

Macchine a stati finiti

Architettura degli elaboratori

M. Favalli



Department of Engineering in Ferrara

Introduzione

Modelli di FSM

Algorithmic State Machines

Introduzione

Modelli di FSM

Algorithmic State Machines

Reti sequenziali

- Le reti combinatorie consentono di risolvere solo alcuni problemi
- Supponiamo di avere una tastiera numerica e di voler costruire una rete che produce un 1 se in ingresso alla tastiera viene digitata una certa sequenza di 4 cifre decimali (altrimenti produce 0)
- La tastiera produce in uscita su 4 bit la codifica binaria del numero corrispondente al tasto che é stato premuto
- É evidente che non posso decidere se produrre un 1 sulla base del valore corrente in uscita alla tastiera
- Per risolvere il problema con reti combinatorie in cui l'uscita dipende solo dagli ingressi correnti dovrei usare 4 tastiere lucchetto con combinazione da bicicletta

Reti sequenziali: lo stato

- Per calcolare l'uscita si deve tener conto della sequenza di valori che sono stati digitati in precedenza
- Le reti in grado di calcolare l'uscita sulla base della storia degli ingressi si chiamano sequenziali
- Quanto deve essere lunga la storia di cui tengo conto ? Mi serve tutta ?
- Non serve tutta, mi basta l'informazione necessaria a calcolare l'uscita dato l'ingresso corrente
- Questa informazione si chiama **stato**

Esempio di stato

- Nel nostro caso si potrebbe tenere traccia dei 3 valori di ingresso digitati precedentemente a quello corrente
- **Serve tutta questa informazione (12 bit) ?**
- La risposta é no, se devo riconoscere la sequenza 4, 1, 8, 9 mi basta sapere quale simbolo devo aspettarmi per rimanere nella sequenza corretta
 - se ho ricevuto un simbolo errato, mi aspetto un 4
 - se ho ricevuto 4 mi aspetto 1
 - se ho ricevuto 4, 1 mi aspetto 8
 - se ho ricevuto 4, 1, 8 mi aspetto 9 e produco l'uscita 1
- Quindi ho 4 informazioni più l'ingresso corrente che posso codificare con 2 bit e ho una rete meno costosa
- Sia l'utilizzo di 12 bit per i 3 valori precedenti, sia il più efficiente uso di 2 bit costituiscono esempi di stato del mio sistema

- Le reti sequenziali che svolgono queste operazioni possono essere modellate in fase di specifica come macchine a stati finiti
- Finite State Machines (FSM)
- Si tratta di sistemi in cui i simboli di ingresso, i simboli di uscita e gli stati sono in numero finito
- I modelli di una rete elettrica (RC) o di un sistema idraulico non sono FSM perché gli ingressi, le uscite e lo stato assumono valori continui, ma hanno comunque uno stato

Modello di macchine a stati finiti

- **Le macchine a stati finiti si utilizzano per modellare di sistemi fisici caratterizzabili mediante:**
 - un insieme di variabili di ingresso (controllabili)
 - un insieme di variabili di uscita (osservabili)
 - un insieme di variabili di stato che non possono essere osservate o misurate direttamente
- **La conoscenza delle variabili di stato consente per una dato ingresso di determinare le uscite del sistema**
- **Il valore corrente delle variabili di stato viene genericamente denotato come "stato" del sistema**

- Si suppone che l'insieme delle possibili configurazioni di ingresso e di uscita e che l'insieme degli stati abbiano una cardinalità finita (realizzabilità)
- \mathcal{X} insieme finito di simboli di ingresso
- \mathcal{Z} insieme finito di simboli di uscita
- \mathcal{S} insieme finito di stati

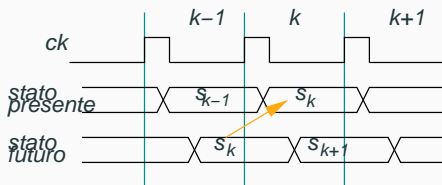
Modello del tempo

- In generale lo stato di una macchina a stati evolve in presenza di un evento sull'ingresso o sullo stato stesso: in questo caso la macchina si dice di tipo asincrono
- Le macchine a stati sincrone possono invece evolvere esclusivamente in presenza di eventi (istanti di campionamento) sul segnale di sincronizzazione (clock)
- Come si é visto, in presenza di sistemi sincroni si può astrarre un modello tempo discreto dal tempo continuo
- In tale caso la macchina si dice sincrona e si é interessati solo al valore di ingresso, uscita e stato negli istanti di sincronizzazione (t_k , con $k = 0, 1, 2, \dots$)
- Si vedranno criteri precisi che consentono di progettare macchine di questo tipo

