



ELEMENTI DI INFORMATICA

DOCENTE: FRANCESCO MARRA

INGEGNERIA CHIMICA

INGEGNERIA ELETTRICA

SCIENZE ED INGEGNERIA DEI MATERIALI

INGEGNERIA GESTIONALE DELLA LOGISTICA E DELLA PRODUZIONE

INGEGNERIA NAVALE


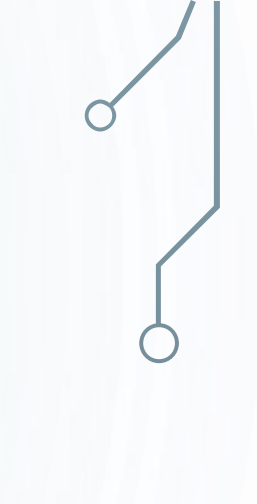
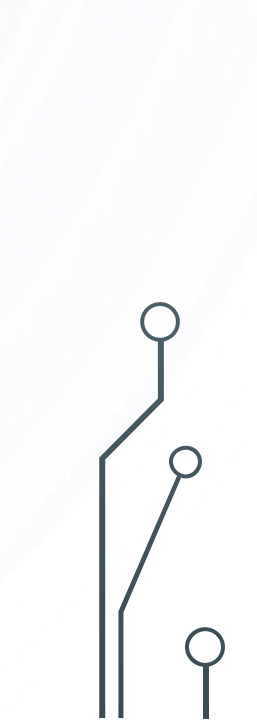
UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DAI PROBLEMI ALLA MACCHINA DI TURING





AGENDA

- Problemi, algoritmi ed esecutori
 - Automi a stati finiti
 - Macchina di Turing
- 
- 
- 

DEFINIZIONE DI PROBLEMA

- Uno degli scopi principali dell'informatica è risolvere problemi con i calcolatori
- Un problema è una classe di domande omogenee alle quali è possibile dare una risposta attraverso un metodo o una procedura uniforme di risoluzione
 - ad esempio, qual è il la somma di due numeri X e Y ?
- Ogni specifica domanda della classe si chiama istanza del problema
 - ad esempio, qual è il la somma di 4 e 7?

DEFINIZIONE DI PROBLEMA

- Un problema opera tipicamente su uno o più termini variabili
 - tali termini si chiamano variabili/dati di ingresso o di input
 - per ogni possibile combinazione di valori delle variabili di ingresso, si genera una diversa istanza del problema
- Ad esempio, qual è il la somma di due numeri X e Y ?
 - X e Y sono le variabili di ingresso del problema
 - se $X=4$ e $Y=7$, si genera l'istanza «qual è il la somma di 4 e 7?»
 - se $X=3$ e $Y=5$, si genera l'istanza «qual è il la somma di 3 e 5?»
 - etc.

DEFINIZIONE DI PROBLEMA

- Il risultato (output) di un problema è tipicamente costituito da uno o più termini variabili
 - tali termini si chiamano variabili/dati di uscita o output
 - le variabili di output dipendono dai dati di ingresso della particolare istanza del problema
- Ad esempio, qual è la somma Z di due numeri X e Y ?
 - X e Y sono le variabili di ingresso, Z è la variabile di uscita del problema
 - se $X=4$ e $Y=7$, si genera l'istanza «qual è la somma di 4 e 7?» $\rightarrow Z = 11$
 - se $X=3$ e $Y=5$, si genera l'istanza «qual è la somma di 3 e 5?» $\rightarrow Z = 8$
 - etc.

ESEMPI DI PROBLEMA

- Preparare una torta alla frutta
 - è noto il risultato
 - non si riesce a ricavare alcuna indicazione sulla ricetta da seguire
 - la ricetta non è di facile individuazione in un libro di cucina per la sua formulazione generica
- Risolvere le equazioni di secondo grado
 - Problema di analisi matematica di cui si conosce chiaramente il procedimento risolvante
- Individuare il massimo tra tre numeri
 - problema impreciso ed ambiguo
 - non specifica se la variabile di uscita dev'essere il valore numerico del massimo o la posizione in cui si trova il massimo tra i numeri assegnati

ESEMPI DI PROBLEMA

- **Inviare un invito ad un insieme di amici**
 - diverse possibili soluzioni per inviare l'invito (posta ordinaria, SMS, posta elettronica, WhatsApp, etc.)
 - bisogna scegliere la soluzione più conveniente
 - Ad es. si può scegliere la soluzione che presenta un costo più basso e più efficiente
- **Individuare le tracce del passaggio di extraterrestri**
 - problema che non ammette soluzione o non risolvibile

