Macchine a stati finiti

Macchine a stati finiti di base

Introduzione

> Una **macchina a stati finiti** (**FSM**), è una *notazione formale* che permette la rappresentazione astratta del comportamento di un sistema

- Rigorosa definizione matematica
- Intuitiva rappresentazione grafica (**diagrammi di stato**)

Definizione

> Le FSM modellano le configurazioni istantanee di un sistema tramite **stati** (*nodi* del diagramma) e le operazioni del sistema tramite **transizioni** (*archi* del diagramma).
>
> Le operazioni possono ricevere **input** (FSM di base) ed eventualmente produrre un **output** (FSM di Mealy o Moore).

Non tutti i requisiti possono essere modellati con una FSM, ad esempio:

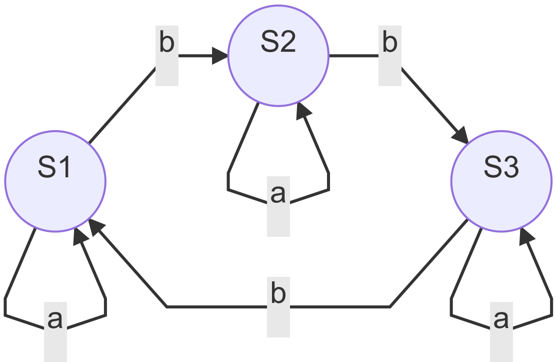
- requisiti *real-time*
- requisiti riguardanti *performance*
- requisiti riguardanti *tipi di computazioni*

Un'altra considerazione da fare è che vi possono essere FSM diverse per lo stesso sistema, dipende dal *livello di astrazione/dettaglio* che si considera.

La definizione più rigorosa di FSM di base è la seguente:

> - S : insieme finito di **stati**
> - I : insieme finito di **eventi di input**

- δ : **funzione di transizione** definita come
    - $\delta : S \times I \mapsto S$



In questo esempio abbiamo:

- $S = \{S1, S2, S3\}$
- $I = \{a, b\}$
- $\delta = \{(S1, a, S1), (S1, b, S2), (S2, a, S2), (S2, b, S3), (S3, a, S3), (S3, b, S1)\}$

Queste FSM sono le più elementari che definiamo, in quanto non hanno output. Sono per questo motivo molto limitate nell'esprimere sistemi complessi, perciò introduciamo adesso le *FSM con output*.

FSM con output

Le FSM con output sono principalmente due:

H2
- **Macchina di Mealy**
- **Macchina di Moore**

Noi studieremo e useremo per la modellazione la prima, la **macchina di Mealy**.

Macchina di Mealy

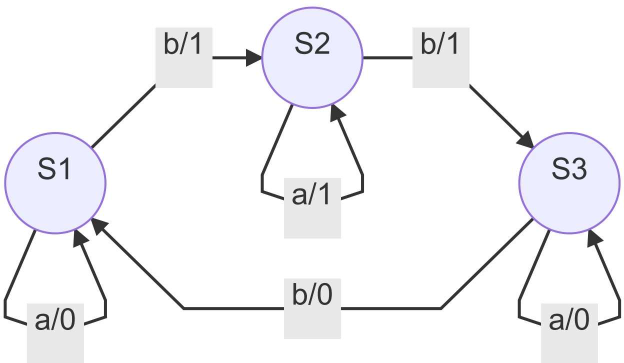
> Una FSM (S, I, δ) con output è una **macchina di Mealy** se produce un output per *ciascuna **transizione***

Formalmente abbiamo:

> Una **macchina di Mealy** è una tupla $(S, I, O, \delta, \lambda)$ dove:
>
> - S : insieme finito di **stati**
> - I : insieme finito di **eventi di input**
> - O : insieme finito di **eventi di output**
> - δ : **funzione di transizione** definita come
>   - $\delta : S \times I \mapsto S$
> - λ : **funzione di output** definita come
>   - $\lambda : S \times I \mapsto O$
>
> Spesso è fissato anche uno **stato iniziale** $S_0 \in S$

