*Scritti da Gabriel: gli appunti comprendono anche le domande a quiz interattive, alla fine di ogni riassunto della lezione*

*02/03/2022: Introduzione: automi a stati finiti deterministici e concetti base*

La domanda fondamentale di inizio corso è determinare cosa effettivamente possa o non possa fare un calcolatore, un automa o altro, capendo la struttura di un certo problema e quindi risolverlo possibilmente. Per descrivere un *problema* noi dobbiamo descrivere i possibili input, i possibili output o risultati che coi aspettiamo e la relazione tra di essi.

A tale scopo sappiamo di avere un certo *algoritmo*, quindi una procedura tale a risolvere un dato problema, sulla base di un certo tipo di calcoli e computazioni.

Analogamente, importante valutare correttamente la *complessità* dell’algoritmo, quindi che l’algoritmo risolva effettivamente il problema presente, valutandone poi la *complessità spaziale/temporale.*

Allo stesso modo, poter risolvere un problema riguarda l’astrazione del problema esprimendolo sotto forma di *linguaggio*, visto come *insieme di stringhe*. Le soluzioni trasformano quindi le linee di input con linee di output. L’esempio di problema può essere l’insieme dei numeri primi.

Tutti i processi computazionali possono essere ridotti ad uno tra la determinazione di *appartenenza* ad un insieme, eseguendo una *mappatura* tra insiemi di stringhe.

A questo punto introduciamo il concetto di *automa*, dispositivo matematico astratto in grado di determinare l’appartenenza di una stringa ad un insieme di stringhe e in grado di trasformare una stringa in un’altra stringa. Esso ha tutti gli aspetti di un computer, avendo I/O, una memoria, è in grado di prendere decisioni e può trasformare l’input in output.

Descriviamo in particolare l’aspetto della *memoria*, distinguendo tra *memoria finita ed infinita* e, naturalmente, anche il tipo di accesso alla memoria, distinguendo tra *limitato e illimitato*.

Parliamo ora di *automi a stati finiti*, che sono il modello computazionale più semplice e dispongono di una quantità di memoria *finita*. Gli automi sono usati come *modello* per ricerche di parole chiave, analizzatori lessicali, progettazione di circuiti digitali e scopi similari.

Un esempio semplice è la porta automatica, capedo tramite un sensore se ci sia o meno una persona rilevando la presenza di una persona. In questo caso gli stati sono: *chiusa* o *aperta*.

Quattro possibili input: *fronte/retro/ambo/nessuna.*

In maniera più precisa, possiamo definire alcuni concetti, come il concetto di *alfabeto*, che è un insieme finito e non vuoto di simboli. Il simbolo che lo identifica è il *∑*, quello di serie/sommatoria (ad esempio *∑ = {0,1},* alfabeto binario, oppure *∑* = {a, b, c, d…. z} insieme delle lettere minuscole, insieme dei caratteri ASCII, ecc.).

Definiamo anche una *stringa*, che sarebbe niente altro che una sequenza finita di simboli da un alfabeto. La *stringa vuota* sarà definita con *ε,* contenuto in *∑0.* La stringa è composta da un certo numero di simboli e questa è la *lunghezza di una stringa*, definita come|w|.

Vi sono poi le *potenze di un alfabeto*, Σk = insieme delle stringhe di lunghezza k con simboli da Σ.

Ad esempio: Σ = {0, 1}

Σ 0 = {ε} Σ 1 = {0, 1} Σ 2 = {00, 01, 10, 11}

L’insieme di tutte le stringhe su Σ è denotato da Σ\* (unione di tutte le stringhe matematicamente parlando) e più in generale, dato un alfabeto, un *linguaggio* è ogni sottoinsieme L ⊆ Σ∗.

