

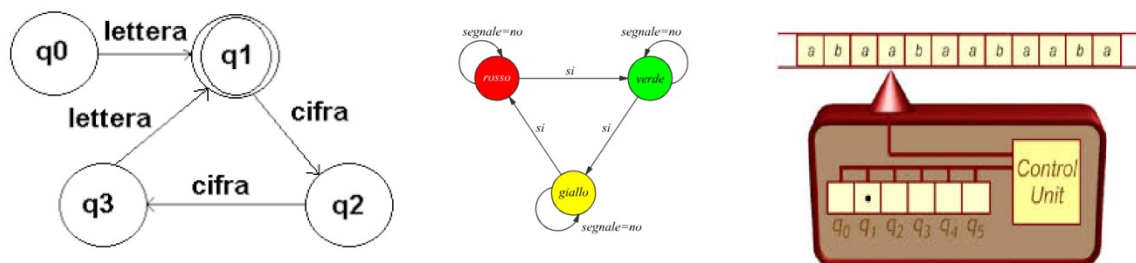
ITIS-LS “Francesco Giordani” Caserta

prof. Ennio Ranucci

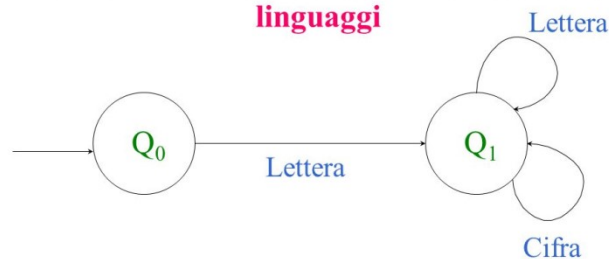
a.s. 2019-2020

*Semplice automa riconoscitore della sequenza 010*

*Esercitazioni Javascript – Linguaggi formali*



**Automi a stati finiti come riconoscitori di linguaggi**



→ Identificatori del C++

primo	secondo	Ind1	VarMia
nome_23	_temp	S23I3	Inizio_File

**ITIS-LS "Francesco Giordani" Caserta**  
**Anno scolastico: 2019/2020**  
**Classe 3<sup>A</sup> sez.B spec. Informatica e telecomunicazioni**

**Data:**

**Numero progressivo dell'esercizio: es1**

**Versione: 1.0**

**Programmatore/i:**

**Sistema Operativo: Windows 10**

**Compilatore/Interprete: Browser - Javascript**

**Obiettivo didattico:**

Automati riconoscitori

**Obiettivo del programma:**

Realizzare l'automata che riconosce la sequenza 010

Gli automati riconoscitori sono in grado di riconoscere la presenza di una particolare sequenza (o stringa) di simboli tra quelli che riceve in ingresso.

Consideriamo un automa riconoscitore che rileva la presenza della sequenza 010 tra tutti i simboli 0 e 1 che si avvicendano in ingresso.

