Macchine a Stati finiti

Prof. Alberto Borghese
Dipartimento di Informatica
alberto.borghese@unimi.it

Università degli Studi di Milano

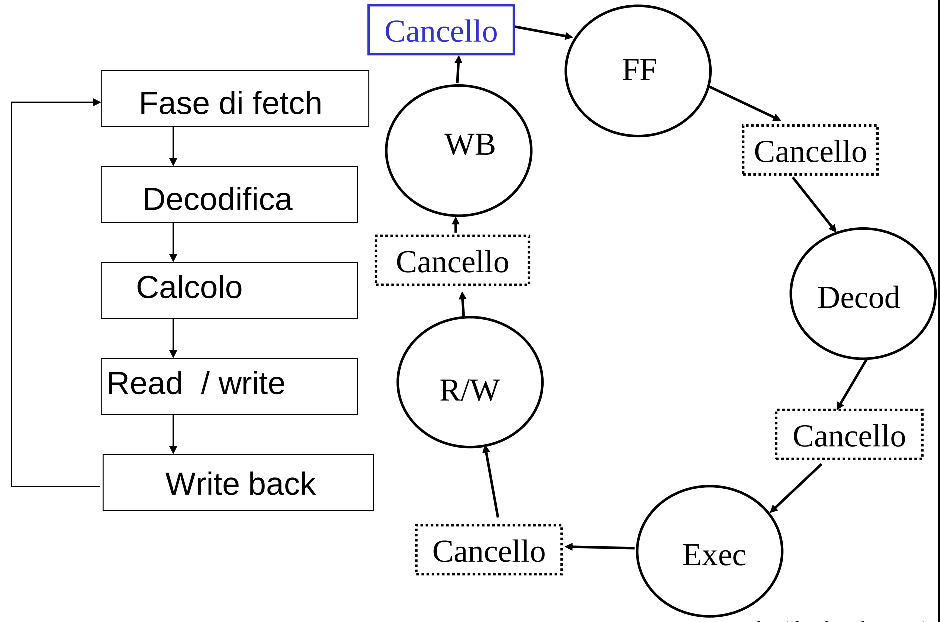
Riferimento al Patterson: Sezione B.10
(per approfondimenti, D.3 e D.4)

Sommario

Macchine a stati finiti

Esempio: sintesi di un controllore per venditore di bibite.

La CPU come macchina sequenziale



Controllore di una macchina venditrice di bibite

Voglio costruire un controllore di una macchinetta che eroga caffè quando l'utente ha inserito 30c.

Accetta in **input**: {0c, 10c, 20c} e non dà resto.

Questo controllore produrrà in **output** 2 informazioni:
- N = No Caffè
- C = Caffè.

Noi vediamo solamente i dati di input e di output.

NB L'uscita non dipende dalla moneta inserita al tempo t, ma dipende dal totale!
Il totale richiede di tenere conto anche delle monete inserite in precendeza.

Il totale è la **situazione** della pancia della macchina -> **stato della macchina**

Lo stato è interno alla macchina.

Come?

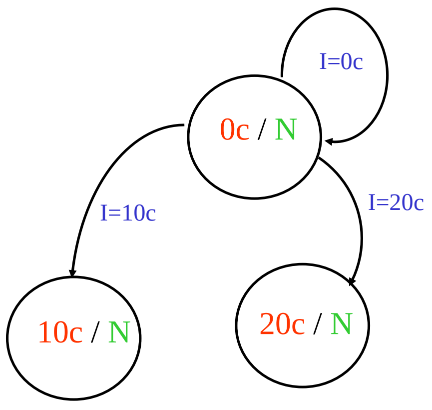
Una macchina venditrice di bibite - I

Voglio costruire un controllore di una macchinetta che eroga caffè quando l’utente ha inserito 30c.

Accetta: {0c, 10c, 20c} e non dà resto.

N = No Caffè
C = Caffè.

Approccio costruttivo.



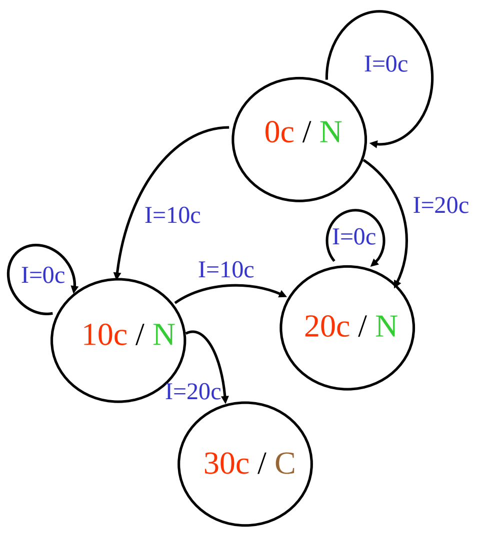
Una macchina venditrice di bibite - II

Voglio costruire un controllore di una macchinetta che eroga caffè quando l’utente ha inserito 30c.

Accetta: {0c, 10c, 20c} e non dà resto.

N = No Caffè
C = Caffè.

Approccio costruttivo.



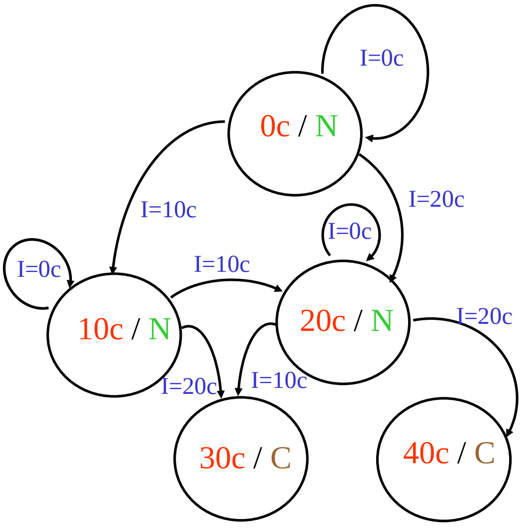
Una macchina venditrice di bibite - III

Voglio costruire una macchinetta che eroga caffè quando l'utente ha inserito 30c.