- I sistemi fisici (macroscopici) sono deterministici, per cui fissato l'ingresso e lo stato, uscita ed evoluzione dello stato sono fissati
- Esistono anche modelli non deterministici, in cui dato l'ingresso e lo stato possono esistere diversi modi in cui la macchina può evolvere
- Tali modelli non corrispondono a sistemi fisici (macroscopici) reali, ma sono utili per descrivere ad esempio le incertezze che possono essere presenti al livello di specifiche

Significato dello stato in una FSM sincrona

- Il simbolo prodotto in uscita all'istante t_k é univocamente determinato dal simbolo di ingresso e dallo stato all'istante t_k
- Lo stato all'istante successivo t_{k+1} é anch'esso determinato in modo univoco da ingresso e stato all'istante t_k
- Lo stato all'istante t_k viene comunemente definito **stato presente**, mentre quello all'istante successivo t_{k+1} viene definito **stato futuro**



Definizione di FSM (sincrona)

Una macchina a stati finiti sincrona (M) é un sistema sincrono caratterizzato da un alfabeto di ingresso finito $\mathcal{X} = \xi_1, \xi_2, \dots, \xi_p$, un alfabeto di uscita $\mathcal{Z} = \zeta_1, \zeta_2, \dots, \zeta_q$ e un insieme finito di stati $\mathcal{S} = \sigma_1, \sigma_2, \dots, \sigma_r$ e da una coppia di funzioni:

- 1. uscita:** $z_k = \lambda(x_k, s_k)$
- 2. stato futuro (next state):** $s_{k+1} = \delta(x_k, s_k)$

Ove $x_k \in \mathcal{X}$, $z_k \in \mathcal{Z}$ e $s_k \in \mathcal{S}$ rappresentano rispettivamente il simbolo di ingresso, di uscita e lo stato all'istante k .

Definizione di FSM sincrona

La definizione precedente corrisponde a una macchina detta di Mealy, ed é definita da:

$$M = \langle S, \mathcal{X}, \mathcal{Z}, \lambda, \delta, \sigma_0 \rangle$$

ove σ_0 rappresenta lo stato a $t = t_0$ (stato iniziale)

La funzione λ é definita come $\lambda : \mathcal{X} \times \mathcal{S} \rightarrow \mathcal{Z}$

La funzione δ é definita come $\delta : \mathcal{X} \times \mathcal{S} \rightarrow \mathcal{S}$

Macchine di Mealy e di Moore

- Un caso particolare di Macchina di Mealy é la macchina di Moore dove l'uscita non dipende dall'ingresso corrente per cui: $\lambda : \mathcal{S} \rightarrow \mathcal{Z}$ e quindi $z_k = \lambda(s_k)$
- Le differenze principali é sono:
 - l'automa di Moore non può rispondere immediatamente a un cambiamento del simbolo presente in ingresso, ma solo con un ciclo di clock di ritardo
 - l'automa di Moore non richiede che gli ingressi siano stabili per un periodo di clock
- Esistono metodi per passare da una rappresentazione all'altra

- Nell'ambito delle reti logiche, il termine **automa** viene spesso utilizzato come sinonimo di FSM
- In realtà un automa é qualcosa di piú generale la cui teoria viene di solito discussa nell'ambito della teoria dei linguaggi
- Una FSM é un automa particolare detto di tipo **traduttore** in quanto trasforma una sequenza di ingressi in una di uscite
- Esistono altri tipi di automa come quelli accettori, ovvero automi che ricevono in ingresso una sequenza e stabiliscono se appartiene o meno a un certo linguaggio