OSSERVAZIONI

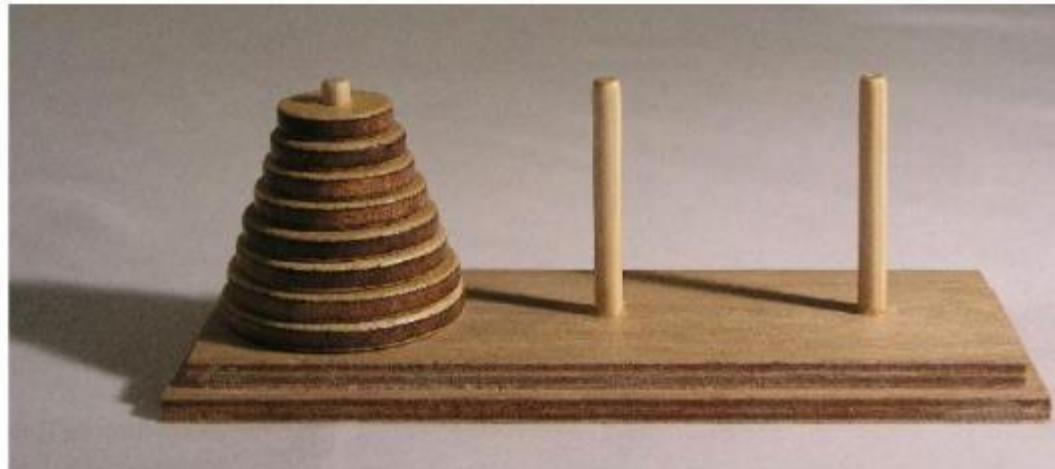
- La descrizione del problema non fornisce, in generale, indicazioni sul metodo risolutivo
 - In alcuni casi può presentare **imprecisioni e ambiguità** che possono portare a **soluzioni errate**
- Per alcuni problemi non esiste una soluzione
 - in tal caso, si parla di problemi **non calcolabili**
 - la teoria della **calcolabilità** valuta se un problema può essere svolto o meno in un procedimento automatico, ovvero se è risolvibile mediante un calcolatore

OSSERVAZIONI

- Alcuni problemi hanno più soluzioni possibili
 - in tal caso, si calcola quella più vantaggiosa sulla base di un insieme di parametri prefissati:
 - costo della soluzione, tempi di attuazione, risorse necessarie alla sua realizzazione, ecc.
- Per alcuni problemi non esistono soluzioni eseguibili in tempi ragionevoli
 - in tal caso, si parla di problemi **non trattabili**
 - la teoria della **trattabilità** valuta la complessità e i costi di esecuzione della soluzione di un problema calcolabile

PROBLEMA CALCOLABILE MA NON TRATTABILE: ESEMPIO

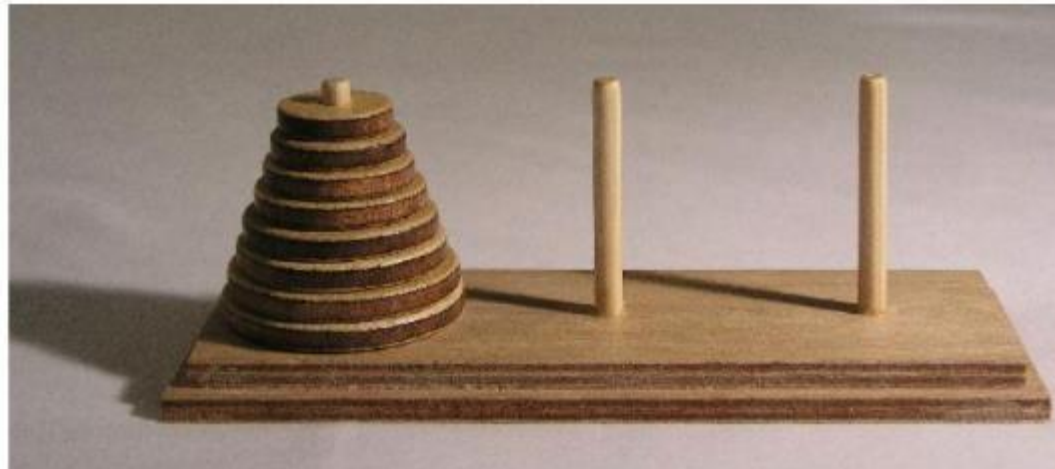
- Torre di Hanoi
 - tre paletti e n dischi di grandezza decrescente
 - l'obiettivo è spostare i dischi da un paletto ad un altro, tenendo conto che un disco può essere appoggiato su un altro disco soltanto se più piccolo di esso



PROBLEMA CALCOLABILE MA NON TRATTABILE: ESEMPIO

- Torre di Hanoi

- sono necessarie almeno $2^n - 1$ mosse per risolverlo
- se $n=64$ e una mossa richiede 1 secondo allora sono necessari $2^{64} - 1$ secondi per risolverlo → 585 miliardi di anni



DEFINIZIONE DI ALGORITMO

- Un algoritmo è una sequenza finita di passi che portano alla realizzazione di un compito
 - un insieme finito di istruzioni che, eseguite secondo un ordine prestabilito, permettono di giungere alla soluzione di un problema
- Aspetti da tenere presente nella definizione
 - insieme finito di passi
 - i passi vanno eseguiti in sequenza
 - i dati di input sono elaborati per giungere alla soluzione del problema
 - i dati di output sono prodotti come soluzione del problema