Accetta: {0c, 10c, 20c} e non dà resto.

N = No Caffè
C = Caffè.

Approccio costruttivo.



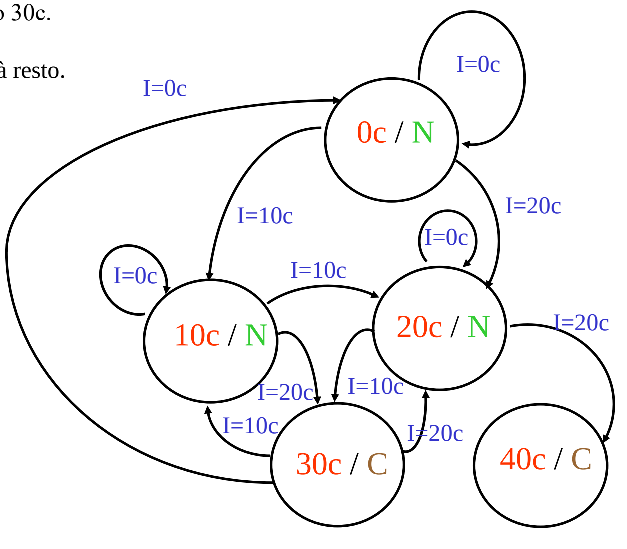
Una macchina venditrice di bibite - IV

Voglio costruire una macchinetta che eroga caffè quando l'utente ha inserito 30c.

Accetta: {0c, 10c, 20c} e non dà resto.

N = No Caffè
C = Caffè.

Approccio costruttivo.



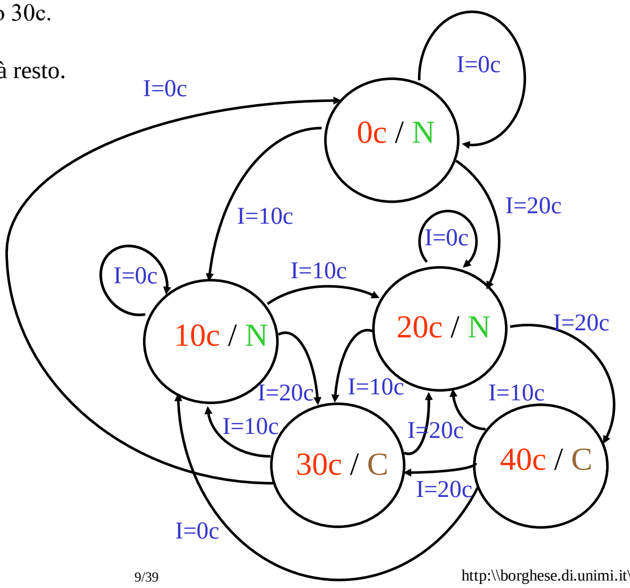
Una macchina venditrice di bibite - V

Voglio costruire una macchinetta che eroga caffè quando l'utente ha inserito 30c.

Accetta: {0c, 10c, 20c} e non dà resto.

N = No Caffè
C = Caffè.

Approccio costruttivo.



Si suppone che la cifra in eccesso rimanga nella pancia della macchina.

State transition graph

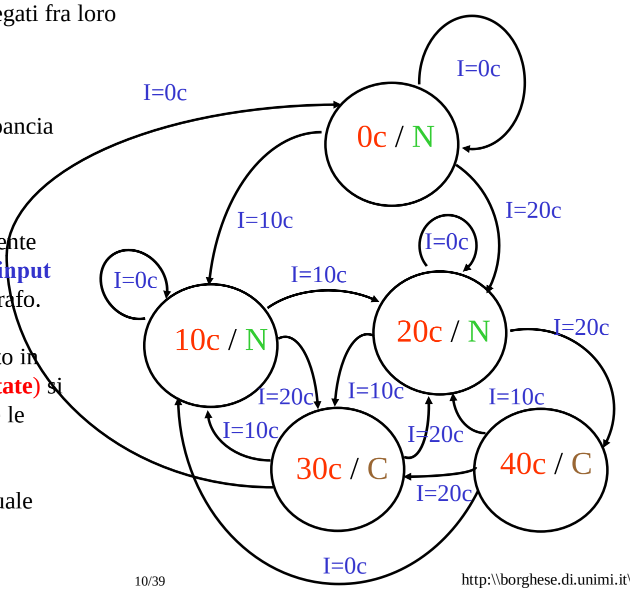
Un grafo è un insieme di elementi detti **nodi** che possono essere collegati fra loro da linee chiamate **archi**.

Il totale è la situazione della pancia della macchina -> **stato della macchina,** i **nodi** del grafo.

La moneta inserita correntemente nella macchina rappresenta **l'input alla macchina**, un **arco** del grafo.

Per ogni stato, viene esaminato in quale **stato prossimo (next state)** si troverà la macchina. Descrive le **transizioni di stato**.

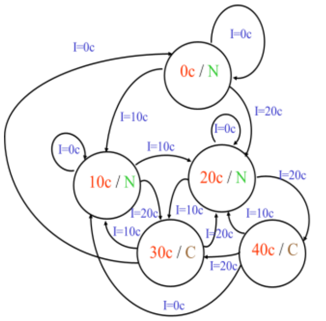
Per ogni **stato** si determina quale sarà l'**uscita** della macchina.



Macchina a Stati Finiti (di Moore)

La Macchina di (Edward) Moore è definita, in teoria degli automi, dalla sestupla:

$< X, I, Y, f(.), g(.), X_{ini} >$



X: insieme degli stati (in numero finito).

I: insieme di ingresso: tutti i simboli che si possono presentare in ingresso.

