Modellare con le Finite State Machines
Angelo Gargantini

Testing e verifica del software
2025

Notazione della specifica

Si possono usare diverse notazioni per specificare il sistema
Esempi

- UML: macchini di stati/diagrammi di interazione
- ASM: ...
- Simulink: ...

Noi useremo le FSM (di Mealy)

- **Macchine a stati finiti (FSM)**

Macchine a stati finiti con output

Una FSM (S, I, δ) con output è:

- una macchina di Mealy se è una FSM che produce un output per **ciascuna transizione**
- una macchina di Moore se è una FSM che produce un output per **ciascun stato**

Macchina di Mealy

Una macchina di Mealy è una tupla

(S, I, O, δ, λ)

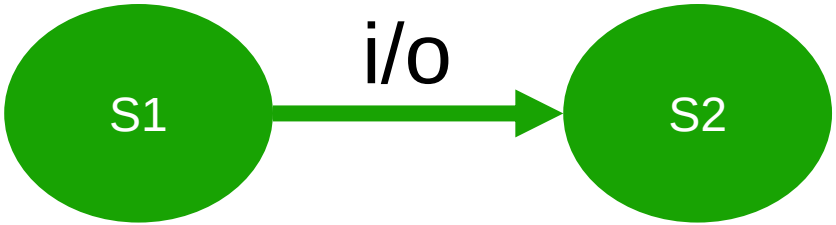
- S : insieme finito di stati
- I : insieme finito di eventi di input
- O : insieme finito di eventi di output
- δ : S x I ->S : funzione di transizione
- λ : S x I ->O : funzione di output

Spesso è anche fissato lo stato iniziale $s_0 \in S$

FSM di Mealy: rappresentazione grafica

Nei diagrammi che rappresentano una FSM di Mealy, ogni arco è etichettato da i/o

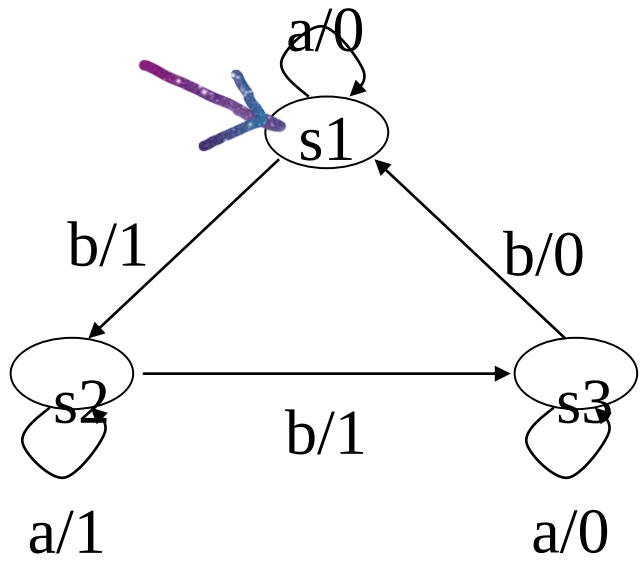
- **i** denota un simbolo di input ed è anche noto come evento di input
- **o** denota un simbolo di output ed è anche noto come evento di output



FSM di Mealy: rappresentazione grafica

Esempio:

S = {s1, s2, s3}
I = {a, b}
O = {0, 1}
δ = { (s1,a,s1), (s1,b,s2),
(s2,a,s2), (s2,b,s3),
(s3,a,s3), (s3,b,s1)}
λ = { (s1,a,0), (s1,b,1),
(s2,a,1), (s2,b,1),
(s3,a,0), (s3,b,0)}

