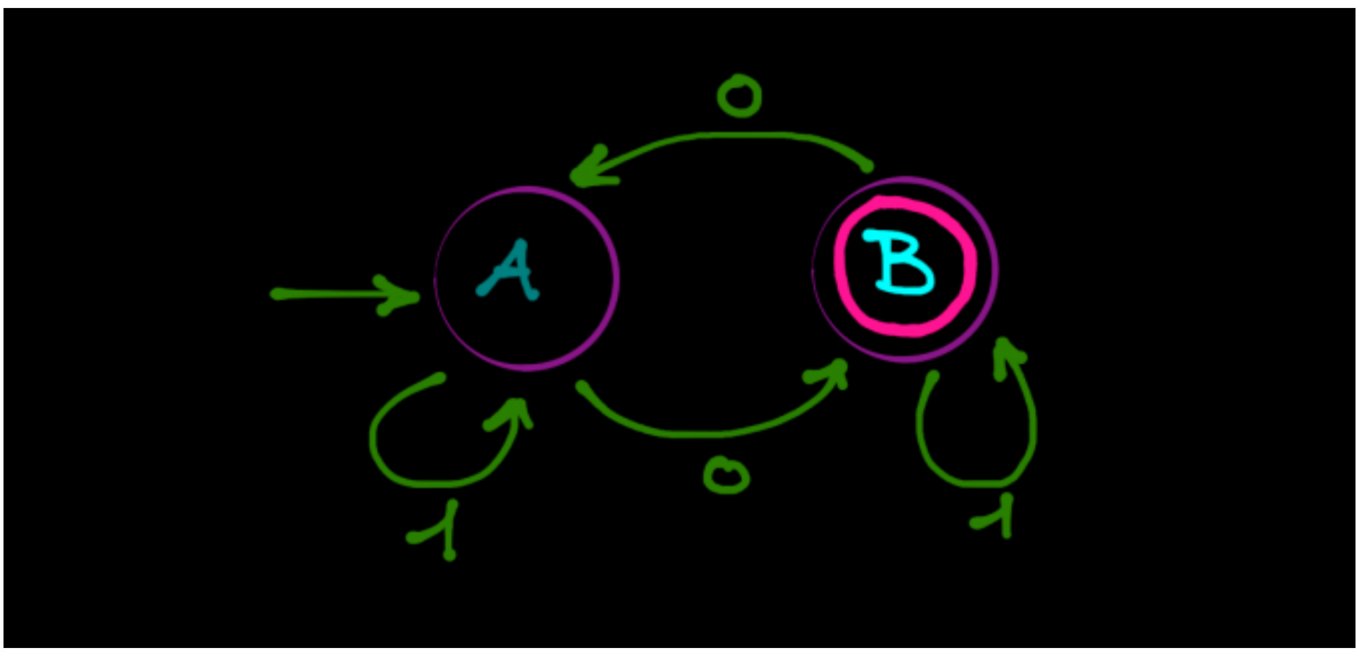
IFIF

APPUNTI DI INFORMATICA, NEWS E ROBE NERD



Esercizio svolto pumping lemma

di Magnifico Amministratore il 13 Novembre 2019 in Programmazione

In questo articolo abbiamo introdotto il concetto del pumping lemma con un esempio semplice, adesso ne proponiamo uno più complesso.

Definiamo il linguaggio L e l'alfabeto Λ come segue

Λ = {a,b,c}

L = $\{a^n\, b^m\, c^k \mid n,m > 0 \wedge k > n + m\}$

Intuitivamente il linguaggio ci dice che il numero delle *c* deve essere maggiore strettamente della somma del numero delle *a* con il numero delle *b* e che ogni lettera dell'alfabeto Λ deve comparire almeno una volta, inoltre, ma dovrebbe essere ovvio, le lettere devono essere disposte nell'ordine in cui sono presentate.

Il linguaggio non è regolare perché l'automa, essendo una macchina con memoria limitata, non riuscirebbe a ricordarsi il numero delle *a* e delle *b* incontrate per confrontarle ogni volta con il numero delle *c* che arrivano in input.

Proseguiamo con l'esercizio seguendo l'enunciato del P.L. che lo trovate qui.

Possiamo dimostrare che il linguaggio non è regolare dicendo che prendendo un qualunque valore $p \in \mathbb{N}^+$ possiamo prendere una stringa *w* di lunghezza maggiore o uguale a *p* sul linguaggio *L* e dividerla in tre parti *xyz* con $|xy| \leq p \wedge y \neq \varepsilon$, ovvero che la lunghezza della parte *xy* deve essere minore o uguale al valore di *p* e che la parte *y* non deve essere vuota. Evinceremo che esiste almeno un valore di $i \in \mathbb{N}$ tale che xy^iz non appartiene al linguaggio.

Scriviamo i passaggi per rendere il tutto più chiaro:

Prendiamo una stringa *w* su *L*: $a^p\, b^p\, c^{2p+1} \in L$

quindi abbiamo preso una stringa con un numero di *a* e *b* uguali e poi con un numero di *c* pari al doppo più 1 della lunghezza di *a* o *b.* Inoltre il valore della lunghezza della stringa *w* è sicuramente più grande di *p* infatti è esattamente p+p+2p+1.

Dividiamola in *xyz*:

x: a^s con $0 \leq s < p$

y: a^t con $0 < t \leq p-s$

z: $a^e\, b^p\, c^{2p+1}$ con $e = p - s - t$

questa suddivisione va bene perché rispetta le condizioni del PL e copre tutte le possibili suddivisioni di quella stringa che abbiamo preso. Infatti *s* è [illegible] e nullo o minore strettamente di *p* e poi *t* è un valore strettamente maggiore di 0 (ricordando: $y \neq \varepsilon$) e minore o uguale alla differenza tra *p* e *s*. Infine *e* rappresenta un valore nel caso la suddivisione presa non prevedesse l'inserimento di tutte le *a* nella stringa *xy*.

Richiede una attimo di concentrazione per comprendere la prima volta ma non è troppo difficile.

Adesso ci resta da far vedere che allora esiste una suddivisione xy^iz con $i \in \mathbb{N}$ tale che la stringa *w* non appartiene al linguaggio.

Il primo valore è i=2, infatti,

$xy^2z = a^s\ a^t\ a^t\ a^e\ b^p\ c^{2p+1} \notin L$

dato che *t* è un numero maggiore di 0 allora la lunghezza totale delle *a* e delle *b* è maggiore della lunghezza generata dalle *c*.

2.6

Article Rating

Hai trovato questo articolo utile?

Vota in anonimato, non devi iscriverti!

Sì 🙂

NO 🙁