DEFINIZIONE DI ESECUTORE

- Un algoritmo viene definito presupponendo la disponibilità di un opportuno esecutore che dovrà eseguire le istruzioni descritte
 - un algoritmo dipende sia dal compito che si vuole realizzare, sia dall'esecutore per il quale viene formulato
- L'esecutore di un algoritmo è l'entità che deve realizzare il compito di giungere alla soluzione di un problema, attuando la sequenza di passi che compongono l'algoritmo stesso
 - un esecutore può far fronte al suo compito se e solo se è in grado di comprendere ed eseguire tutti i passi della sequenza

SCHEMA DI SOLUZIONE DI UN PROBLEMA

- **Soluzione di un problema:**

- Informazioni di ingresso (Input)

- Le informazioni da fornire per risolvere il problema

- Algoritmo

- sequenza finita di passi che portano alla realizzazione di un problema/compito

- Esecutore

- L'entità che esegue l'algoritmo
- Deve comprendere le singole istruzioni e deve essere capace di eseguirle

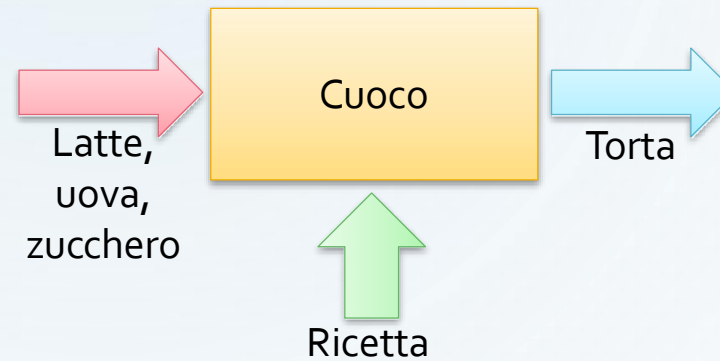
- Informazioni di uscita (Output)

- I risultati prodotti dall'esecutore

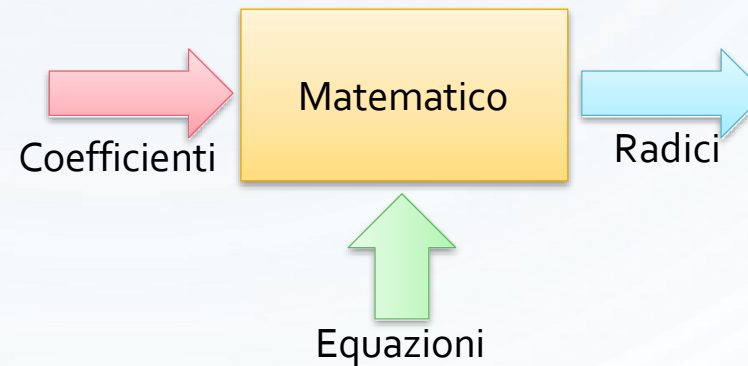


ALGORITMI ED ESECUTORI: ESEMPI

Torta



Soluzioni equazione di
secondo grado



DEFINIZIONE DI PROGRAMMA

- Un **programma** è un algoritmo scritto in un linguaggio comprensibile all'esecutore (linguaggio di programmazione)
- Il **linguaggio** serve a descrivere in modo non ambiguo tutte e sole le operazioni che l'esecutore è in grado di eseguire
 - descrive tutto ciò che occorre sapere dell'esecutore per poter formulare algoritmi
- Si può identificare l'esecutore con il suo linguaggio di programmazione, ignorando il suo funzionamento interno

DEFINIZIONE DI CALCOLATORE

- Un **calcolatore** è un apparecchio elettronico progettato per eseguire autonomamente e velocemente attività diverse
 - non ha nessuna capacità decisionale o discrezionale
 - si limita a compiere determinate azioni secondo procedure prestabilite
- Il **processore** di un calcolatore è un esecutore che interpreta ed esegue le singole istruzioni di un algoritmo scritto in un prefissato linguaggio di programmazione
 - è un **automa**, cioè una macchina che esegue algoritmi
 - a partire da un opportuno insieme di dati iniziali, produce in uscita i risultati dell'esecuzione dell'algoritmo per quei dati iniziali

AUTOMA A STATI FINITI

- Astrazione del concetto di *macchina che esegue algoritmi*
 - basato sul concetto di **stato**, cioè la particolare condizione di funzionamento in cui può trovarsi la macchina
- Applicabile a qualsiasi sistema che evolve nel tempo per effetto di sollecitazioni esterne
 - ogni sistema se soggetto a sollecitazioni in ingresso risponde in funzione della sua situazione attuale eventualmente emettendo dei segnali di uscita
 - l'effetto della sollecitazione in ingresso è il **mutamento dello stato** del sistema stesso
 - il sistema ha sempre uno **stato iniziale** di partenza da cui inizia la sua evoluzione
 - eventualmente, può terminare in uno **stato finale** dopo aver attraversato una serie di stati intermedi