Introduzione

Modelli di FSM

Algorithmic State Machines

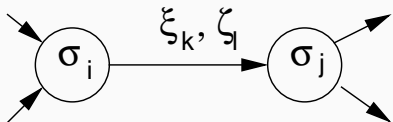
- Una macchina a stati finiti può essere descritta sia in modo comportamentale (cosa fa) che strutturale (come è fatta)
- Si hanno 2 descrizioni comportamentali
 - grafo di transizione dello stato (STG)
 - tabella di transizione dello stato
- Si ha poi il modello di Huffman che è invece descrizione strutturale

Grafo di transizione dello stato

- È una rappresentazione grafica molto comoda nelle fasi iniziali di progetto in cui si passa da una descrizione informale della macchina al suo modello comportamentale
- Si tratta di un grafo orientato in cui ogni stato è rappresentato da un nodo, e ogni arco corrisponde a una transizione dello stato
- Per ogni coppia di stati appartenenti alla relazione (stato presente - stato futuro), esiste un arco orientato che va dallo stato presente a quello futuro

Esempi

Macchina di Mealy: ciascun arco é annotato dal simbolo di ingresso corrispondente alla transizione di stato e dal simbolo di uscita prodotto



Macchina di Moore: il simbolo di uscita é annotato all'interno del nodo

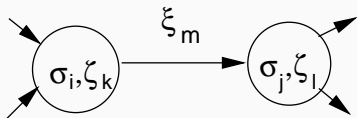


Tabella di transizione dello stato

- Si tratta di una descrizione tabellare in cui si ha una riga per ogni stato (presente) e una colonna per ogni simbolo di ingresso
- La casella σ_i, ξ_k di tale tabella contiene i valori di stato futuro e uscita forniti dalle funzioni δ e λ
- Nella macchina di Moore, per compattezza, l'informazione sull'uscita (che dipende solo dallo stato) é riportata in unica colonna

Rappresentazione utile per manipolazioni sistematiche della FSM

Tabella di transizione dello stato (Mealy)

	ξ_0	ξ_1	ξ_k	ξ_q
σ_0						
....						
σ_i				$\delta(\sigma_i, \xi_k), \lambda(\sigma_i, \xi_k)$		
....						
σ_p						

Tabella di transizione dello stato (Moore)

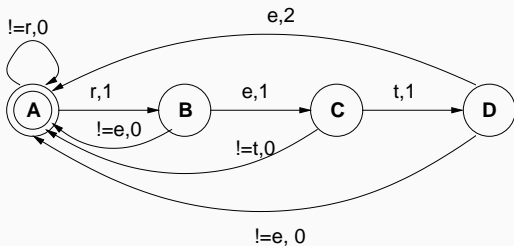
	ξ_0	ξ_1	ξ_k	ξ_q	
σ_0							$\lambda(\sigma_0)$
....							
σ_i				$\delta(\sigma_i, \xi_k)$			$\lambda(\sigma_i)$
....							
σ_p							$\lambda(\sigma_p)$

Esempio di STG (1)

Si realizzi una macchina a stati (automa) che riceve in ingresso una sequenza di caratteri e riconosce la parola "rete". L'uscita prodotta é 0 se non si sta riconoscendo la parola, 1 mentre la si sta riconoscendo e 2 una volta che sia stat riconosciuta. Si utilizzi una macchina di Mealy.

Nota: i simboli errati vengono scartati.

$$M = \langle S = \{A, B, C, D\}, \mathcal{X} = \{a, b, c, d, \dots, z\}, \mathcal{Z} = \{0, 1, 2\}, \lambda, \delta, \sigma_0 = A \rangle$$

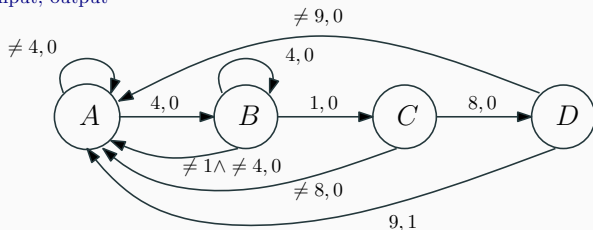


Esempio di STG - Mealy (2)