Y: insieme di uscita: tutti i simboli che si possono produrre in uscita.

f(.): funzione stato prossimo: $X^* = f(X,I)$. Definisce l'evoluzione della macchina nel tempo. L'evoluzione è deterministica.

g(.): funzione di uscita: $Y= g(X)$, è **funzione solamente dello stato attuale** nelle macchine di Moore. Non è funzione dello stato prossimo, X^*.

Stato iniziale: X_{ini}. Per il buon funzionamento della macchina è previsto uno stato iniziale, al quale la macchina può essere portata mediante un comando di reset.

La nostra macchina di Moore

La Macchina di Moore è definita, in teoria degli automi, dalla sestupla:

$< X, I, Y, f(.), g(.), X_{ini} >$

X: insieme degli stati (in numero finito): {0c, 10c, 20c, 30c, 40c}

I: insieme di ingresso: tutti i simboli che si possono presentare in ingresso: {0c, 10c, 20c}

Y: insieme di uscita: tutti i simboli che si possono generare in uscita: {Niente, Caffè}.

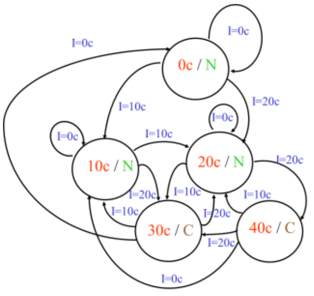
f(.): funzione stato prossimo: $X^* = f(X,I)$. Definisce l'evoluzione della macchina nel tempo. L'evoluzione è deterministica.

g(.): funzione di uscita: $Y= g(X)$, è **funzione solamente dello stato attuale** nelle macchine di Moore. Non è funzione dello stato prossimo, X^*.

Stato iniziale: X_{ini}. Per il buon funzionamento della macchina è previsto uno stato iniziale, al quale la macchina può essere portata mediante un comando di reset.

Descrizione di una macchina di Moore

STG: State Transition Graph (Diagramma degli stati o Grafo delle transizioni). Ad ogni nodo è associato uno stato. Un arco orientato da uno stato x_i ad uno stato x_j, contrassegnato da un simbolo (di ingresso) α, rappresenta una transizione (passaggio di stato) che si verifica quando la macchina, essendo nello stato x_i, riceve come ingresso il simbolo α.



STT: State Transition Table (Tabella degli Stati). Per ogni coppia, (Stato presente – Ingresso), si definisce l'Uscita e lo Stato Prossimo. La forma è tabellare e ricorda le tabelle della verità da cui è derivata.

Sommario

Macchine a stati finiti

Esempio: sintesi di un controllore per venditore di bibite.

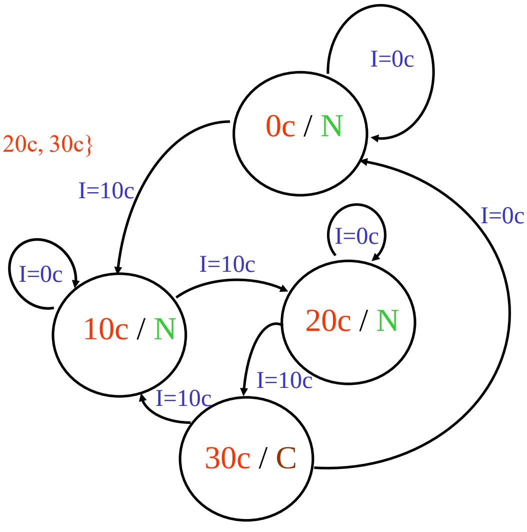
STG di una macchina venditrice di bibite (Semplificata)

Voglio costruire una macchinetta che eroga caffè quando l'utente ha inserito 30c.
Accetta solamente monete da 10c.

I = {0c, 10c}
Y = {**N**ulla, **C**affè}
X = "Monete accumulate" = {0, 10c, 20c, 30c}

$X^* = f(X,I)$
$Y = g(X)$

$X_o = 0c$

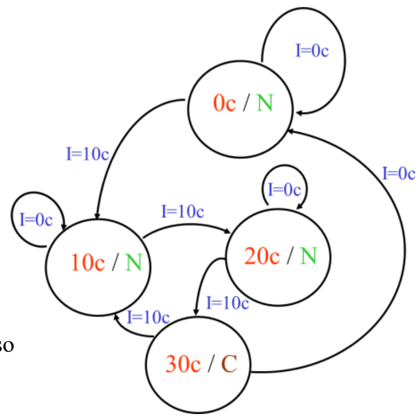


La Macchina di Moore è definita, in teoria degli automi, dalla sestupla: < X, I, Y, f(.), g(.), X_{ini} >

STT della vendor machine - I

X \ I	I: 0c	I: 10c
0c	**0c**	**10c**
10c	**10c**	**20c**
20c	**20c**	**30c**
30c	**0c**	**10c**

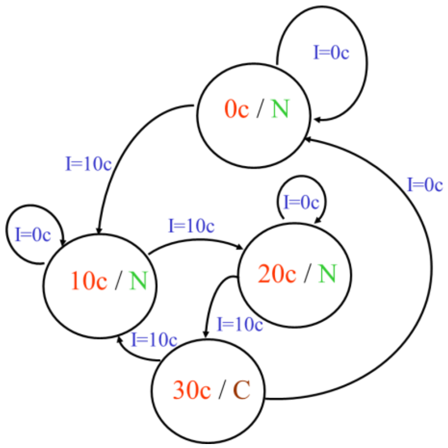
X*



Il controllore controlla a ogni iterazione l'ingresso e a ogni iterazione aggiorna lo **stato prossimo**, $X^* = f(X,I)$.