AUTOMA A STATI FINITI: MACCHINA DI MEALY

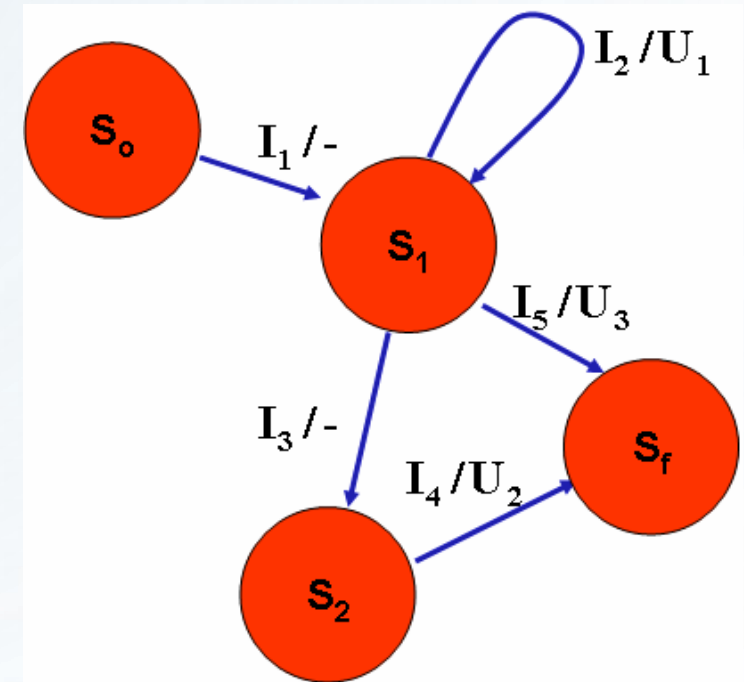
- Una **macchina di Mealy** è un automa a stati finiti i cui valori di uscita sono determinati dallo stato attuale e dall'ingresso corrente
- Può essere definita tramite una quintupla di elementi **(Q, I, U, t, w)**
 - **Q** è un insieme finito di **stati** interni caratterizzanti l'evoluzione della macchina
 - **I** è un insieme finito di sollecitazioni in **ingresso**
 - **U** è un insieme finito di **uscite**
 - **$t: Q \times I \rightarrow Q$** è una funzione di transizione degli stati, che determina lo stato successivo della macchina a partire da uno stato e da un ingresso fissati
 - **$w: Q \times I \rightarrow U$** è una funzione di uscita, che determina l'uscita della macchina a partire da uno stato e da un ingresso fissati

AUTOMA A STATI FINITI: MACCHINA DI MOORE

- Una **macchina di Moore** è un automa a stati finiti i cui valori di uscita sono determinati solo dallo stato attuale
- Può essere definita tramite una quintupla di elementi **(Q, I, U, t, w)**
 - **Q** è un insieme finito di **stati** interni caratterizzanti l'evoluzione della macchina
 - **I** è un insieme finito di sollecitazioni in **ingresso**
 - **U** è un insieme finito di **uscite**
 - **$t: Q \times I \rightarrow Q$** è una funzione di transizione degli stati, che determina lo stato successivo della macchina a partire da uno stato e da un ingresso fissati
 - **$w: Q \rightarrow U$** è una funzione di uscita, che determina l'uscita della macchina a partire da uno stato fissato

AUTOMA A STATI FINITI: RAPPRESENTAZIONE A GRAFO

- **Grafo**
 - un nodo per rappresentare gli stati del sistema
 - archi orientati ad indicare le transizioni
- **Stati intermedi**
 - nodi con archi entranti ed uscenti
- **Stato iniziale**
 - l'unico nodo con nessun arco entrante
- **Stato finale**
 - se esiste, nodo con nessun arco uscente



Rappresentazione grafica di una
macchina di Mealy

AUTOMA A STATI FINITI: RAPPRESENTAZIONE TABELLARE

- Tabella (vale anche il viceversa)
 - tante righe quanti sono gli ingressi
 - tante colonne quanti sono gli stati
- Stati nei quali il sistema transita per effetto delle sollecitazioni in *ingresso* (*funzione t*)
- indicazione dell'eventuale uscita prodotta nella transizione (*funzione w*)

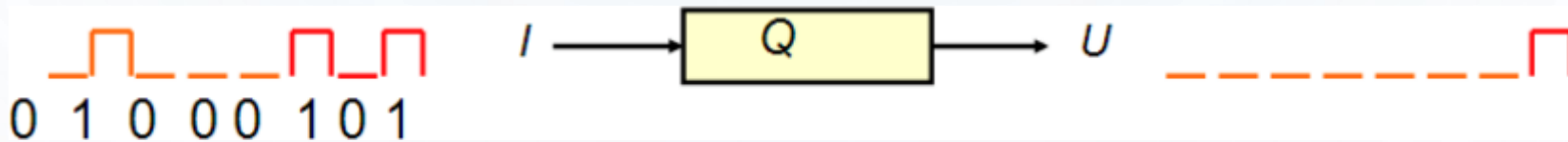
$$\begin{aligned} Q &= (S_0, S_1, S_2, S_f) \\ I &= (I_1, I_2, I_3, I_4, I_5) \\ U &= (U_1, U_2, U_3) \end{aligned}$$