Si realizzi una macchina a stati (automa) che riconosce la sequenza di cifre 4, 1, 8, 9. L'uscita prodotta é 0 se non si sta riconoscendo la sequenza, 1 se la si é riconosciuta

$$M = \langle S = \{A, B, C, D\}, \mathcal{X} = \{0, 1, 2, 3, \dots, 9\}, \mathcal{Z} = \{0, 1\}, \lambda, \delta, \sigma_0 = A \rangle$$

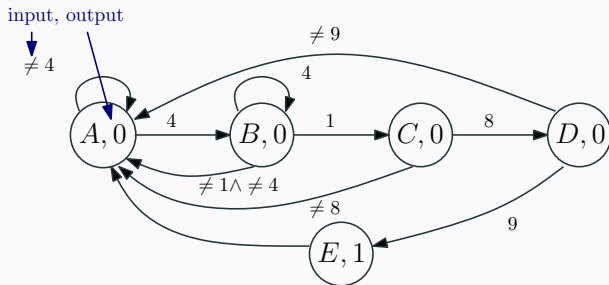
input, output



Esempio di STG - Moore (3)

Si realizzi una macchina a stati (automa) che riconosce la sequenza di cifre 4, 1, 8, 9. L'uscita prodotta é 0 se non si sta riconoscendo la sequenza, 1 se la si é riconosciuta

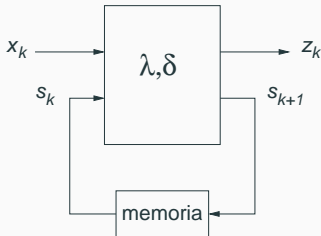
$$M = \langle S = \{A, B, C, D, E\}, \mathcal{X} = \{0, 1, 2, 3, \dots, 9\}, \mathcal{Z} = \{0, 1\}, \lambda, \delta, \sigma_0 = A \rangle$$



- Dato uno stato iniziale (σ_i) e una sequenza di ingressi, si può calcolare la risposta del sistema in maniera piuttosto semplice sia utilizzando il grafo o la tabella di transizione dello stato
- Si noti che il processo di elaborazione descritto da una FSM é sequenziale
- Come realizzereste in C o in un altro linguaggio ad alto livello un simulatore di FSM ?

Modello di Huffman

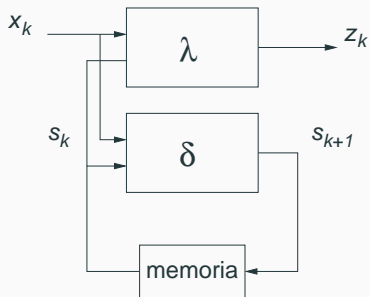
- Affinché le funzioni δ e λ possano calcolare lo stato futuro e l'uscita, lo stato presente deve rimanere stabile in ingresso per un periodo di clock.
- Questo richiede una opportuna rete di ritardo che impedisca che i cambiamenti dello stato futuro si riflettano immediatamente su quello presente



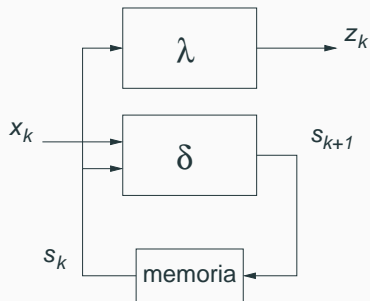
- Si nota che il blocco λ, δ non ha memoria e quindi può essere realizzato in maniera combinatoria

Modello di Huffman

Modello di Mealy



Modello di Moore



Supponendo che i simboli di ingresso e di uscita abbiano una codifica binaria, tale blocco funzionale può essere descritto in diversi modi:

- al livello comportamentale: espressioni dello stato futuro e dell'uscita (eventualmente rappresentate con ROBDD)**
- al livello strutturale: rete logica combinatoria**