I diagrammi di stato di una FSM di Mealy hanno per ogni arco un'etichetta **i**/**o**, dove:

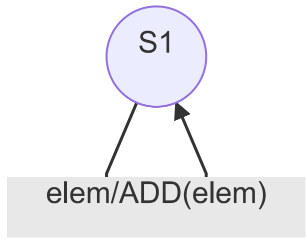
- **i** : denota un **evento di input**
- **o** : denota un **evento di output**



Abbiamo:

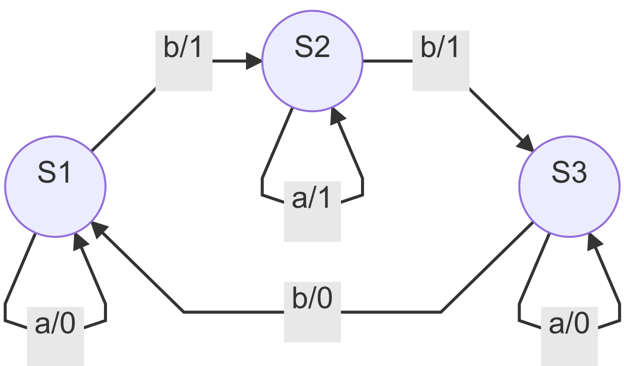
- $S = \{S1, S2, S3\}$
- $I = \{a, b\}$
- $O = \{0, 1\}$
- $\delta = \{(S1, a, S1), (S1, b, S2), (S2, a, S2), (S2, b, S3), \\ (S3, a, S3), (S3, b, S1)\}$
- $\lambda = \{(S1, a, 0), (S1, b, 1), (S2, a, 1), (S2, b, 1), (S3, a, 0), (S3, b, 0)\}$

Gli eventi di output delle macchine di Mealy possono anche essere **azioni della macchina**, cioè delle *funzioni che producono un output*. Ad esempio:



È possibile "compattare" la definizione di FSM di Mealy nel seguente modo, che sarà quello che useremo da ora in poi:

> Una **macchina di Mealy** è una tupla (S, I, O, T) dove:
>
> - 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') dove:
>   - s : **stato sorgente**
>   - i : **evento di input**
>   - o : **evento di output**
>   - s' : **stato di arrivo**



Abbiamo, con la nuova definizione, i seguenti oggetti:

- $S = \{S1, S2, S3\}$
- $I = \{a, b\}$

- $O = \{0, 1\}$
- $T = \{(S1, a, 0, S1), (S1, b, 1, S2), (S2, a, 1, S2), (S2, b, 1, S3),$
  $(S3, a, 0, S3), (S3, b, 0, S1)\}$

Macchine a stati finiti estese

Introduzione

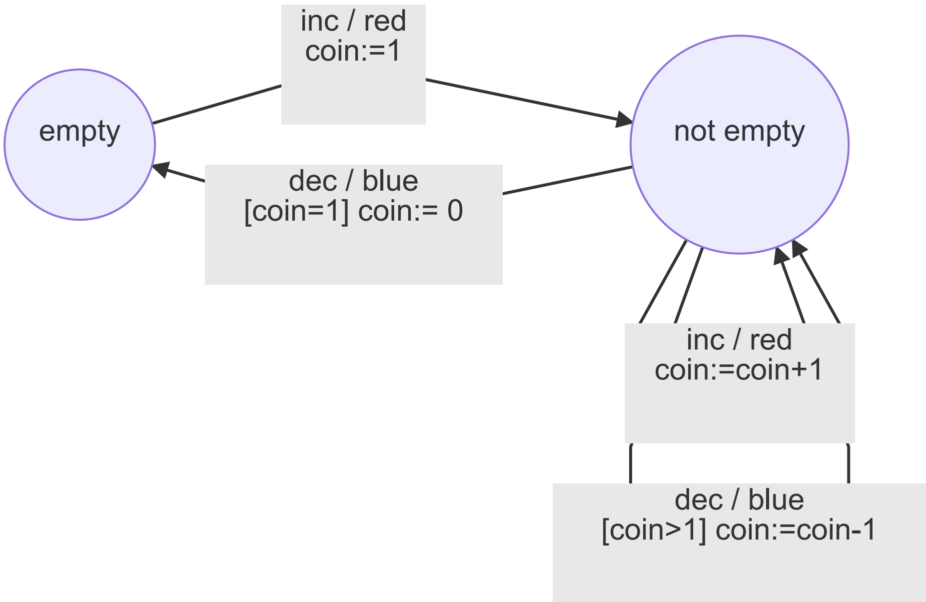
La FSM viste finora hanno un limite dovuto al fatto che non viene modellata la memoria della macchina ma solo il suo stato. Con le FSM di base l'unico modo è rappresentare uno stato per ogni diverso stato di memoria, ma chiaramente in caso di sistemi complessi questo diviene infattibile.
È possibile potenziare ancora di più la notazione delle FSM di Mealy, in modo tale che sia possibile rappresentare la memoria della macchina.