I/S	S ₀	S ₁	S ₂	S _f
I ₁	S ₁ /-			
I ₂		S ₁ /U ₁		
I ₃		S ₂ /-		
I ₄			S _f /U ₂	
I ₅		S _f /U ₃		

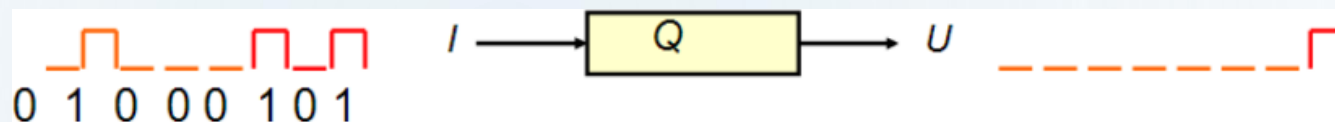
Rappresentazione tabellare di una
macchina di Mealy

ESEMPIO

- Vogliamo realizzare una macchina in grado di riconoscere una sequenza di bit 101 in ingresso
 - la macchina avrà un unico ingresso **I** su cui arriva una sequenza di 1 e 0
 - un'unica uscita **U** che vale 1 quando in ingresso viene riconosciuta la sequenza 101

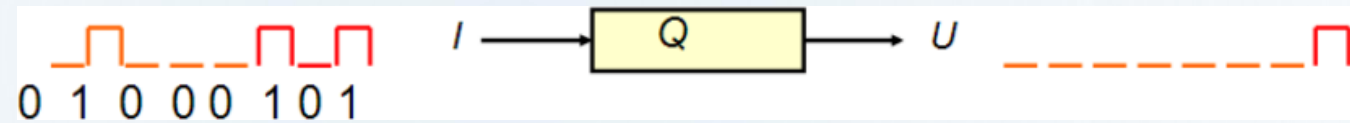


ESEMPIO



- Possiamo usare una macchina a stati finiti con tre stati S_0 , S_1 , S_2 con i seguenti significati:
 - S_0 è lo stato in cui si è in attesa di riconoscere in ingresso il primo bit della sequenza desiderata, ovvero si attende "1" in ingresso
 - S_1 è lo stato in cui ci si trova se è stato ricevuto il primo bit della sequenza desiderata e si attende di riconoscere in ingresso il secondo bit della sequenza, ovvero si è ricevuto "1" e si attende "0"
 - S_2 è lo stato in cui ci si trova se sono stati ricevuti in ingresso sia il primo che il secondo bit della sequenza desiderata e si attende di riconoscere in ingresso l'ultimo bit della sequenza, ovvero è stata ricevuta la sequenza "10" in ingresso e si attende "1". Quando arriva un nuovo bit, in ogni caso, si ritorna in S_0 ma con uscite diverse a seconda che si sia ricevuto "1" o "0"

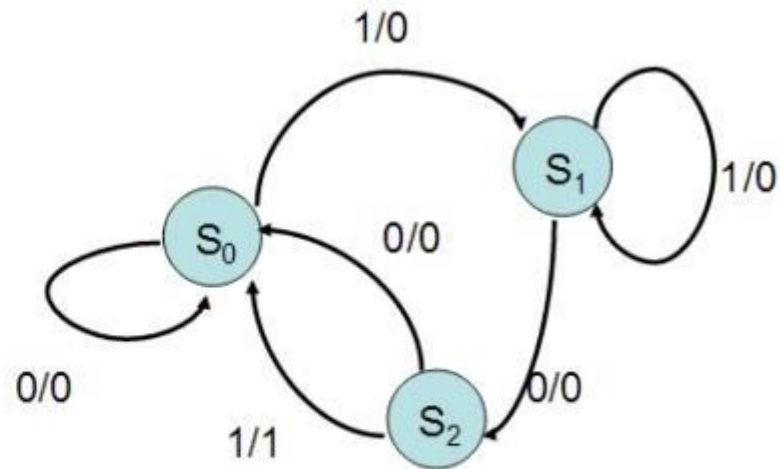
ESEMPIO



- Automa di Mealy in grado di riconoscere la sequenza "101"

