in Forma Normale di Chomsky

Per ogni CFG G, si ha che:

$$\exists \text{ CFG } G' \text{ in CNF} \mid L(G) = L(G')$$

Dimostrazione

Data una **CFG** $G = (V, \Sigma, R, S)$, costruiamo una **CFG** G' in **CNF** equivalente a G:

1. Aggiunta della variabile iniziale

Viene aggiunta una nuova variabile S_0 e una regola:

$$S_0 \to S$$

dove S_0 è la nuova variabile iniziale.

---

2. Eliminazione delle ε-regole

Finché in R esiste una ε-regola $A \to \varepsilon$ dove $A \in V - \{S_0\}$:

- tale regola viene eliminata;
- per ogni regola in R contenente occorrenze di A, vengono aggiunte nuove regole ottenute eliminando tutte le possibili combinazioni di occorrenze di A.
- **Esempio:**
  Se viene rimossa $A \to \varepsilon$ e in R esiste $B \to uAvAw$ con $u, v, w \in (V \cup \Sigma)^*$,
  vengono aggiunte le regole:

$$B \to uvAw \mid uAvw \mid uvw$$

---

3. Eliminazione delle regole unitarie

Ogni regola nella forma $A \to B$ (dette *regole unitarie*) viene eliminata.
Per ogni regola nella forma $B \to u$, dove $u \in (V \cup \Sigma)^*$, si aggiunge la regola corrispondente $A \to u$

4. Riduzione delle regole con più di due simboli

Per ogni regola $A \to u_1u_2 \dots u_k$ con $k \geq 3$ e $\forall i \in [1, k],\ u_i \in (V \cup \Sigma)$:

- vengono aggiunte nuove variabili $A_1, A_2, \dots, A_{k-2}$;
- si introducono le seguenti regole:

$$\begin{aligned} A &\to u_1A_1 \\ A_1 &\to u_2A_2 \\ &\vdots \\ A_{k-3} &\to u_{k-2}A_{k-2} \\ A_{k-2} &\to u_{k-1}u_k \end{aligned}$$

La regola iniziale $A \to u_1u_2 \dots u_k$ viene poi eliminata.

---

5. Sostituzione dei terminali nelle regole binarie

Per ogni regola rimanente della forma $A \to u_1u_2$ con $u_1, u_2 \in (V \cup \Sigma)$:

- se $u_1 \in \Sigma$, si aggiunge una nuova variabile U_1 e una regola $U_1 \to u_1$,
  sostituendo la regola originale con $A \to U_1u_2$;
- analogamente, se $u_2 \in \Sigma$, si introduce una nuova variabile U_2 con regola $U_2 \to u_2$.

Conclusione

Poiché le operazioni svolte dall'algoritmo **non modificano le stringhe generabili** dalla CFG, segue automaticamente che:

$$L(G) = L(G')$$

Esempio di trasformazione in CNF

□

Consideriamo la seguente grammatica G non in CNF, dove S è la variabile iniziale:
$G : S \to ASA \mid aB$
$A \to B \mid S$
$B \to b \mid \varepsilon$

- Aggiungiamo S_0 e la regola $S_0 \to S$:
  $G : S_0 \to S$
  $S \to ASA \mid aB$
  $A \to B \mid S$
  $B \to b \mid \varepsilon$
- Eliminazione della ε-regola $B \to \varepsilon$
  $G : S_0 \to S$
  $S \to ASA \mid aB \mid a$
  $A \to B \mid S \mid \varepsilon$
  $B \to b \mid \cancel{\varepsilon}$
- Eliminazione della ε-regola $A \to \varepsilon$
  $G : S_0 \to S$
  $S \to ASA \mid aB \mid a \mid SA \mid AS \mid S$
  $A \to B \mid S \mid \cancel{\varepsilon}$
  $B \to b$

- Eliminazione di $S \to S$:
  $G : S_0 \to S$
  $S \to ASA \mid aB \mid a \mid SA \mid AS \mid \cancel{S}$
  $A \to B \mid S$
  $B \to b$
- Eliminazione di $S_0 \to S$:
  $G : S_0 \to \cancel{S} \mid ASA \mid aB \mid a \mid SA \mid AS$
  $S \to ASA \mid aB \mid a \mid SA \mid AS$
  $A \to B \mid S$
  $B \to b$
- Eliminazione di $A \to B$ e $A \to S$:
  $G : S_0 \to ASA \mid aB \mid a \mid SA \mid AS$
  $S \to ASA \mid aB \mid a \mid SA \mid AS$
  $A \to \cancel{B} \mid \cancel{S} \mid b \mid ASA \mid aB \mid a \mid SA \mid AS$
  $B \to b$
- Separazione delle regole con più di due elementi a destra
  $G : S_0 \to \cancel{ASA} \mid AA_1 \mid aB \mid a \mid SA \mid AS$
  $S \to \cancel{ASA} \mid AA_1 \mid aB \mid a \mid SA \mid AS$
  $A \to b \mid \cancel{ASA} \mid AA_1 \mid aB \mid a \mid SA \mid AS$
  $A_1 \to SA$
  $B \to b$
- Conversione dei terminali nelle regole binarie
  $G : S_0 \to AA_1 \mid \cancel{aB} \mid UB \mid a \mid SA \mid AS$
  $S \to AA_1 \mid \cancel{aB} \mid UB \mid a \mid SA \mid AS$
  $A \to b \mid AA_1 \mid \cancel{aB} \mid UB \mid a \mid SA \mid AS$
  $A_1 \to SA$
  $U \to a$
  $B \to b$

Grammatica finale in CNF

$G : S_0 \to AA_1 \mid UB \mid a \mid SA \mid AS$
$S \to AA_1 \mid UB \mid a \mid SA \mid AS$
$A \to b \mid AA_1 \mid UB \mid a \mid SA \mid AS$
$A_1 \to SA$
$U \to a$
$B \to b$