In particolare, avremo le *Extended Finite State Machines* (***EFSM***), che estendono le FSM con il concetto di **variabile**.

Definizione

> Una **EFSM** è una tupla (S, I, O, V, T) dove:
>
> - S : insieme finito di **stati**
> - I : insieme finito di **eventi di input**
> - O : insieme finito di **eventi di output**
> - V : insieme finito di **variabili**
> - T : insieme finito di **transizioni**
>   - Una **transizione** è una tupla (s, i, o, g, a, s') dove:
>   - s : **stato sorgente**
>   - i : **evento di input**
>   - o : **evento di output**
>   - g : *predicato delle variabili in* V, detto **guardia**
>   - a : *assegnamento di una variabile in* V, detto **azione**
>   - s' : **stato di arrivo**

- inserendo monete (evento `inc`) il valore di `coin` viene incrementato
- emettendo monete (evento `dec`) il valore di `coin` viene decrementato
- la luce diventa `red` quando si inseriscono monete, diventa `blue` quando si richiedono monete
- la macchina non restituisce monete quando è vuota



Formalmente:

- $S = \{\text{empty, not empty}\}$
- $I = \{\text{inc, dec}\}$
- $O = \{\text{red, blue}\}$
- $V = \{\text{coin}\}$
- $T = \{(\text{empty, inc, red, coin:=1, not-empty}),$
  $(\text{not-empty, dec, blue, coin=1, coin:=0, empty}),$
  $(\text{not-empty, inc, red, coin:=coin+1, not-empty}),$
  $(\text{not-empty, dec, blue, coin>1, coin:=coin-1, not-empty})\}$