	0	1
s_0	$s_0/0$	$s_1/0$
s_1	$s_2/0$	$s_1/0$
s_2	$s_0/0$	$s_0/1$

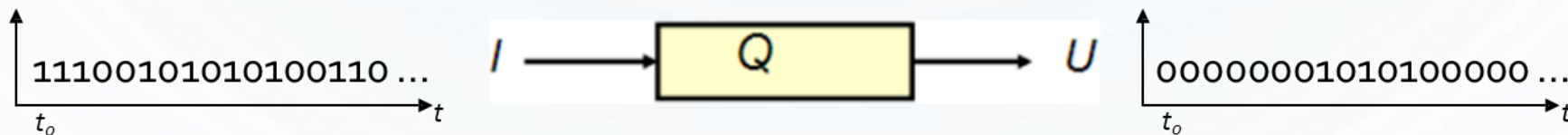
descrizione tramite tabella



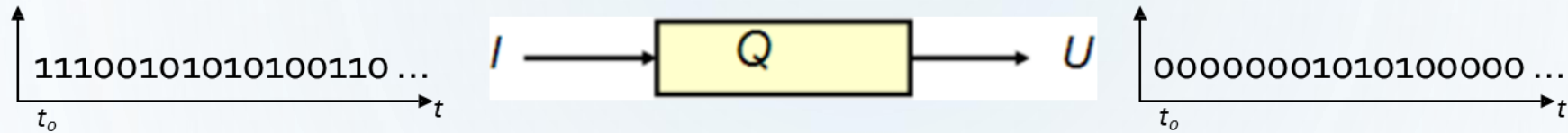
descrizione tramite grafo

ESERCIZIO: SOTTO-SEQUENZE 0101

- Si definisca la tabella e il grafo di transizione degli stati di una macchina di Mealy in grado di riconoscere una sequenza di bit $0_i 1_{i+1} 0_{i+2} 1_{i+3}$ in ingresso, dove b_i rappresenta il bit b ricevuto al tempo $t=i$
 - la macchina avrà un unico ingresso I su cui arriva una sequenza di **1** e **0**
 - un'unica uscita U che vale 1 se gli ultimi quattro bit ricevuti in ingresso formano la sotto sequenza **0101**, in corrispondenza dell'ultimo bit della sotto sequenza riconosciuta
 - la macchina dev'essere in grado di riconoscere anche sotto sequenze corrette consecutive



ESERCIZIO: SOTTO-SEQUENZE 0101



$I = \{I_0, I_1\}$

I_0 = ricevuto 0 ingresso

I_1 = ricevuto 1 ingresso

$U = \{0, 1\}$

0 = gli ultimi 4 bit ricevuti sono una sotto-sequenza di interesse

1 = gli ultimi 4 bit ricevuti non sono una sotto-sequenza di interesse

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

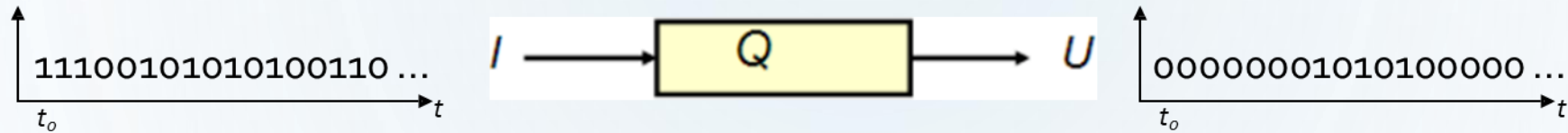
A = la macchina attende di riconoscere il primo bit della sequenza

B = la macchina ha riconosciuto 0 e attende di riconoscere 01

C = la macchina ha riconosciuto 01 attende di riconoscere 010

D = la macchina ha riconosciuto 010 e attende di riconoscere 0101

ESERCIZIO: SOTTO-SEQUENZE 0101



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

A = la macchina attende di riconoscere il primo bit della sequenza

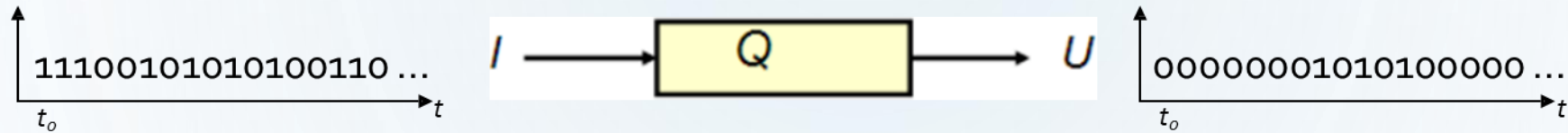
B = la macchina ha riconosciuto 0 e attende di riconoscere 01