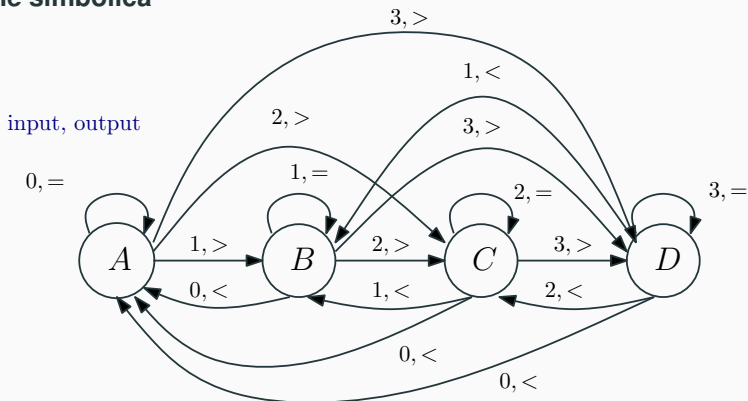
FSM definite su ingressi e uscite binarie

- Per ora abbiamo visto descrizioni delle FSM con gli ingressi definiti in maniera simbolica (lettere, cifre)
- In ambito digitale, tali simboli sono sostituiti da configurazioni binarie che li codificano e quindi abbiamo variabili (x_i) di ingresso e (z_i) di uscita
- Per esempio nella FSM che produce 1 in presenza della sequenza di ingresso 4, 1, 8, 9 il simbolo di ingresso viene codificato come intero senza segno con 4 variabili binarie
- Da qui ci occuperemo prevalentemente di questo tipo di FSM

Esempio: FSM simbolica

Si vuole realizzare una FSM di Mealy che riceve in ingresso 4 simboli diversi (0, 1, 2, 3) e produce in uscita il simbolo $>$, $<$ o $=$ dipendentemente dalla relazione d'ordine fra il simbolo corrente e quello precedente (se arriva, ad esempio, 0 e poi 1, allora si produce $>$)

Versione simbolica



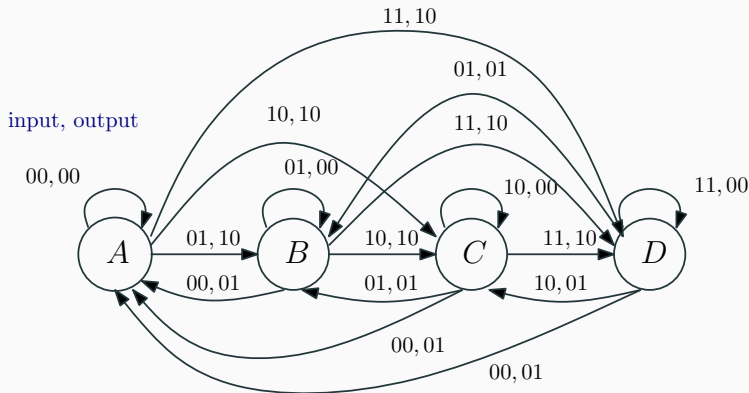
Esempio: tabella di transizione dello stato

Si ottiene dal STG

stato presente	ingresso			
	0	1	2	3
<i>A</i>	<i>A</i> , =	<i>B</i> , >	<i>C</i> , >	<i>D</i> , >
<i>B</i>	<i>A</i> , <	<i>B</i> , =	<i>C</i> , >	<i>D</i> , >
<i>C</i>	<i>A</i> , <	<i>B</i> , <	<i>C</i> , =	<i>D</i> , >
<i>D</i>	<i>A</i> , <	<i>B</i> , <	<i>C</i> , <	<i>D</i> , =

Esempio: FSM con ingressi e uscite codificati in binario

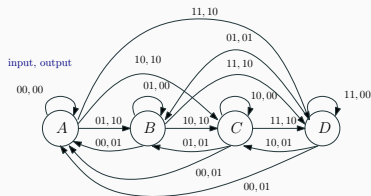
ingresso		uscita	
simbolo	codifica $x_1 x_0$	simbolo	codifica $z_1 z_0$
0	00	=	00
1	01	<	01
2	10	>	10
3	11		



Esempio: tabella di transizione dello stato

Si ottiene dal STG

stato presente	ingressi $x_1 x_0$			
	00	01	10	11
A	A, 00	B, 10	C, 10	D, 10
B	A, 01	B, 00	C, 10	D, 10
C	A, 01	B, 01	C, 00	D, 10
D	A, 01	B, 01	C, 01	D, 00

