ROR(L)

* DFA con stesso stato iniziale
* D’ = D
  + Legge input
  + (Hanno stesso alfabeto)
  + Avanza di tanti simboli quanti la posizione attuale rispetto alla fine
  + Scambia i valori
  + Avanza al successivo input
* Questo per tutti i simboli

Data la tupla di un DFA (quintupla):

con Q = stati, = alfabeto, = funz. di transizione, = stato iniziale, = insieme di stati finali

Funzione di transizione:

Lo stato finale viene raggiunto quando hai scambiato tutti i simboli

Estesa:

* stesso stato iniziale 🡪

Continuando:

* stesso stato iniziale 🡪

Classica funzione di transizione DFA:

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

DROPOUT = linguaggio regolare che toglie un simbolo ogni volta dalla stringa.

* Stesso stato iniziale
* Funzione di transizione
  + Aggiunge uno stato “pozzo” oppure uno stato “scorciatoia” che ad ogni input “salta” un simbolo
  + quando attraversa lo stato corretto
* Stesso stato finale

Siccome per chiusura dei linguaggi regolari il tuo automa può esistere, il linguaggio è regolare.

Riferimento logico: <https://brainly.com/question/38483265>

Immagine che contiene testo, schermata, Carattere, algebra

Descrizione generata automaticamente

Immagine che contiene testo, Carattere, schermata, documento

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, algebra

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamenteMostra che L è regolare: <https://cs.stackexchange.com/questions/1331/how-to-prove-a-language-is-regular>

Se con simboli:

* + dove (sigma) è la transizione che stai compiendo coi simboli
  + (il fatto di aggiungere “a” ad “aa”, o formalizzato o spiegato a parole)
  + da i=1, vale per i=2, i=3, … i=n

Se con parole:

* Come sopra, ma spieghi dicendo a parole tutto
* Balbetti aggiungendo una “a” ad ogni “a” già presente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamenteEsempio TM:

Normalmente, seguo questa logica:

* Parto dall’inizio con una TM a nastro singolo
  + Devo simulare con la TM a nastro singolo la variante

Il verso è dalla TM a nastro singolo alla TM variante

* Parto dalla funzione di transizione
* Svolgo tutti gli stati, sapendo che la TM a nastro singolo si muove a e a (a sx e a dx)

Immagine che contiene testo, Carattere, schermata, numero

Descrizione generata automaticamente

Struttura:



Quando mi sposto devo considerare di non andare oltre ai limiti (cancelletti/inizio/fine) e posso usare nuovi simboli (uno qualsiasi) per “far capire” alla TM (che non ha RAM o altro) dove si trovava e andare così avanti, descrivendo a parole quello che vogliamo farle fare.

Si conclude dicendo:

* se accetta, allora accetta (stato di accettazione raggiunto)
* altrimenti rifiuta

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Decidibilità 🡪 Usa un qualcosa di decidibile per descrivere L 🡪 normalmente, una TM

L’esercizio dice: hai due DFA, entrambi devono accettare la stringa.

Quindi 🡪 uso due TM (normalmente, quando scrivo il linguaggio, mi basta dire

1. accettano entrambe il linguaggio 🡪
2. Usiamo la descrizione di un oggetto decidibile per dimostrare che il mio linguaggio è decidibile 🡪 lo esprimo con una TM

* legge
* Scrive simboli partendo dallo stato iniziale simulando il DFA
* Arriva allo stato finale terminando l’input
* Se accetta, allora accetta
* Altrimenti rifiuta

= l’unione è un’operazione regolare (si dice che è chiusa)

Simile per

* legge
* Scrive simboli partendo dallo stato iniziale simulando il DFA
* Arriva allo stato finale terminando l’input
* Se accetta, allora accetta
* Altrimenti rifiuta

Se entrambi sono arrivati alla fine, entrambi i DFA sottostanti hanno accettato la stringa

(Si chiama = dare una descrizione a livello implementativo / ad alto livello di ciò che fa la macchina)

Essendo che L è decidibile, allora avendo usato una TM che termina (DFA termina sempre), allora il linguaggio è decidibile.