Sequenza: 010

Insiemi caratteristici dell'automata

$I=\{0,1\}$

$U=\{0,R\}$

0= non è stata riconosciuta alcuna sequenza 010

R= è stata riconosciuta la sequenza 010

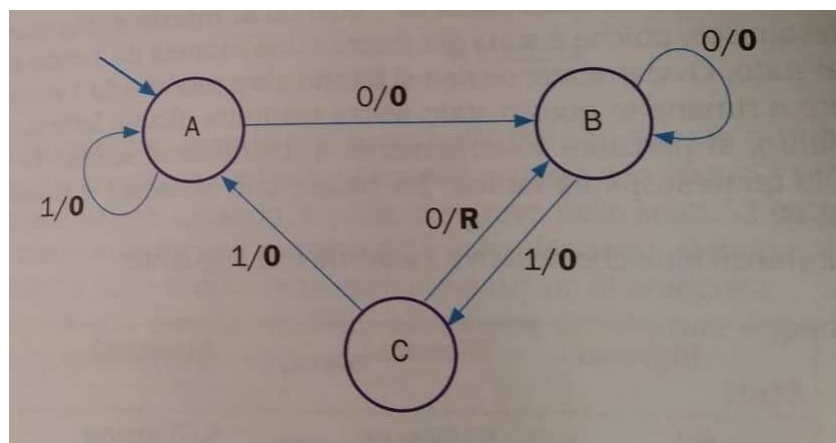
$S= \{A, B, C\}$

A = (Stato iniziale) si rimane in questo stato finché esce 1

B= è stato riconosciuto il primo zero della sequenza

C=è stata riconosciuta la sottosequenza 01

Diagramma degli stati

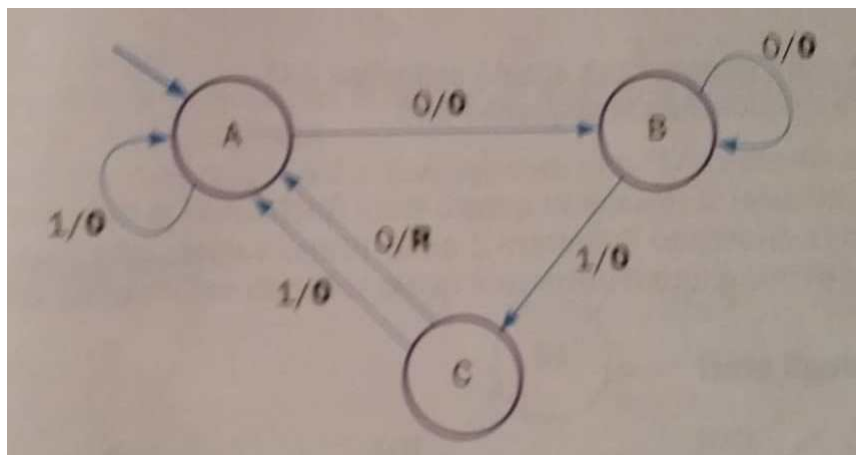


Consideriamo di avere i seguenti valori in ingresso: 01010. I primi tre valori individuano la sequenza cercata (010), per cui al terzo istante l'uscita va a R (sequenza trovata), ma l'ultimo 0 della sequenza valida potrebbe essere utilizzato anche come primo elemento di una nuova sequenza, nel qual caso l'uscita tornerebbe a 1 nel quinto intervallo di tempo. Se, invece, non si vuole considerare utile quell valore, l'uscita dovrebbe rimanere a 0. Nel primo caso si dice che il riconoscitore ammette ripetizioni o sovrapposizioni, nel secondo caso che non le ammette.

L'automa che stiamo considerando ammette ripetizioni, quindi le uscite andrebbero due volte a R (sequenza riconosciuta) con gli ingressi 01010.

Il diagramma che segue, invece, è quello relativo al riconoscitore che non ammette ripetizione. Con gli ingressi 01010 le uscite andrebbero a R (sequenza riconosciuta) solo una volta.

Automa riconoscitore della sequenza 010 che non ammette ripetizioni:



```

<html>
<head>
  <SCRIPT Language="JavaScript">
    function Aggiungi0()
      {
        document.F1.T1.value = document.F1.T1.value + "0"
      } //fine Aggiungi0
    function Aggiungi1()
      {
        document.F1.T1.value = document.F1.T1.value + "1"
      } //fine Aggiungi1
    function CancellaUnCarattere()
      {
        Stringa = document.F1.T1.value
        document.F1.T1.value = Stringa.substring(0, Stringa.length-1)
      } //fine CancellaUnCarattere
    function Riconosci()
      {
        var Stato = "A"
        document.F1.RIS.value = ""
        var Str=document.F1.T1.value
        for(i=0;i<Str.length;i++)
          {
  