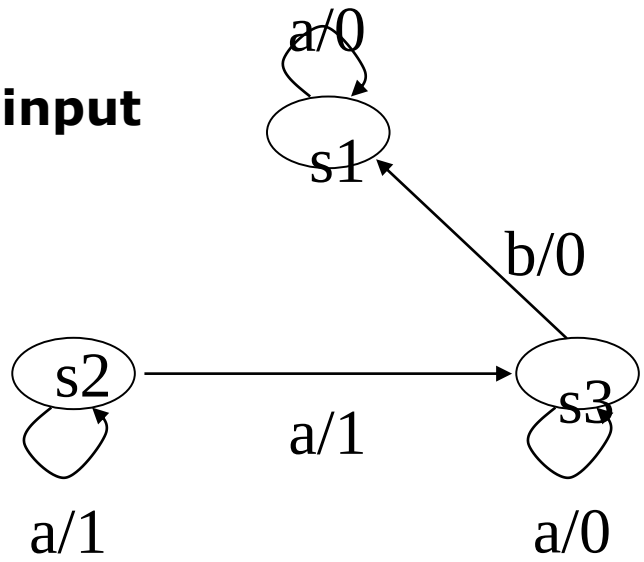


Errori nella definizione dalla mia macchina

1. non definisco cosa fare quando ho un certo input

- Arriva input b in S1

2. ho diverse transizioni con lo stesso input

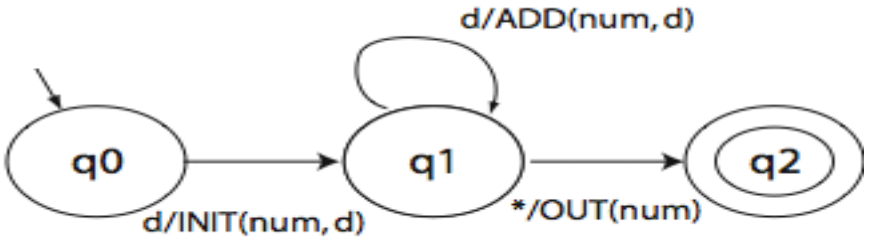


FSM: azioni sulle transizioni

Un evento di output può anche essere una azione della macchina

Esempio: macchina per convertire una sequenza di cifre decimali (d) in un intero (num)

- Azioni:
  - INIT: inizializza "num"
  - ADD: aggiunge la cifra "d" al valore corrente di "num"
  - OUT: output del numero



FSM di Mealy

È utile vedere una FSM di Mealy come la tupla (S, I, O, T)

- S : insieme finito di stati
- I : insieme finito di eventi di input
- O : insieme finito di eventi di output
- T : insieme finito di transizioni
  - Una transizione è una tupla (s, i, o, s’)
    - s : stato sorgente
    - i : evento di input
    - o : evento di output
    - s’ : stato target

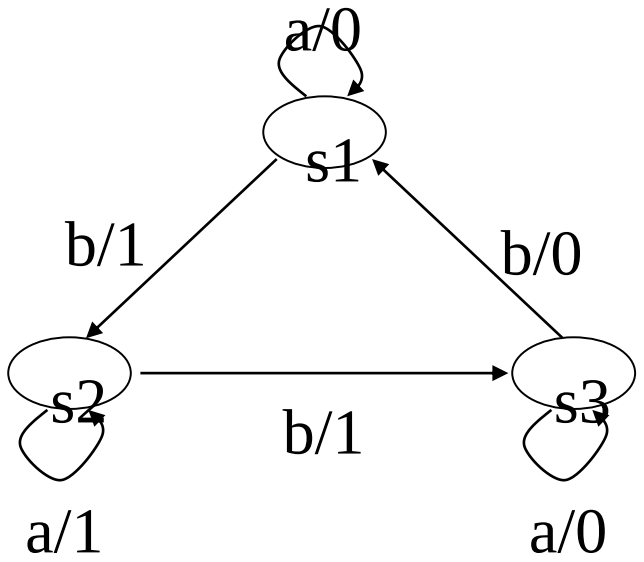
FSM di Mealy: esempio

S = {s1, s2, s3}

I = {a, b}

O = {0, 1}

T = {(s1,a,0,s1),
(s1,b,1,s2),
(s2,a,1,s2),
(s2,b,1,s3),
(s3,a,0,s3),
(s3,b,0,s1)}



Nota importante

Nel seguito per FSM intenderemo macchine di Mealy

Per esse utilizzeremo la definizione come la tupla (S, I, O, T)

Limiti delle FSM

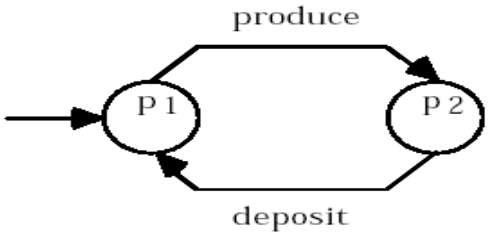
E’ possibile rappresentare solo un numero finito di stati

Esplosione del numero di stati:

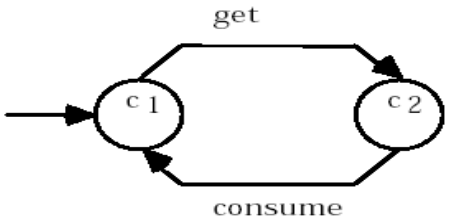
- dato un numero di FSM con k1, k2, … kn stati ciascuna, la loro composizione è una FSM con k1 * k2 *… * kn stati
- tale crescita è esponenziale nel numero di FSM
- ci piacerebbe una crescita lineare, ossia k1 + k2 +… + kn stati

Esplosione degli stati: esempio

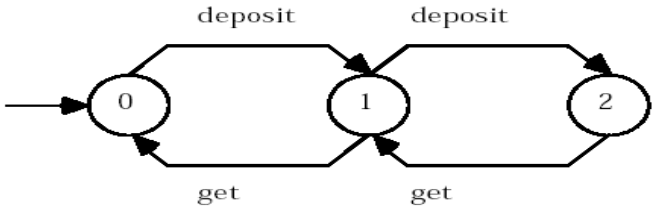
Produttore



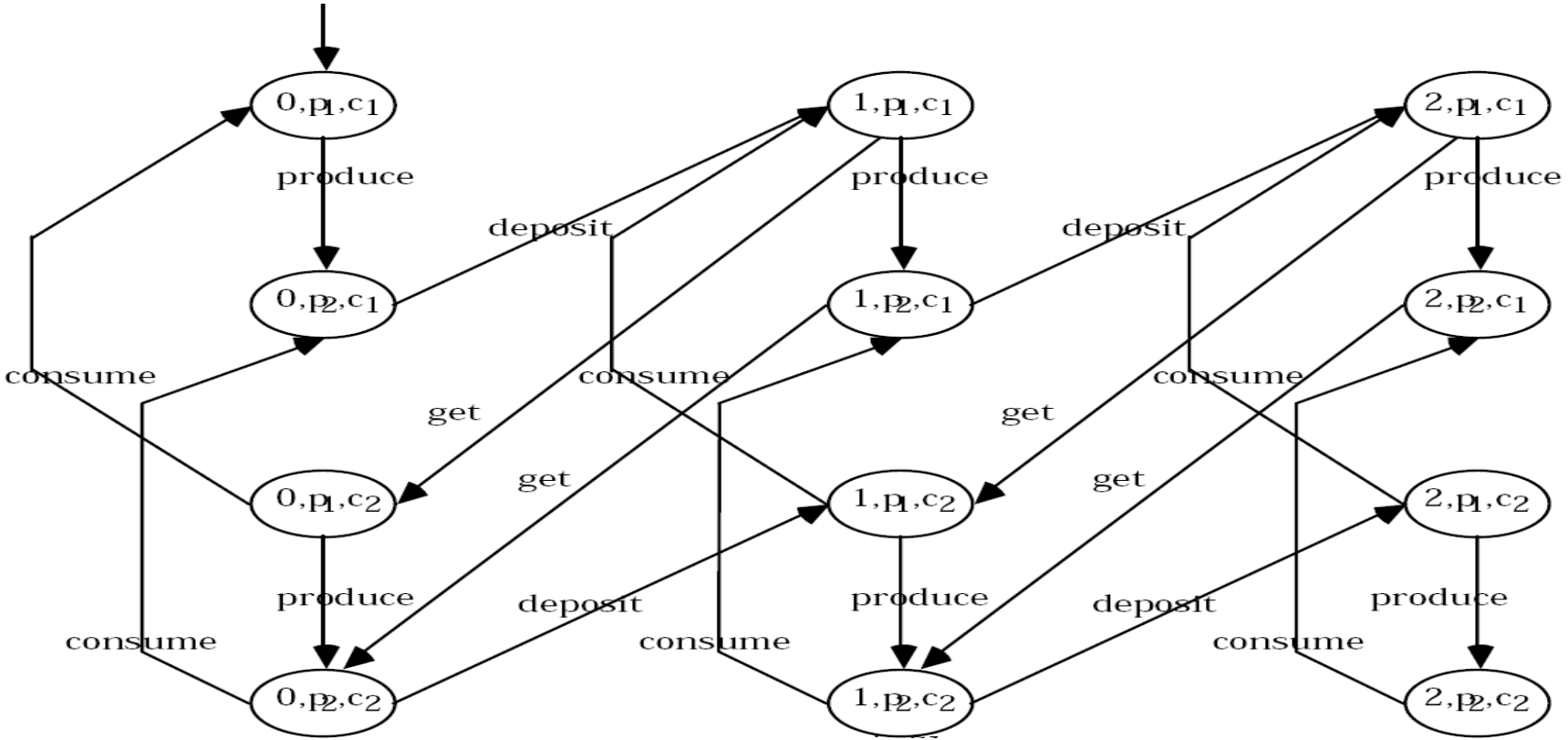
Consumatore



Magazzino



Combinando le FSM



Limiti delle FSM

Per superare i limiti di composizionalità delle FSM, sono state definite opportune estensioni (tra cui le Statecharts di UML) che sono dotate dei concetti di sottomacchina e permettono

- Composizione sequenziale
- Composizione parallela
- Modularità

Esercizio

Modellare con una FSM il comportamento di una sbarra che consente l'accesso/l'uscita di un parcheggio.

Per accedere al parcheggio, il semaforo deve essere verde. Mentre la sbarra è aperta, il semaforo è rosso.

Quando l'auto è entrata, la sbarra si chiude ed il semaforo ritorna verde.