C = la macchina ha riconosciuto 01 attende di riconoscere 010

D = la macchina ha riconosciuto 010 e attende di riconoscere 0101

Stati	Ingressi	
	0	1
A	B/0	A/0
B	B/0	C/0
C	D/0	A/0
D	B/0	C/1

ESERCIZIO: SOTTO-SEQUENZE 0101



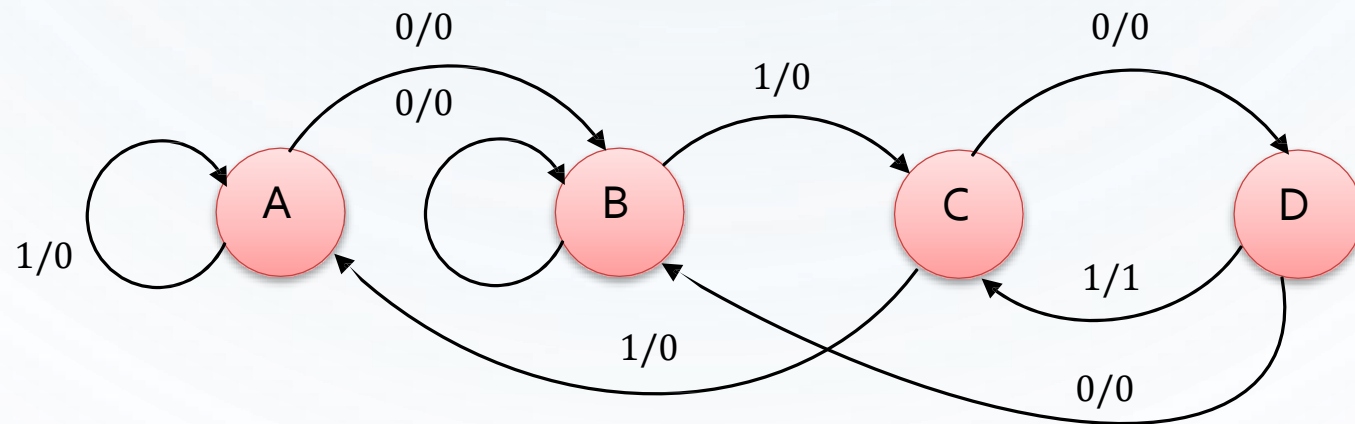
$S = \{A, B, C, D\}$

A = la macchina attende di riconoscere il primo bit della sequenza

B = la macchina ha riconosciuto 0 e attende di riconoscere 01

C = la macchina ha riconosciuto 01 attende di riconoscere 010

D = la macchina ha riconosciuto 010 e attende di riconoscere 0101

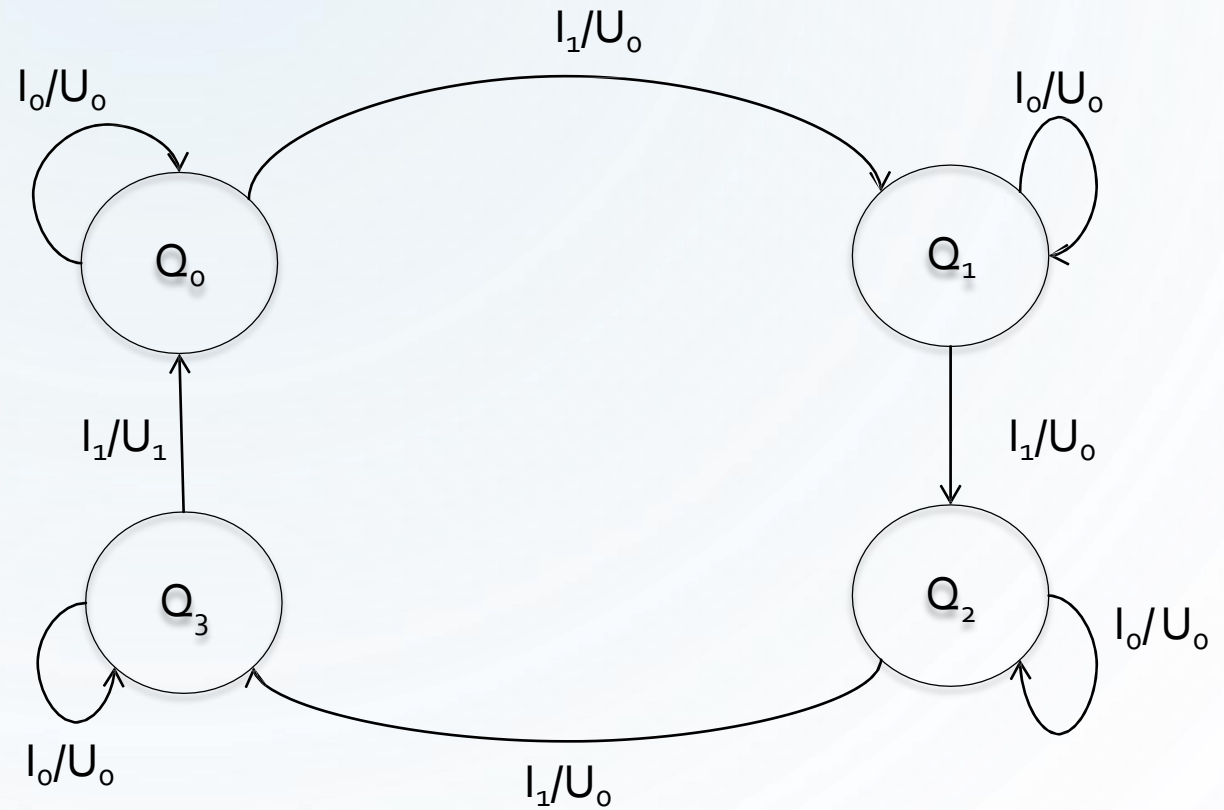


ESERCIZIO: DISTRIBUTORE BIBITE

- Si definisca la tabella e il grafo di transizione degli stati di una macchina di Mealy per la distribuzione automatica di bibite
 - tutti i tipi di bibita costano 2 Euro
 - la macchina accetta in ingresso solo monete da 50 centesimi
 - I_0 = nessuna moneta
 - I_1 = moneta inserita
 - la macchina produce in uscita la consegna di una bibita
 - U_0 = bibita non erogata
 - U_1 = bibita erogata

ESERCIZIO: DISTRIBUTORE BIBITE

- I_0 = nessuna moneta
- I_1 = moneta inserita
- U_0 = bibita non erogata
- U_1 = bibita erogata
- Q_0 = 0 centesimi
- Q_1 = 50 centesimi
- Q_2 = 1 euro
- Q_3 = 1 euro e 50 centesimi



ESERCIZIO: DISTRIBUTORE BIBITE

- I_0 = nessuna moneta
- I_1 = moneta inserita
- U_0 = bibita non erogata
- U_1 = bibita erogata
- Q_0 = 0 centesimi
- Q_1 = 50 centesimi
- Q_2 = 1 euro
- Q_3 = 1 euro e 50 centesimi