```

```

var Car=Str.substring(i,i+1)
    alert("Ho letto il carattere "+ Car +" mi trovo nello stato "+ Stato)
if (Car == "0" && Stato == "A")
    Stato = "B"
else
    if (Car == "1" && Stato == "B")
        Stato = "C"
    else
        if (Car == "1" && Stato == "C")
            Stato = "A"
        else
            if (Car == "0" && Stato == "C")
                {
                    Stato = "A"
                    document.F1.RIS.value = "SEQUENZA RICONOSCIUTA"
                    alert("Sequenza riconosciuta")
                    break // blocca l'esecuzione del for appena riconosce 010
                }
            alert("Sono passato allo Stato: "+ Stato)
        } //fine for
    if(i == Str.length)
        {
            document.F1.RIS.value = "SEQUENZA NON RICONOSCIUTA"
            alert("Sequenza non riconosciuta")
        }
    } //fine Riconosci
</SCRIPT>
</head>
<body>
<H2> AUTOMA RICONOSCITORE della sequenza 010 </H2>
<FORM NAME="F1">
    Inserisci la successione di 0 e 1 <br><br>
    <INPUT TYPE ="TEXT" name="T1" READONLY>
    <INPUT TYPE ="BUTTON" onclick="Aggiungi0()" value="Aggiungi 0">
    <INPUT TYPE ="BUTTON" onclick="Aggiungi1()" value="Aggiungi 1">
    <INPUT TYPE ="BUTTON" onclick="CancellaUnCarattere()"
        value="Cancella un carattere"> <br><br>
    <INPUT TYPE ="BUTTON" onclick="Riconosci()" value="Riconosci sequenza">
    <INPUT TYPE ="TEXT" name="RIS" size=40 value="In fase di riconoscimento...">
</FORM>
</body>
</html>

```

*Gli automi sono spesso utilizzati per descrivere linguaggi formali in informatica teorica, e per questo sono chiamati accettori o riconoscitori di un linguaggio.*

*L'insieme dei possibili simboli che possono essere forniti ad un automa costituisce il suo alfabeto.*

*Una sequenza di simboli (detto anche stringa o parola) appartiene al linguaggio se essa viene accettata dal corrispondente automa, ovvero se porta l'automata in uno stato valido, che sia lo stesso o un altro stato. Un sottoinsieme del linguaggio riconosciuto, chiamato linguaggio marcato porta l'automata dal suo stato iniziale ad uno stato finale o marcato.*

*A diverse classi di automi corrispondono diverse classi di linguaggi, caratterizzate da diversi livelli di complessità.*

*Un automa può quindi riconoscere più linguaggi (produzione di più sequenze).*

*Gli automi a stati finiti sono dotati di un insieme finito di stati, scandiscono una stringa di simboli in ingresso (simbolo per simbolo) in maniera ordinata per decidere se essa appartenga o meno ad un linguaggio.*

*Formalmente tali automi sono delle quintuple,  $(Q, I, f, q_0, F)$ , formate da un alfabeto finito dei simboli in ingresso ( $I$ ), un insieme finito di stati ( $Q$ ) tra cui si distingue uno stato iniziale ( $q_0$ ) ed un sottoinsieme di stati, detti finali ( $F$ ), ed una funzione di transizione ( $f$ ). Tale funzione, descritta mediante una tabella di transizione degli stati, o un multidigrafo, è definita per coppie (stato corrente, simbolo scandito) e stabilisce la transizione da compiere, ossia lo stato in cui si transita leggendo il dato simbolo.*

*Il funzionamento dell'automata può essere così descritto:*

*partendo dallo stato iniziale e dal primo simbolo della stringa in ingresso si decide in base alla funzione di transitare in un determinato stato (potrebbe anche essere lo stesso stato);*

*finché esiste un altro simbolo nella stringa da scandire si opera alla stessa maniera fino ad esaurire la stringa in ingresso;*

*la stringa si dirà accettata se si giunge in uno stato appartenente al sottoinsieme degli stati finali.*

*Tali automi sono in grado di riconoscere i linguaggi regolari.*

*I linguaggi regolari (grammatiche di tipo 3) sono esattamente quelli riconosciuti dagli automi a stati finiti.*

*Gli automi a stati finiti risultano quindi essere i riconoscitori dei sistemi generativi che sono le grammatiche di tipo 3.*