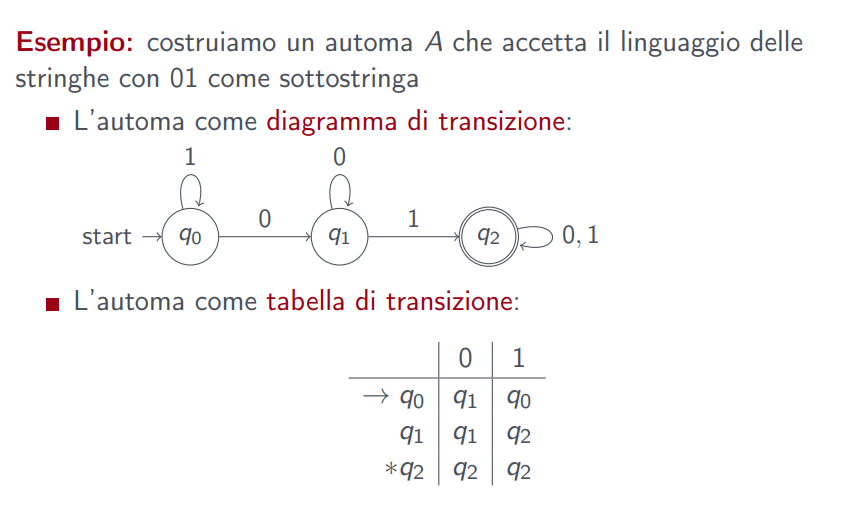
Il *linguaggio vuoto* non contiene nessuna parola ed è definito dal simbolo ∅; si ricordi che il linguaggio vuoto è diverso da una parola vuota.

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*
* F ⊆ Q è un insieme di *stati finali*

Una *funzione di transizione* è una funzione con dominio finito, prendendo come finiti l’alfabeto di input e l’insieme degli stati che l’automa può assumere viene rappresentato in forma tabellare come *tabella di transizione* oppure sotto forma di grafo orientato, i cui archi rappresentano le transizioni tra uno stato e l’altro; qui si parla di *diagramma di transizione*.

Esempio pratico:



Data una parola w = w1w2 . . .wn, la computazione dell’automa *A* con input *w* è una sequenza di stati r0,r1 ..r*n*che rispetta due condizioni:

1. r0 = q0 (inizia dallo stato iniziale)
2. δ(ri ,wi+1) = ri+1 per ogni i = 0, . . . , n − 1 (rispetta la funzione di transizione)

Diciamo che la computazione accetta la parola w se: rn ≡ F (la computazione termina in uno stato finale).

Un DFA *A* accetta la parola *w* se la computazione accetta *w*. Formalmente, il linguaggio accettato da A è L(A) = {w ∈ Σ∗ | A accetta w}. I linguaggi accettati da automi a stati finiti sono detti *linguaggi regolari*

Domande interattive della lezione:

* Quali sono gli input della funzione di transizione di un DFA? 🡪 Uno stato e un simbolo
* Un DFA deve avere un solo stato finale? 🡪 Falso
* Qual è l’input del problema “È un numero primo?” 🡪 {0,1,2,3,4,5..}
* Quale insieme di stringhe rappresenta il problema nel caso “È un numero primo?” 🡪 {2,3,5,7,11,13..}
* Per ogni elemento, stabilire se è carattere, parola, linguaggio:

1. a 🡪 carattere
2. abracadabra 🡪 parola
3. {abracadabra, apostrofo, a} 🡪 linguaggio di tre parole
4. {abracadabra} 🡪 linguaggio composto da una parola

* Dato l’alfabeto ∑ = {0,1} quante sono le stringhe appartenenti al linguaggio? 🡪 16
* Quale dei seguenti linguaggi sull’alfabeto {0,1} contiene un numero infinito di stringhe? 🡪 Tutte le stringhe che iniziano con 1
* Dato l’alfabeto ∑ = {0,1} quante sono le stringhe contenute in Σ 0? 🡪 ε
* Quante stringhe ci sono nel linguaggio sull’alfabeto {0,1} di tutte le stringhe di lunghezza *n*? 🡪 2n
* Un DFA possiede una quantità di memoria molto limitata 🡪 Vero
* L’insieme di tutti gli stati di un automa si indica con 🡪 Q
* Quali sono gli input della funzione di transizione di un DFA? 🡪 Uno stato e un simbolo
* Un DFA deve avere un solo stato finale 🡪 Falso