Per uscire dal parcheggio, l'autista deve inserire il biglietto pre-pagato nel dispositivo che comanda la sbarra.

Quando l'auto è uscita la sbarra si chiude.

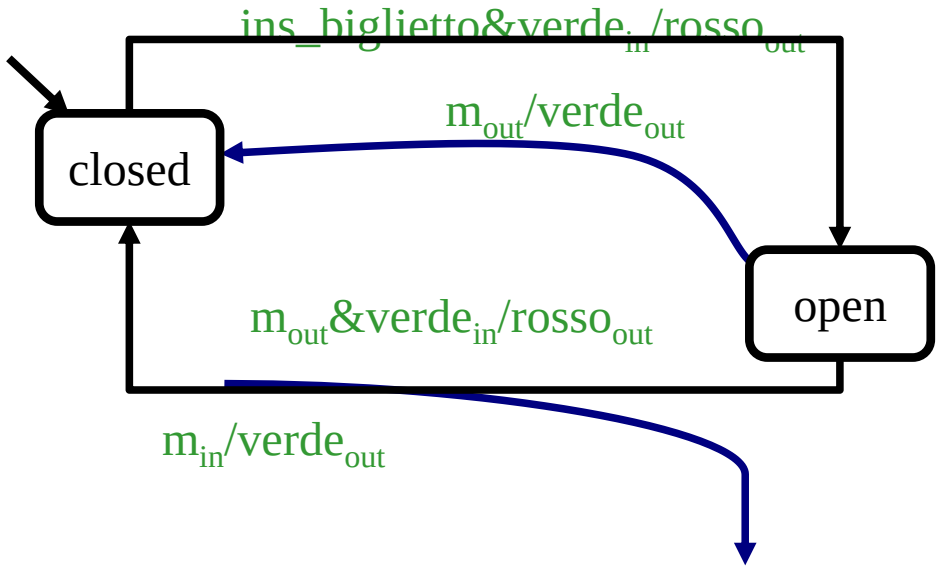
Soluzione esercizio

$verde_{in}$, $verde_{out}$, $rosso_{in}$, $rosso_{out}$: colori semaforo

m_{in} : segnale di presenza dell'auto nel parcheggio

m_{out}: segnale di presenza dell'auto fuori il parcheggio

ins_biglietto: segnale di inserimento biglietto



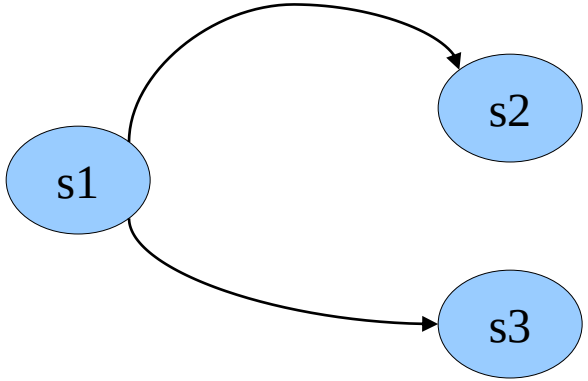
In sintesi

- Abbiamo visto:
  - le Macchine a Stati finiti sono estese con il concetto di output
  - la differenza tra macchine di Mealy e di Moore
  - i limiti delle macchine a stati finiti dovuti alla non-composizionalità dei modelli
- Ricordate che:
  - l'output può anche essere rappresentato da un'azione della macchina
- Infine:
  - da ora in avanti per FSM intenderemo una macchina di Mealy

- **Alcune estensioni**

Non determinismo

In caso la macchina sia non determinisca che fare?