STT della vendor machine - II

X \ Y	Y
0c	**Nulla**
10c	**Nulla**
20c	**Nulla**
30c	**Caffè**

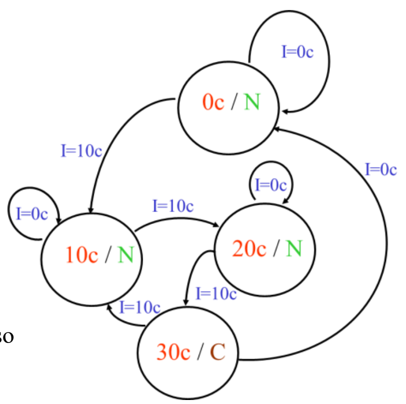


Il controllore controlla a ogni iterazione lo **stato attuale** aggiorna l'**uscita**, Y = g(X)

STT della vendor machine - III

I \ X	I: 0c	I: 10c	Y
0c	**0c**	**10c**	**Nulla**
10c	**10c**	**20c**	**Nulla**
20c	**20c**	**30c**	**Nulla**
30c	**0c**	**10c**	**Caffè**

X*

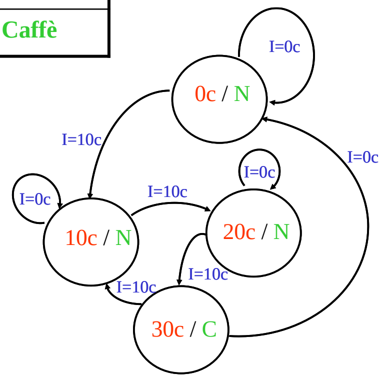


Il controllore controlla a ogni iterazione l'ingresso e a ogni iterazione aggiorna lo **stato prossimo**, X* = f(X,I) e l'**uscita**, Y = g(X)

Dal linguistico al numerico

X \ I	I: 0c	I: 10c	Y
0c	**0c**	**10c**	**Nulla**
10c	**10c**	**20c**	**Nulla**
20c	**20c**	**30c**	**Nulla**
30c	**0c**	**10c**	**Caffè**

I = {0c, 10c}
Y = {Nulla, Caffè}
X = “Monete accumulate”
= {0, 10c, 20c, 30c}



Occorre codificare ingresso, stato ed uscita.
=
Assegnare un **codice numerico** ad ogni possibile stato, ingresso, uscita.

Abbiamo definito: < X, I, Y, f(.), g(.), X_{ini} >

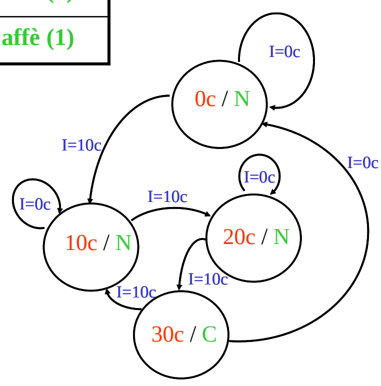
Codifica della STT della vendor machine

X \ I	I: 0c (0)	I: 10c (1)	Y
0c (0)	**0c (0)**	**10c (1)**	**Nulla (0)**
10c (1)	**10c (1)**	**20c (2)**	**Nulla (0)**
20c (2)	**20c (2)**	**30c (3)**	**Nulla (0)**
30c (3)	**0c (0)**	**10c (1)**	**Caffè (1)**

X*

I = {0c, 10c} = {0, 1}
Y = {Nulla, Caffè} = {0, 1}
X = {0, 10c, 20c, 30c} = {0, 1, 2, 3}

X* = f(X,I) da sintetizzare
Y = g(X) da sintetizzare



Abbiamo definito: < X, I, Y, f(.), g(.), X_{ini} >

Significato della codifica

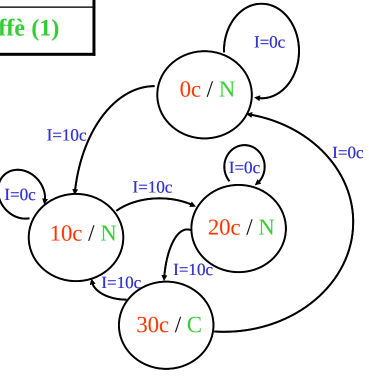
X \ I	I 0c (0)	I 10c (1)	Y
0c (0)	**0c (0)**	**10c (1)**	**Nulla (0)**
10c (1)	**10c (1)**	**20c (2)**	**Nulla (0)**
20c (2)	**20c (2)**	**30c (3)**	**Nulla (0)**
30c (3)	**0c (0)**	**10c (1)**	**Caffè (1)**

X*

I = {0c, 10c} = {0, 1}
Y = {Nulla, Caffè} = {0, 1}
X = {0, 10c, 20c, 30c} = {0, 1, 2, 3}

if (stato == 3) the Y =1

if (stato = 0 and I = 1) then stato_prossimo = 1
if (stato = 0 and I = 0) then stato_prossimo = 0
if (stato = 1 and I = 1) then stato_prossimo = 2
if (stato = 1 and I = 0) then stato_prossimo = 1
if (stato = 2 and I = 1) then stato_prossimo = 3
if (stato = 2 and I = 0) then stato_prossimo = 2
if (stato = 3 and I = 1) then stato_Prossimo = 1
if (stato = 3 and I = 0) then stato_Prossimo = 0



Codifica binaria della STT

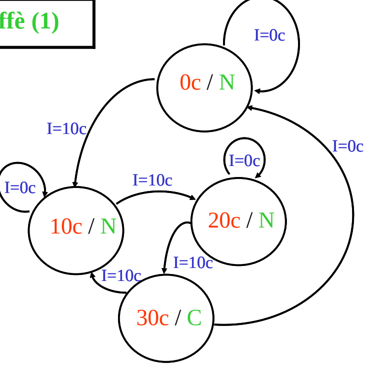
X \ I	I 0c (0)	I 10c (1)	Y
0c (00)	**0c (00)**	**10c (01)**	**Nulla (0)**
10c (01)	**10c (01)**	**20c (10)**	**Nulla (0)**
20c (10)	**20c (10)**	**30c (11)**	**Nulla (0)**
30c (11)	**0c (00)**	**10c (01)**	**Caffè (1)**

