# Automa DFA

Un Automa a Stati Finiti Deterministico (DFA) è una quintupla A = (Q, Σ, δ, q0, F), dove:

* Q è un insieme finito di stati
* Σ è un alfabeto finito (= simboli in input)
* δ è una funzione di transizione (q, a) → q’
* q0 ∈ Q è lo stato iniziale
* Immagine che contiene testo, Carattere, schermata, linea

  Descrizione generata automaticamenteF ⊆ Q è un insieme di stati finali

Esso è determinato da un inizio ed una fine.

Gli stati possono sempre essere determinati da una tabella di transizione che per ogni stato o coppia di stati determina una certa direzione.

# Automa NFA

Un Automa a Stati Finiti Non Deterministico (NFA) è una quintupla A = (Q, Σ, δ, q0, F), dove:

* Q è un insieme finito di stati
* Σ è un alfabeto finito (= simboli in input)
* δ è una funzione di transizione che prende in input (q, a) e restituisce un sottoinsieme di Q
* q0 ∈ Q è lo stato iniziale
* F ⊆ Q è un insieme di stati finali

Esso si trova contemporaneamente in più stati diversi

# Equivalenza NFA-DFA

* Dato un NFA costruisco un DFA con gli stessi stati e uno stato finale
* Ne “espando” le transizioni fino a coprire definitamente tutti i casi con una tabella di transizione

# Automi epsilon-NFA

Essi hanno le stesse caratteristiche dei precedenti, ma non consumano transizioni grazie agli stati . Queste sono definite come -transizioni ed esiste la -chiusura.

# Linguaggi regolari

Sono definiti dai DFA e prevedono l’utilizzo di una serie di operazioni che sono *chiuse* e deterministicamente eseguibili.

Possono essere definite anche *da un insieme finito di stringhe* (espressioni regolari)

# Linguaggi non regolari e Pumping Lemma

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamenteDato un automa, esiste almeno uno stato che si ripete.

Immagine che contiene testo, Carattere, schermata, algebra

Descrizione generata automaticamente

Tricks:

* Provare con lunghezza 0 centrale
* Basta che una delle due parti di stringa sia sbilanciata

# Provare che un linguaggio sia regolare

<https://cs.stackexchange.com/questions/1331/how-to-prove-a-language-is-regular>

# Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamenteGrammatiche context-free

Una grammatica genera stringhe attraverso la derivazione:

* si scrive la variabile iniziale
* si trova una variabile che è stata scritta e una regola che inizia con quella variabile. Si sostituisce la variabile con il lato destro della regola
* Immagine che contiene testo, schermata, Carattere

  Descrizione generata automaticamentesi ripete (fino a quando non ci sono più variabili)

Grammatiche ambigue = Possibile qualsiasi derivazione

# Forma normale di Chomsky

Immagine che contiene testo, Carattere, schermata, ricevuta

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

(Es. Automi semplici pp. 52 – 55)

# Linguaggi non context-free

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

# Macchine di Turing