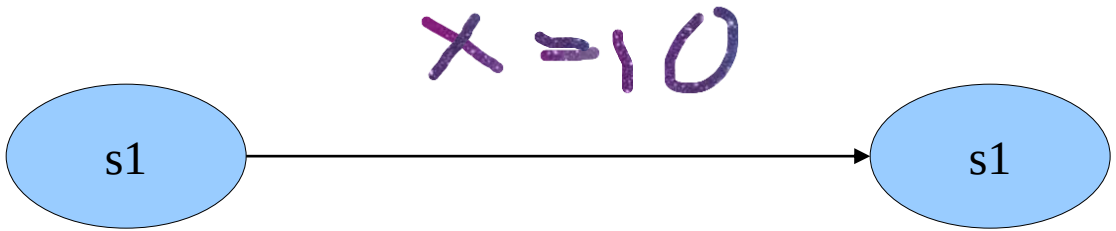
Alcune soluzioni possibili:

- Uso di runtime testing: genero mentre eseguo i casi di test
- Test non come sequenze ma come alberi

Aggiungo variabili

Spesso ho macchine con variabili

Agiungo variabili agli stati
+ guardie e assegnamenti
EFSM



Altri esempi: UML/ Abstract State machines/NuSMV

Esercizio FSM

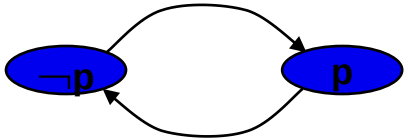
Scrivi una macchina a stati finiti con almeno 4 stati e due input e due output e trovane un transition tuor (euleriano). Introduci un difetto e scopri se il tuo test è in grado di scoprirlo.

Kripke structures

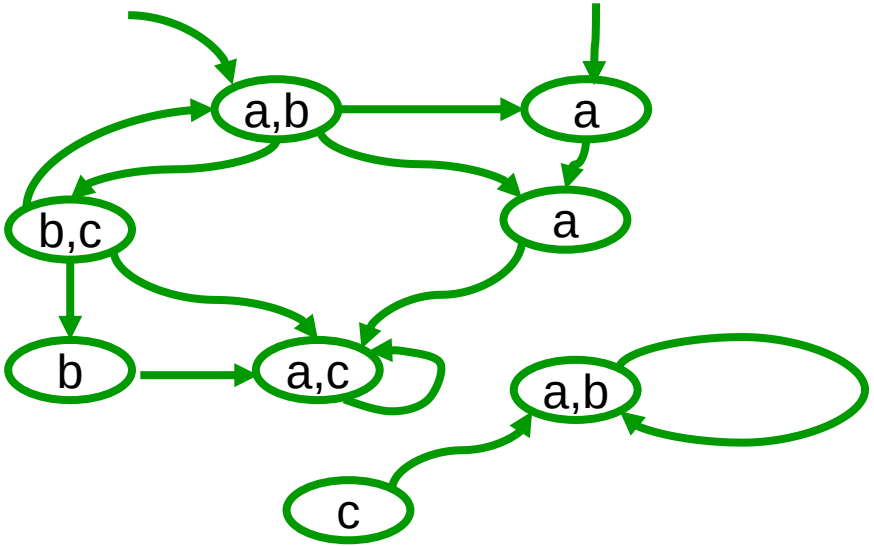
- A Kripke structure (*S,AP,R,L*) consists of
  - set of states S, atomic propositions AP
  - set of transitions $R \subseteq S' \times S$
  - labeling L = $S \rightarrow$ powerset(*AP*)

- Example: Kripke model of a program

```
repeat
    p := true;
    p := false;
end
```



Kripke structure / transition system



Esecuzione di una Kripke structure

- π=s0s1s2... is a run in M from s iff
- s = s0 and for every $i \geq 0$: $(si,si+1) \in R$