X*

I = {0c, 10c} = {0, 1}
Y = {Nulla, Caffè} = {0, 1}
X = {0, 10c, 20c, 30c} = {00, 01, 10, 11}

X* = f(X,I) da sintetizzare
Y = g(X) da sintetizzare

Abbiamo definito: < X, I, Y, f(.), g(.), X_{ini} >

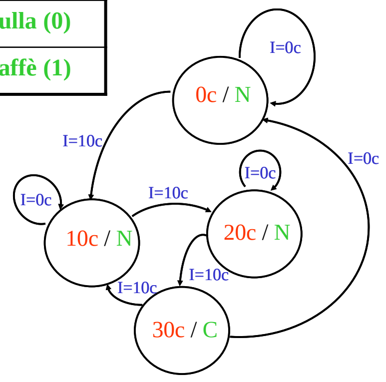


Significato della Codifica binaria della STT - Y

X \ I	I: 0c (0)	10c (1)	Y
0c (00)	**0c (00)**	**10c (01)**	**Nulla (0)**
10c (01)	**10c (01)**	**20c (10)**	**Nulla (0)**
20c (10)	**20c (10)**	**30c (11)**	**Nulla (0)**
30c (11)	**0c (00)**	**10c (01)**	**Caffè (1)**

X*

I = {0c, 10c} = {0, 1}
Y = {<u>N</u>ulla, <u>C</u>affè} = {0, 1}
X = {0, 10c, 20c, 30c} = {00, 01, 10, 11}



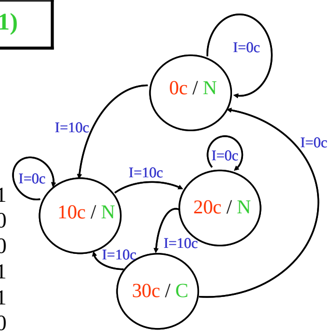
if (X_1 ==1 and X_0 == 1) the Y =1

Significato della Codifica della STT - X_0*

X \ I	I: 0c (0)	10c (1)	Y
0c (00)	**0c (00)**	**10c (01)**	**Nulla (0)**
10c (01)	**10c (01)**	**20c (10)**	**Nulla (0)**
20c (10)	**20c (10)**	**30c (11)**	**Nulla (0)**
30c (11)	**0c (00)**	**10c (01)**	**Caffè (1)**

I = {0c, 10c} = {0, 1}
Y = {<u>N</u>ulla, <u>C</u>affè} = {0, 1}
X = {0, 10c, 20c, 30c} = {00, 01, 10, 11}

X*



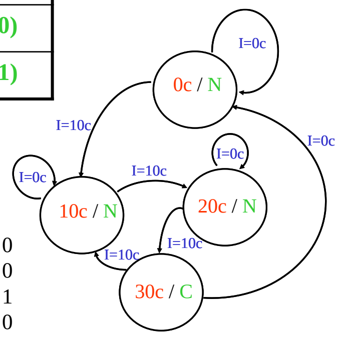
if ((X_1 ==0 and X_0 == 0) and I = 1) then LSB_stato_prossimo = X_0* = 1
if ((X_1 ==0 and X_0 == 0) and I = 0) then LSB_stato_prossimo = X_0* = 0
if ((X_1 ==0 and X_0 == 1) and I = 1) then LSB_stato_prossimo = X_0* = 0
if ((X_1 ==0 and X_0 == 1) and I = 0) then LSB_stato_prossimo = X_0* = 1
if ((X_1 ==1 and X_0 == 0) and I = 1) then LSB_stato_prossimo = X_0* = 1
if ((X_1 ==1 and X_0 == 0) and I = 0) then LSB_stato_prossimo = X_0* = 0
if ((X_1 ==1 and X_0 == 1) and I = 1) then LSB_stato_Prossimo = X_0* = 1
if ((X_1 ==1 and X_0 == 1) and I = 0) then LSB_stato_Prossimo = X_0* = 0

Significato della Codifica della STT - S_1*

I / X	I: 0c (0)	I: 10c (1)	Y
0c (00)	**0c (00)**	**10c (01)**	**Nulla (0)**
10c (01)	**10c (01)**	**20c (10)**	**Nulla (0)**
20c (10)	**20c (10)**	**30c (11)**	**Nulla (0)**
30c (11)	**0c (00)**	**10c (01)**	**Caffè (1)**

I = {0c, 10c} = {0, 1}
Y = {Nulla, Caffè} = {0, 1}
X = {0, 10c, 20c, 30c} = {00, 01, 10, 11}

X*



if ((X_1 ==0 and X_0 == 0) and I = 1) then MSB_stato_prossimo = X_1* = 0
if ((X_1 ==0 and X_0 == 0) and I = 0) then MSB_stato_prossimo = X_1* = 0
if ((X_1 ==0 and X_0 == 1) and I = 1) then MSB_stato_prossimo = X_1* = 1
if ((X_1 ==0 and X_0 == 1) and I = 0) then MSB_stato_prossimo = X_1* = 0
if ((X_1 ==1 and X_0 == 0) and I = 1) then MSB_stato_prossimo = X_1* = 1
if ((X_1 ==1 and X_0 == 0) and I = 0) then MSB_stato_prossimo = X_1* = 1
if ((X_1 ==1 and X_0 == 1) and I = 1) then MSB_stato_Prossimo = X_1* = 0
if ((X_1 ==1 and X_0 == 1) and I = 0) then MSB_stato_Prossimo = X_1* = 0

Macchina a Stati Finiti (di Moore)

La Macchina di Moore è definita, in teoria degli automi, dalla sestupla:

< X, I, Y, f(.), g(.), X_{ini} >

X: insieme degli stati (in numero finito).