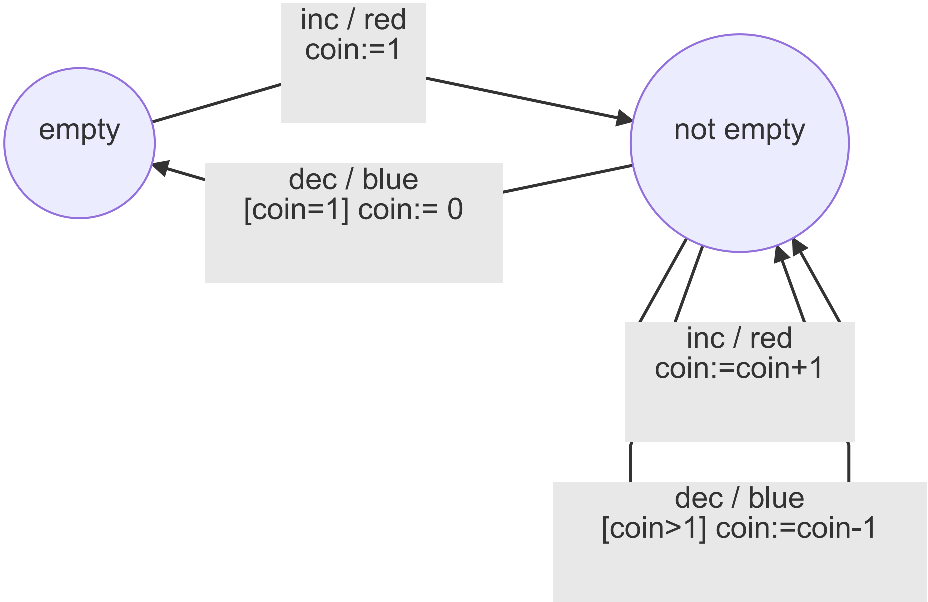
Stato globale

H4

> Per una EFSM (S, I, O, V, T), una coppia (s, σ) è detta **stato globale** se:
>
> - s è uno stato

- σ è una valutazione su V

Tornando all'esempio del salvadanaio:



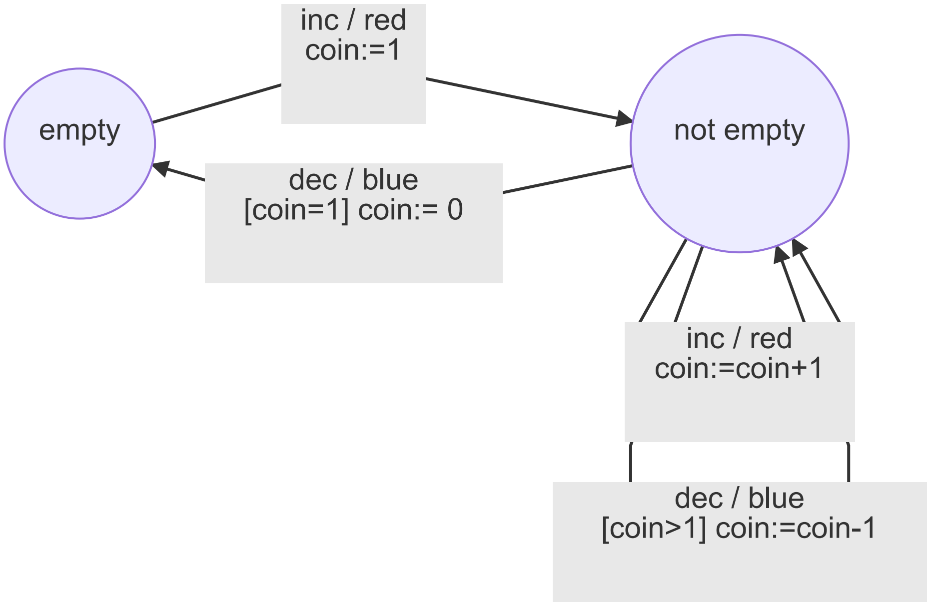
Avremo i seguenti possibili **stati globali**:

- (empty, coin=0)
- $(\text{not-empty, coin=1})$
- $(\text{not-empty, coin=2})$
- ...

Transizione globale

> Per una EFSM (S, I, O, V, T), una tupla $((s, \sigma), i, o, (s', \sigma'))$ è detta **transizione globale** se:
>
> - esiste una transizione (s, i, o, g, a, s') tale che:
>   - σ soddisfa la guardia g
>   - $\sigma'(v) = \sigma(\exp)$ dove l'azione a è $v := \exp$

Tornando all'esempio del salvadanaio:



Avremo le seguenti **transizioni globali**:

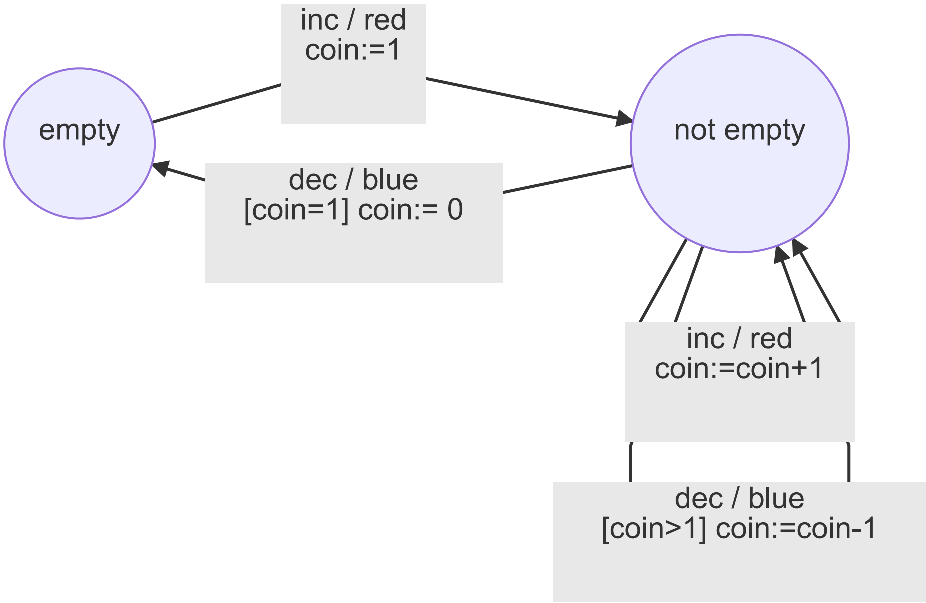
- $((\text{empty, coin}=0), \text{inc, red}, (\text{not-empty, coin}=1))$
- $((\text{not-empty, coin}=1), \text{dec, blue}, (\text{empty, coin}=0))$
- $((\text{not-empty, coin}=1), \text{inc, red}, (\text{not-empty, coin}=2))$
- ...

Grafo di raggiungibilità

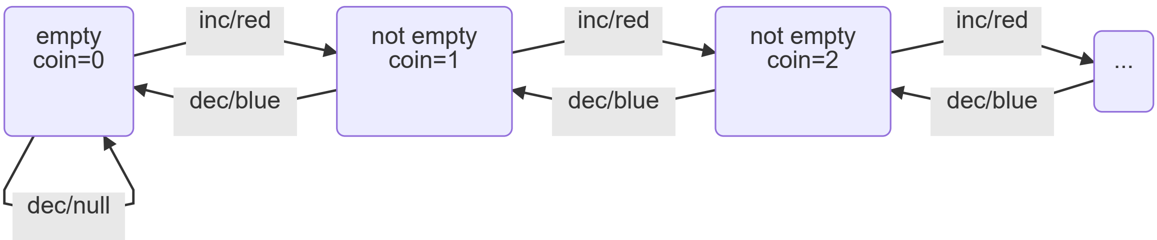
H4

> Il **grafo di raggiungibilità** di una EFSM (S, I, O, V, T) è un *grafo orientato* in cui:
>
> - i nodi sono gli **stati globali**
> - gli archi sono le **transizioni globali**

Se le variabili hanno un range finito, il **grafo di raggiungibilità** è una FSM.



il suo **grafo di raggiungibilità** sarà il seguente:



In questo esempio il grafo non è una FSM poiché non abbiamo posto dei vincoli alla variabile `coin`. Se ci fossero, sarebbe un FSM.

Limiti delle FSM

Tutte le FSM che abbiamo visto finora hanno dei limiti legati al caso in cui noi volessimo *comporre* diverse "macchine semplici" in sistemi più complessi.

Infatti, in generale vale che date n FSM con $k_1, k_2, \ldots, k_n$ stati ciascuna, la loro **composizione** è una FSM con $k1 \times k2 \times \cdots \times k_n$ stati totali. Tale crescita è *esponenziale* rispetto al numero si FSM di base.

Sono state quindi definite opportune estensioni delle FSM che sono dotate dei concetti di **sottomacchina** e che consentono:

- **composizione sequenziale**
- **composizione parallela**
- **modularità**

Tali estensioni sono:

- Macchine di comunicazione
- Macchine a stati UML