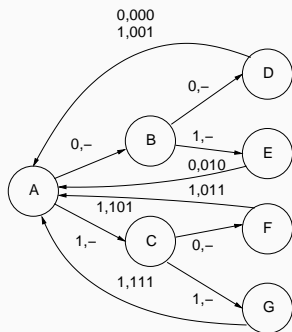


Macchine non completamente specificate

- **Esistono macchine che risultano essere non completamente specificate**
- **Per alcuni valori di stato presente e di ingresso l'uscita può non essere rilevante. In tale caso si indica un'indifferenza nello STG o nella tabella di transizione**
- **Alcune configurazioni o sequenze di ingresso possono non essere possibili e quindi per alcuni stati presenti e ingressi le uscite possono non essere specificate**
- **Queste condizioni non vengono rappresentate sullo STG, mentre nella tabella si indica una condizione di indifferenza per lo stato futuro lo stato**

Esempio

Macchina che analizza parole di 3 bit che rappresentano un numero intero senza segno ricevuto serialmente a partire dal bit di maggior peso. Compito della macchina é fornire in uscita (sull'ultimo bit ricevuto) la codifica binaria del numero ricevuto. La sorgente non invia numeri non primi



	0	1
A	B,-	C,-
B	D,-	E,-
C	F,-	G,-
D	A,000	A,001
E	A,010	A,011
F	-, -	A,101
G	-, -	A,111

Introduzione

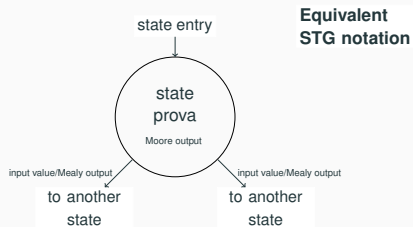
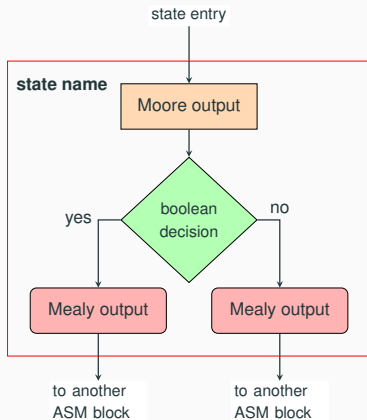
Modelli di FSM

Algorithmic State Machines

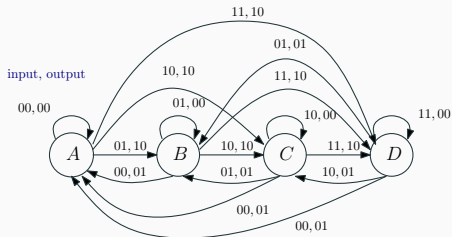
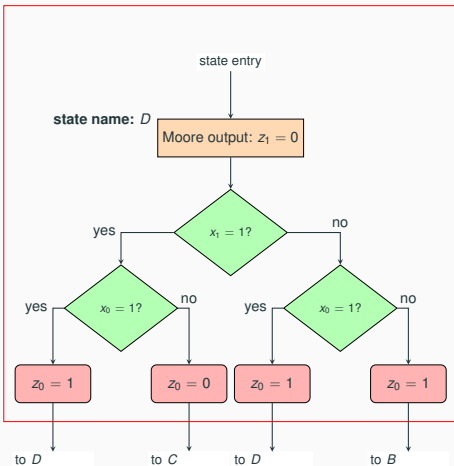
Algorithmic State Machines

- **Le Algorithmic State Machines (ASM) son un formalismo alternativo a STG**
- **Riprendono il formalismo dei diagrammi di flusso usati nell'ambito del software**
- **A ogni stato della FSM é associato un blocco che definisce:**
 - **i valori per le uscite di Moore (se esistono), ovvero le uscite che dipendono solo dallo stato**
 - **i valori per le uscite di Mealy (se esistono), ovvero le uscite che dipendono da stato e ingresso**
 - **un test che determina la selezione dello stato futuro**
- **Per evitare la proliferazione di stati, il blocco può avere più test e più rami di uscita**

Blocco di una ASM



Esempio



- **Abbiamo visto come le macchine a stati consentano di modellare reti sequenziali**
- **I formalismi introdotti, come lo STG, consentono di descrivere in maniera comportamentale una FSM**
- **Rimane da vedere il modo in cui passare da questi modelli a un modello strutturale**