Pumping Lemma

Se prendiamo in considerazione un linguaggio composto da stringhe aventi un numero uguali di 0 e 1:

$$L = \{0^n 1^n \mid n \in \mathbb{N}\}$$

Nel tentativo di creare un automa che riconosca tale linguaggio, si nota che servirebbe un' automa che avesse ***stati infiniti***, poiché dovrebbe memorizzare la quantità di 0 e 1 letti. Quindi, non è possibile costruire un automa a stati finiti che riconosca tale linguaggio.

> **Pumping lemma per i linguaggi regolari**
>
> Dato un linguaggio L, se $L \in REG$ allora $\exists p \in \mathbb{N}$, detto ***lunghezza del pumping***, tale che $\forall\ w := xyz \in L$, con $|w| \geq p$ e $x, y, z \in \Sigma^*$ (ovvero sono sottostringhe), si ha:
>
> - $\forall i \in \mathbb{N}\ xy^iz \in L$, ovvero che è possibile concatenare y per i volte rimanendo in L
> - $|y| > 0$, quindi $y \neq \epsilon$
> - $xy \leq p$, ossia y deve trovarsi nei primi p simboli di w

Contesto

Il lemma serve a **caratterizzare i linguaggi regolari**:
se un linguaggio L è **regolare**, allora deve rispettare **una certa proprietà strutturale**.
Questa proprietà riguarda il modo in cui si possono "ripetere" (o *pompare*) dei pezzi delle parole di L restando sempre dentro al linguaggio.

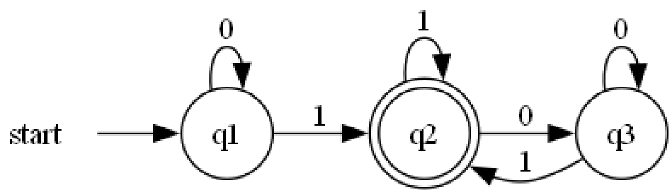
Dimostrazione:

- Dato $L \in REG$. Poiché L è regolare, esiste un **automa a stati finiti deterministico (DFA)** $D := (Q, \Sigma, \delta, q_0, F)$ tale che $L = L(D)$, cioè il linguaggio riconosciuto da D.
- Si consideri $p := |Q|$. cioè **il numero di stati** dell'automa. Questo è importante perché se una stringa è più lunga di p, l'automa, nel leggerla, sarà costretto a **visitare almeno uno stato due volte**. Data la stringa $w := w_1 ... w_n \in L$ dove $w_1, \dots, w_n \in \Sigma$ e dove $n \geq p$, consideriamo la sequenza di stati $r_1, \dots, r_{n+1}$ tramite cui w viene accettata da D, dove:
  - $r_1 = q_0$ (stato iniziale)
  - $r_{k+1} = \delta(r_k, w_k)$ per ogni $k = 1, \dots, n$
    Alla fine, $r_{n+1} \in F$, perché $w \in L$.
- Si nota quindi che $|r_1, \dots, r_{n+1}| = n + 1$, ossia che il numero di stati attraversati sia $n + 1$. Inoltre, in quanto $n \geq p$, ne deriva che $n + 1 \geq p + 1$. Tuttavia, poiché $p := |Q|$ e $n + 1 \geq p + 1$, ne segue necessariamente che $\exists i, j \mid 1 \leq i < j \leq p + 1 \land r_i = r_j$, ovvero lo stesso stato viene raggiunto due volte entro i primi $p + 1$ passi.
- Considerando ora le seguenti sottostringhe di w:
  - $x = w_1 ... w_{i-1}$, tramite cui si ha che $\delta * (r_1, x) = r_i$
  - $y = w_i ... w_{j-1}$, tramite cui si ha che $\delta * (r_i, y) = r_j = r_i$ → quindi y è un **ciclo** nell'automa.
  - $z = w_j ... w_n$, tramite cui si ha che $\delta * (r_j, z) = r_{n+1}$
- Poiché $\delta * (r_i, y) = r_i$, ossia y porta sempre r_i in se stesso, ne segue automaticamente che

$$\forall k \in N\ \delta^*(r_i, y_k) = r_i \implies \delta(r_1, xy^kz) = r_{n+1} \in F \implies xy^kz \in L(D) = L$$

Esempio $\square$

Si consideri il linguaggio $L = \{x \in \{0,1\}^* \mid x := y1,\ \exists y \in \{0,1\}^*\}$



Dato che è un linguaggio regolare, vale il ***Pumping Lemma***. Ad esempio, se si prende $p = 5$ e la stringa $w := 0100010101 \in L$, si può separare w in 3 sottostringhe $x := 010$, $y := 00$ e $z := 10101$ tali che:

- $xy^0z = 01010101 \in L$
- $xy^1z = 0100010101 \in L$
- $xy^2z = 010000010101 \in L$
- $xy^3z = 01000000010101 \in L$
- $\ldots$

Osservazione

Il Pumping lemma per i linguaggi regolari può essere utilizzato per dimostrare che un linguaggio non è regolare.

Esempio:

- Consideriamo il linguaggio $L = 0^n1^n \mid n \in N$
- Supponiamo per assurdo che L sia regolare. In tal caso, ne segue che per esso debba valere il pumping lemma, dove p è la lunghezza del pumping.
- Consideriamo quindi la stringa $w := 0^p1^p \in L$. Dato che $|w| \geq p$, possiamo suddividerla in tre sottostringhe $x, y, z \in \Sigma^*$ tali che $w = xyz$:
  - Se y è composta da soli 0, allora ogni stringa generata dal pumping non sarà in L in quanto il numero di 0 sarà superiore al numero di 1
  - Se y è composta da soli , allora ogni stringa generata dal pumping non sarà in L in quanto il numero di sarà superiore al numero di 1
  - Se y è composta sia da 0 che da 1, allora ogni stringa generata dal pumping non sarà in L in quanto esse assumeranno la forma $0000 \ldots 101010 \ldots 1111$
- Di conseguenza, poiché in ogni caso viene contraddetto il pumping lemma, ne segue necessariamente che L non sia regolare.