I: insieme di ingresso: tutti i simboli che si possono presentare in ingresso. In caso di codifica binaria, se abbiamo n linee in ingresso (variabili binarie), avremo 2^n possibili simboli da leggere in ingresso (configurazioni).

Y: insieme di uscita: tutti i simboli che si possono generare in uscita. In caso di codifica binaria, se abbiamo m linee in uscita (variabili binarie), avremo 2^m possibili simboli in uscita (configurazioni).

f(.): funzione stato prossimo: X' = f(X,I). Definisce l'evoluzione della macchina nel tempo. L'evoluzione è deterministica. La funzione è una funzione logica.

g(.): funzione di uscita: Y= g(X) nelle macchine di Moore. E' una funzione logica.

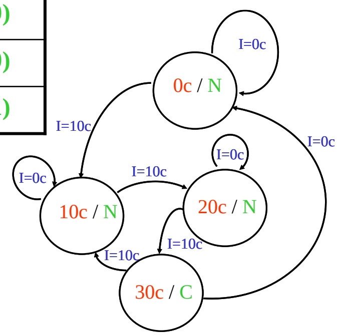
Stato iniziale: X_{ini}. Per il buon funzionamento della macchina è previsto uno stato iniziale, al quale la macchina può essere portata mediante un comando di reset.

Codifica binaria della FSM della Vendor Machine

I / X - X_1X_0	I: 0c (0)	I: 10c (1)	Y
0c (00)	**0c (00)**	**10c (01)**	**Nulla (0)**
10c (01)	**10c (01)**	**20c (10)**	**Nulla (0)**
20c (10)	**20c (10)**	**30c (11)**	**Nulla (0)**
30c (11)	**0c (00)**	**10c (01)**	**Caffè (1)**

X*

X è su 2 cifre =>
2 bit X_1 e X_0



I = {0c, 10c} = {0, 1}
Y = {Nulla, Caffè} = {0, 1}
X = {0, 10c, 20c, 30c} = {00, 01, 10, 11}

X* = f(X,I) da sintetizzare
Y = g(X) da sintetizzare

Abbiamo definito: < X, I, Y, f(.), g(.), X_{ini} >

Sintesi della funzione di uscita della FSM della Vendor Machine

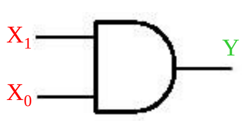
I / X - X_1X_0	Y
(0c) 00	**(Nulla) 0**
(10c) 01	**(Nulla) 0**
(20c) 10	**(Nulla) 0**
(30c) 11	**(Caffè) 1**

X è su 2 cifre =>
2 bit X_1 e X_0

X_1	X_0	Y
0	0	0
0	1	0
1	0	0
1	1	1

I = {0c, 10c} = {0, 1}
Y = {Nulla, Caffè} = {0, 1}
X = {0, 10c, 20c, 30c} = {00, 01, 10, 11}

X* = f(X,I) a sintetizzare
Y = g(X) da sintetizzare



Sintesi della funzione stato prossimo della FSM della Vendor Machine

I / X - X_1X_0	0c (0)	10c (1)
0c (00)	**0c (00)**	**10c (01)**
10c (01)	**10c (01)**	**20c (10)**
20c (10)	**20c (10)**	**30c (11)**
30c (11)	**0c (00)**	**10c (01)**

X è su 2 cifre => 2 bit: X_1 e X_0

Devo sintetizzare:
$X_1^{t+1} = f_1(X_0^t, X_1^t, I)$
$X_0^{t+1} = f_0(X_0^t, X_1^t, I)$

$X^* = \{X^*_1\ X^*_0\}$

$X_1\ X_0\ I$	$X^*_1\ X^*_0$
0 0 0	0 0
0 0 1	0 1
0 1 0	0 1
0 1 1	1 0
1 0 0	1 0
1 0 1	1 1
1 1 0	0 0
1 1 1	0 1

I = {0, 1}
Y = {0, 1}
X = {00, 01, 10, 11}

X* = f(X,I)
$Y = g(X) = X_1\ X_0$

I 2 bit di stato prossimo vengono sintetizzati sepratamente. Sono entrambi funzione dei 2 bit di stato all'istante attuale e del bit di ingresso

A

Sintesi della funzione stato prossimo della FSM della Vendor Machine

I / X_1X_0	0	1	Y
0 0	**00**	**01**	**0**
0 1	**01**	**10**	**0**
1 0	**10**	**11**	**0**
1 1	**00**	**01**	**1**

X è su 2 cifre => 2 bit: X_1 e X_0

Devo sintetizzare:
$X_1^{t+1} = f_1(X_0^t, X_1^t, I)$
$X_0^{t+1} = f_0(X_0^t, X_1^t, I)$

$X^* = \{X^*_1\ X^*_0\}$

$X_1\ X_0\ I$	$X^*_1\ X^*_0$
0 0 0	0 0
0 0 1	0 1
0 1 0	0 1
0 1 1	1 0
1 0 0	1 0
1 0 1	1 1
1 1 0	0 0
1 1 1	0 1

I = {0, 1}
Y = {0, 1}
X = {00, 01, 10, 11}

$X^* = f(X,I) \Rightarrow X^*_0 = I\ \overline{X}_1\overline{X}_0 + \overline{I}\ \overline{X}_1\ \underline{X_0} + I\ \underline{X_1}\overline{X}_0 + I\ X_1\ X_0 =$

$I(X_1 + X_0) + I\ X_1\ X_0$

$X^*_1 = \ldots.$

I 2 bit di stato prossimo vengono sintetizzati sepratamente. Sono entrambi funzione dei 2 bit di stato all'istante attuale e del bit di ingresso

