TEOREMA (PUMPING LEMMA)

Per ogni linguaggio regolare L $\exists$ una costante n tale che, se $z \in L$ e $|z| \geq n$, allora possiamo scrivere $z = uvw$ con $|uv| \leq n$ $|v| \geq 1$ e ottenere che $uv^iw \in L$ per ogni $i \geq 0$

OSS

Questo teorema ci da una condizione necessaria ma non sufficiente per i linguaggi regolari

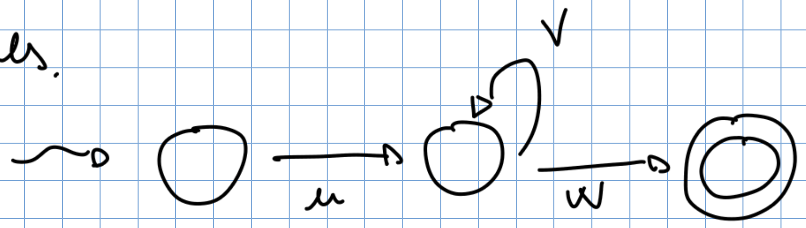
Non posso utilizzarlo per dimostrare che un linguaggio L è regolare ma posso usarlo per dimostrare che un L' non è regolare

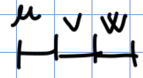
INFORMALE

Se L è regolare $\exists$ un numero n che ha queste caratteristiche:

Se ho una stringa z che appartiene al linguaggio $\geq n$ la posso dividere in 3 parti, ognuna lunga "u" oppure "w" e la parte v la posso ripetere quante volte voglio ottenendo ancora stringhe $\in L$

es.



z 

$z \in L(A)$

DIMOSTRAZIONE

Dato un linguaggio L regolare $\exists$ ASFD che lo riconosce. Sia $A = < \Sigma, Q, \Delta, q_0, F >$ un automa che riconosce L e sia $|Q| = n$

Sia $z \in L$ con $|z| = k \geq n \Rightarrow \bar{\Delta}(q_0, z) \in F$

Supponiamo che $q_{i_0}, q_{i_1}, \ldots\ldots, q_{i_k}$ sia la sequenza di stati attraversati durante la computazione z con $q_{i_0} = q_0$ e $q_{i_k} = \bar{\Delta}(q_0, z) \in F$

In altre parole q_{i_1} è lo stato in cui arriva l'automa A dopo aver letto il primo simbolo di z, q_{i_2} lo stato in cui arriva l'automa A dopo aver letto il secondo simbolo di z ....

Indichiamo con z_h il prefisso di z di lunghezza h, allora avremo che q_{i_h} è lo stato in cui arriva l'automa

A dopo aver letto z_h elementi, cioè $q_{i_h} = \bar{\delta}(q_0, z_h)$

Dal momento che $k \geq n$ deve esistere almeno 1 stato in cui l'automa si porta almeno 2 volte durante la computazione su z, cioè esistono 2 stati nella sequenza $q_{i_0}, q_{i_1}, \ldots, q_{i_k}$ che coincidono

In realtà questi 2 stati che coincidono si possono trovare già tra i primi $n+1$ elementi della sequenza $q_{i_0}, q_{i_1}, \ldots, q_{i_k}$ che coincidono

Possiamo affermare che esistono 2 indici r, s con $0 \leq r < s \leq n$ tali che $q_{i_r} = q_{i_s}$

Cioè lo stato in cui arriva l'automa leggendo il prefisso di z di lunghezza r (leggendo z_r) è esattamente lo stesso stato in cui arriva l'automa leggendo il prefisso di z di lunghezza s (leggendo z_s) cioè ancora:

$$\bar{\delta}(q_0, z_r) = q_{i_r} = q_{i_s} = \bar{\delta}(q_0, z_s)$$

Poniamo $u = z_r \quad uv = z_s \quad uvw = z$

quindi $|uv| = |z_s| = s \leq n$ e $|v| \geq 1$

perché $|u| = |z_r| = r < s = |z_s| = |uv|$

Inoltre dobbiamo dimostrare che $uv^i w \in L \quad \forall i \geq 0$

INDUZIONE:

PASSO BASE: $i = 0$ $\qquad \overset{*}{\bar{\delta}(q_0, z_r) = q_{i_r} = q_{i_s} = \bar{\delta}(q_0, z_s)}$

$$\bar{\delta}(q_0, uv^0w) = \bar{\delta}(q_0, u\varepsilon w) = \bar{\delta}(q_0, uw) = \bar{\delta}(\bar{\delta}(q_0, u), w) \overset{*}{=}$$

$$= \bar{\delta}(\bar{\delta}(q_0, z_s), w) = \bar{\delta}(\bar{\delta}(q_0, uv), w) = \bar{\delta}(q_0, uvw) = \bar{\delta}(q_0, z) \in F$$

cioè $uv^0w = uw \in L$

PASSO INDUTTIVO: $\quad i > 0$

PER IPOTESI INDUTTIVA $uv^{i-1}w \in L$ cioè $\bar{\delta}(q_0, uv^{i-1}w) \in F$

Allora $\bar{\delta}(q_0, uv^i w) = \bar{\delta}(q_0, uv\underbrace{v^{i-1}w}) = \bar{\delta}(\bar{\delta}(q_0, uv), v^{i-1}w) =$

$$= \bar{\delta}(\bar{\delta}(q_0, z_r), v^{i-1}w) = \bar{\delta}(\bar{\delta}(q_0, u), \underbrace{v^{i-1}w}) = \bar{\delta}(q_0, uv^{i-1}w) \in F$$

$\bar{\delta}(q_0, uv^i w) \in F \Rightarrow uv^i w \in L$

GRAMMATICHE

linguaggi formali $\Rightarrow$ GRAMMATICHE

Problema di ordine a calcolare una funzione $f: \mathbb{N} \to \mathbb{N}$ può innescare un nuovo problema: quello di calcolare e riconoscere un linguaggio

$$L = \{ a^n b^{f(n)} \mid n \geq 0 \}$$

- possiamo usare le espressioni regolari per definire un linguaggio formale
- un altro approccio è quello generativo dove si usano strumenti formali che sono appunto le grammatiche formali, che consentono di costruire delle regole di produzione
- approccio riconoscitivo che consiste nell'utilizzare macchine astratte dette automi riconoscitori che definiscono algoritmi di riconoscimento dei linguaggi stessi, vale a dire algoritmi che che per un dato $L \subseteq \Sigma^*$ stabiliscono se una stringa $x \in \Sigma^*$ appartiene a L o meno

definizione generica di ASF (NON C'ENTRA CON LE GRAMMATICHE)

GRAMMATICHE DI CHOMSKY

La grammatica è un formalismo che permette di definire un insieme di stringhe mediante l'imposizione di un particolare metodo per la loro costruzione

Deriva dallo studio di Noam Chomsky per la costruzione di frasi di lingua inglese

DEFINIZIONE

Una grammatica formale G è una quadrupla

$$\boxed{G = \langle V_T, V_N, P, S \rangle}$$

V_T = è l'insieme finito non vuoto di simboli dell'alfabeto terminale i cui elementi sono detti caratteri terminali

V_N = è l'insieme finito non vuoto di simboli dell'alfabeto non terminale i cui elementi sono detti caratteri non terminali o variabili o categorie sintattiche

P = relazione binaria di cardinalità finita su

$$(V_T \cup V_N)^* \circ V_N \circ (V_T \cup V_N)^* \times (V_T \cup V_N)^*$$

P è detto insieme delle regole di produzione o delle regole sintattiche

UNA COPPIA $<\alpha, \beta> \in P$, indichiamo con $\alpha \to \beta$ la relazione tra α e β

$S \in V_N$ = è detto assioma ed è il simbolo non terminale di inizio ossia la categoria sintattica più generale

NOTAZIONI

$$V = V_T \cup V_N \Rightarrow [V^* \circ V_N \circ V^*] \times V^*$$

$a, b, c \in V_T$

$A, B, X, Y \in V_N$

$x, y, z, w \in V_T^*$

$\alpha, \beta, \gamma \in (V_T \cup V_N)^* = V^*$

V_T^* l'insieme di tutte le stringhe che possono essere formate usando i simboli presenti nell'alfabeto V_T

es.

se $V_T = \{a, b\}$ $V_T^* = \{\varepsilon, a, b, aa, ab, bb \dots\}$

OSS.

Le produzioni di una grammatica rappresentano le regole mediante le quali una stringa composta tutta di caratteri terminali può essere trasformata (RISCRITTA) in un'altra

OSS

Il linguaggio generato dalle grammatiche è l'insieme delle stringhe costituite solo dai caratteri terminali ai quali si può arrivare partendo dall'assioma S e applicando una sequenza, arbitrariamente lunga di passi di riscrittura.

Una grammatica è data da un insieme di regole che applicate generano tutte le stringhe del linguaggio (metodo enumerativo)