Stati	Ingressi	
	I_0	I_1
Q_0	Q_0/U_0	Q_1/U_0
Q_1	Q_1/U_0	Q_2/U_0
Q_2	Q_2/U_0	Q_3/U_0
Q_3	Q_3/U_0	Q_0/U_1

MODELLO DI MACCHINA DI TURING

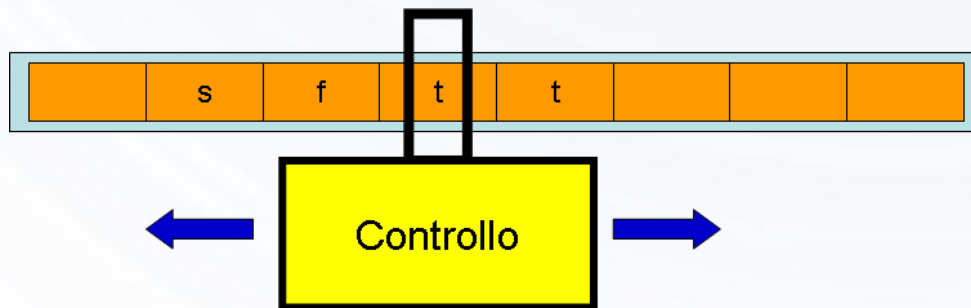


- È un particolare automa composto da una testina di scrittura/lettura capace di scrivere, leggere, e spostarsi su nastro bidirezionale potenzialmente illimitato
 - il nastro è costituito da celle su cui è possibile leggere e scrivere simboli
 - gli insiemi degli ingressi e delle uscite sono **insiemi di simboli**
 - trasforma un nastro di simboli t in un altro nastro di simboli t'
- È un modello fondamentale dell'informatica
 - permette di raggiungere risultati teorici sulla **calcolabilità** e sulla **complessità** degli algoritmi

MODELLO DI MACCHINA DI TURING



- Descrive una macchina composta da:
 - una memoria costituita da un nastro di dimensione infinita diviso in celle
 - ogni cella contiene un simbolo oppure è vuota
 - una testina di lettura/scrittura posizionabile sulle celle del nastro
 - un dispositivo di controllo che, per ogni coppia (*stato*, *simbolo letto*), determina il cambiamento di stato ed esegue azioni elaborative (resta ferma, sposta a destra, sposta a sinistra, scrivi un simbolo)



MACCHINA DI TURING: DEFINIZIONE FORMALE

- Definita dalla quintupla (A, S, f_m, f_s, f_d)
 - A è l'insieme finito dei **simboli** di ingresso e uscita
 - S è l'insieme finito degli **stati** (di cui uno è quello di terminazione)
 - f_m è la funzione di macchina $A \times S \rightarrow A$ che determina i simboli da scrivere
 - f_s è la funzione di stato $A \times S \rightarrow S$
 - f_d è la funzione di direzione $A \times S \rightarrow D = \{Sinistra, Destra, Nessuna\}$
- La macchina è capace di:
 - leggere un simbolo dal nastro
 - scrivere sul nastro il simbolo specificato dalla funzione di macchina
 - transitare in un nuovo stato interno specificato dalla funzione di stato
 - spostarsi sul nastro di una posizione come indicato dalla funzione di direzione
- La macchina si ferma quando raggiunge lo stato di terminazione

MACCHINA DI TURING E ALGORITMI

- Una macchina di Turing il cui dispositivo di controllo è capace di leggere da un nastro anche la **descrizione dell'algoritmo** è una **macchina universale** capace di simulare il lavoro compiuto da un'altra macchina qualsiasi
 - leggere dal nastro la descrizione dell'algoritmo richiede di saper
 - interpretare il linguaggio con il quale esso è stato descritto
- La **macchina di Turing Universale** è l'interprete di un linguaggio

TESI DI CHURCH E TURING

- Non esiste alcun formalismo, per modellare una determinata computazione meccanica, che sia più potente della Macchina di Turing e dei formalismi ad essi equivalenti
 - ogni algoritmo può essere codificato in termini di Macchina di Turing
 - un problema è non risolubile alitmicamente se nessuna Macchina di Turing è in grado di fornire la soluzione al problema in tempo finito
- Problemi **decidibili** → sono meccanicamente risolvibili da una macchina di Turing
- Problemi **indecidibili** → non sono meccanicamente risolvibili da una macchina di Turing

TESI DI CHURCH E TURING: CONSEGUENZE

- Se un problema si può calcolare, allora esisterà una macchina di Turing in grado di risolverlo
 - la classe delle funzioni calcolabili coincide con quella delle funzioni calcolabili da una macchina di Turing
- Un algoritmo risolvete un dato problema calcolabile è indipendente dal sistema
 - per esso esiste una macchina di Turing in grado di risolverlo
- Un algoritmo è indipendente dal linguaggio usato per descriverlo
 - per ogni linguaggio si può sempre definire una macchina di Turing universale

DOMANDE, DUBBI, PERPLESSITÀ