Sintesi della funzione stato prossimo della FSM della Vendor Machine

X_1X_0 \ I	I = 0	I = 1
0 0	00	01
0 1	01	10
1 0	10	11
1 1	00	01

X è su 2 cifre => 2 bit: X_1 e X_0

Devo sintetizzare:

$X_1^* = X_1^{t+1} = f_1(X_0^t, X_1^t, I)$

$X_0^* = X_0^{t+1} = f_0(X_0^t, X_1^t, I)$

$X^* = \{X^*_1\ X^*_0\}$

$X_1\ X_0\ I$	$X^*_1\ X^*_0$
0 0 0	0 0
0 0 1	0 1
0 1 0	0 1
0 1 1	1 0
1 0 0	1 0
1 0 1	1 1
1 1 0	0 0
1 1 1	0 1

I = {0, 1}

Y = {0, 1}

X = {00, 01, 10, 11}

$X^* = f(X,I) \Rightarrow X^*_0 = I\ \overline{X}_1\overline{X}_0 + \overline{I}\ \overline{X}_1\ X_0 + \underline{I}\ X_1\underline{\overline{X}_0} + I\ X_1\ X_0 =$

$I(X_1 + X_0) + I\ X_1\ X_0$

$X^*_1 = X_1\ \overline{X}_0 + \overline{X}_1\ X_0\ I$

I 2 bit di stato prossimo vengono sintetizzati sepratamente. Sono entrambi funzione dei 2 bit di stato all'istante attuale e del bit di ingresso

Sintesi della funzione stato prossimo della FSM della Vendor Machine

X_1X_0 \ I	I = 0	I = 1	Y
0 0	00	01	0
0 1	01	10	0
1 0	10	11	0
1 1	00	01	1

X è su 2 cifre => 2 bit: X_1 e X_0

Devo sintetizzare:

$X_1^{t+1} = f_1(X_0^t, X_1^t, I)$

$X_0^{t+1} = f_0(X_0^t, X_1^t, I)$

$X_1\ X_0\ I$	$X^*_1\ X^*_0$
0 0 0	0 0
0 0 1	0 1
0 1 0	0 1
0 1 1	1 0
1 0 0	1 0
1 0 1	1 1
1 1 0	0 0
1 1 1	0 1

X*

I = {0, 1}

Y = {0, 1}

X = {00, 01, 10, 11}

$X^* = f(X,I) \Rightarrow \quad X^*_0 = I\ (X_1 + \overline{X}_1\ \overline{X}_0) + \overline{I}\ \overline{X}_1 X_0 =$

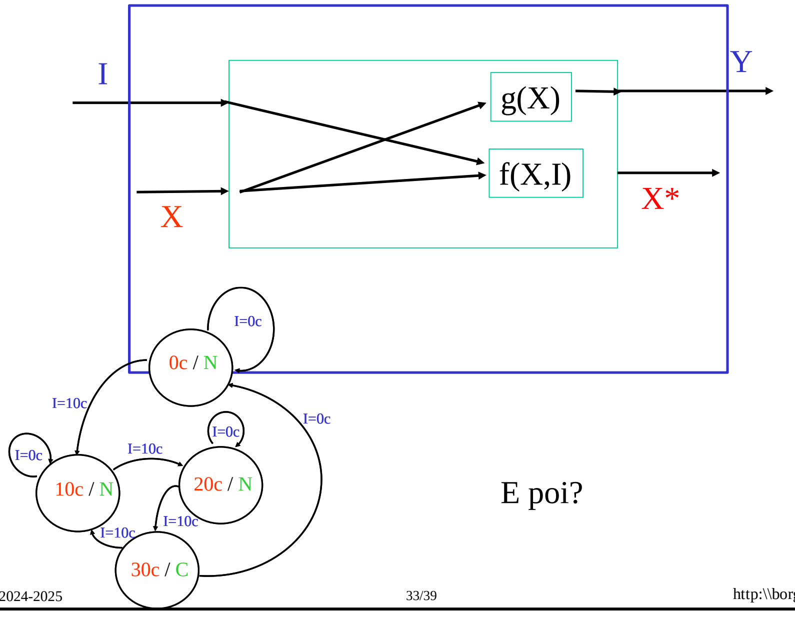
$\underline{\ }I(X_1 + X_0) + I\ X_1X_0$

$X^*_1 = X_1\ X_0 + X_1\ X_0\ I$

$Y = g(X) = X_1\ X_0$

I 2 bit di stato prossimo vengono sintetizzati separatamente. Sono entrambi funzione dei 2 bit di stato all'istante attuale e del bit di ingresso

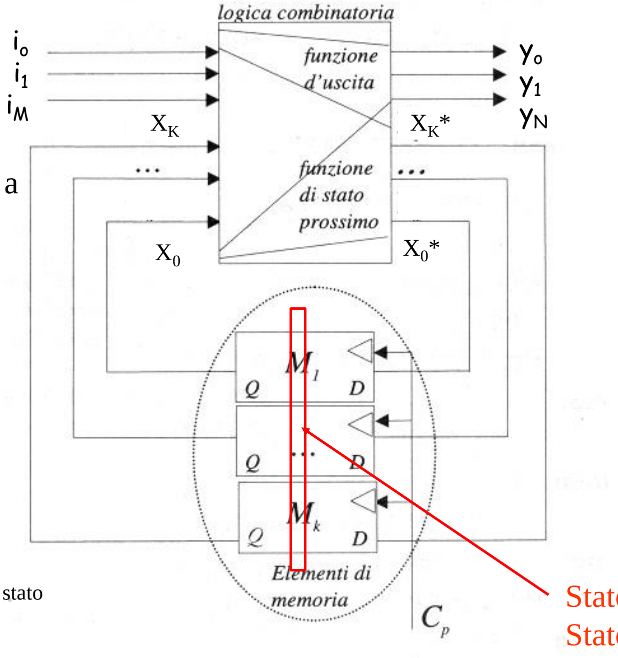
La situazione



E poi?

Macchina di Huffman

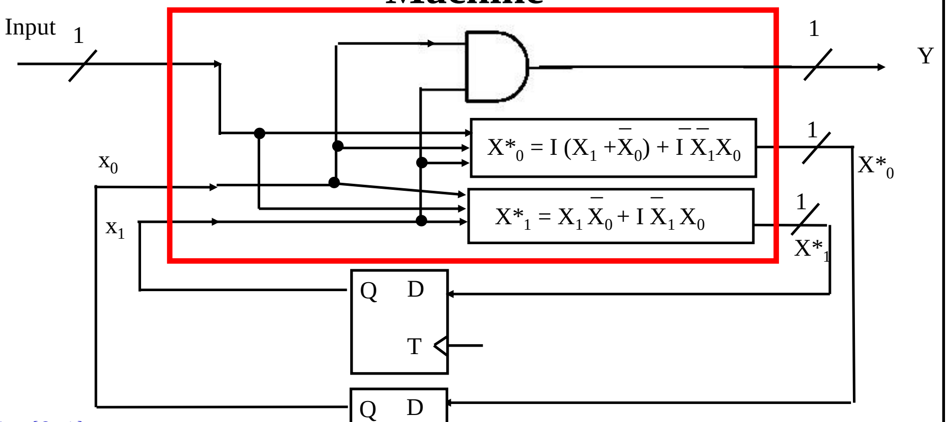
Macchina a stati finiti binaria



M ingressi
K variabili di stato
N uscite

Stato prossimo -> Stato presente

Sintesi del circuito della FSM della Vendor Machine



$I = \{0, 1\}$

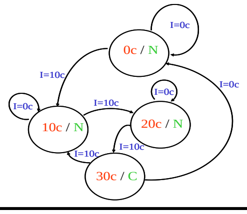
$Y = \{0, 1\}$

$X = \{00, 01, 10, 11\}$

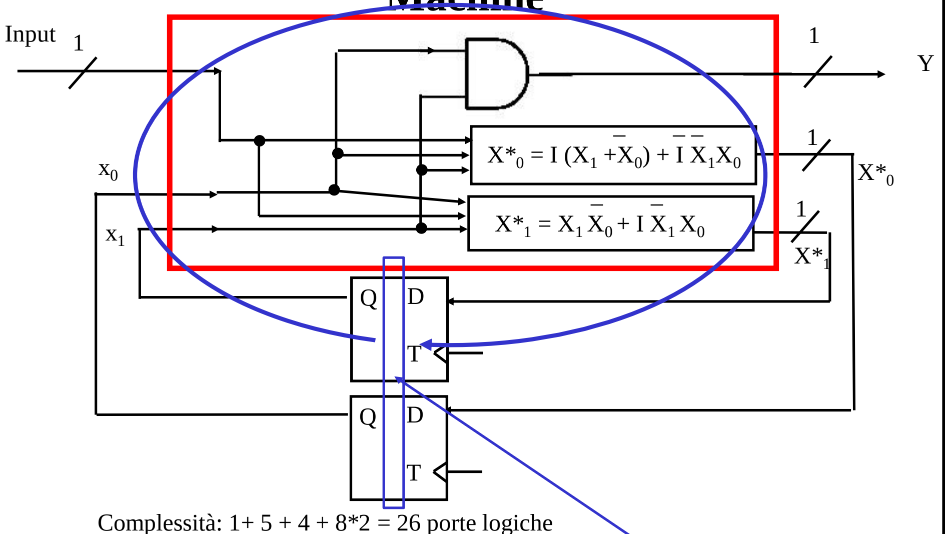
$X^* = f(X,I) \Rightarrow \quad X^*_0 = I(\overline{X_1} + \overline{X_0}) + \overline{I}\,\overline{X_1}X_0$

$X^*_1 = X_1 X_0 + X_1 X_0 I$

$Y = g(X) = X_1 X_0$



Sintesi del circuito della FSM della Vendor Machine



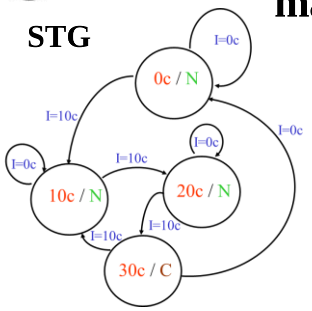
Complessità: 1+ 5 + 4 + 8*2 = 26 porte logiche

Stato prossimo -> Stato presente

Cammino critico: 3 + 2*3 = 9

I passi per la progettazione di una macchina di Huffman

STG



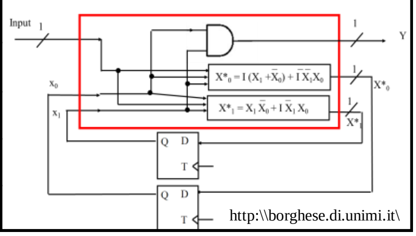
I / X	0c	10c	Y
0c	**0c**	**10c**	**Nulla**
10c	**10c**	**20c**	**Nulla**
20c	**20c**	**30c**	**Nulla**
30c	**0c**	**10c**	**Caffè**

STT

STT codificata

I / X - X_1X_0	0c (0)	10c (1)	Y
0c (00)	**0c (00)**	**10c (01)**	**Nulla (0)**
10c (01)	**10c (01)**	**20c (10)**	**Nulla (0)**
20c (10)	**20c (10)**	**30c (11)**	**Nulla (0)**
30c (11)	**0c (00)**	**10c (01)**	**Caffè (1)**

Macchina di Huffman



Una vendor machine più completa.

Monete diverse dai 10c.

Scelta di bevande diverse.

Bevande diversi con costi diversi.

Periodo di refrattarietà nella quale non si possono inserire monete (periodo di preparazione del caffè).

.....

Sommario

Macchine a stati finiti.

Esempio: sintesi di un controllore per venditore di bibite.