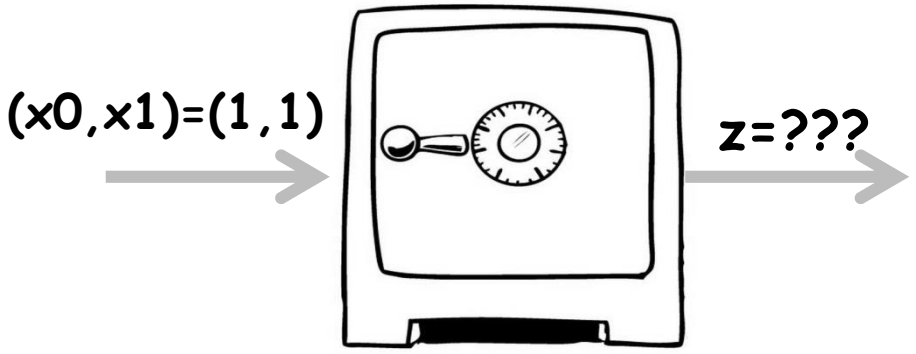
Capitolo 4
Reti Sequenziali

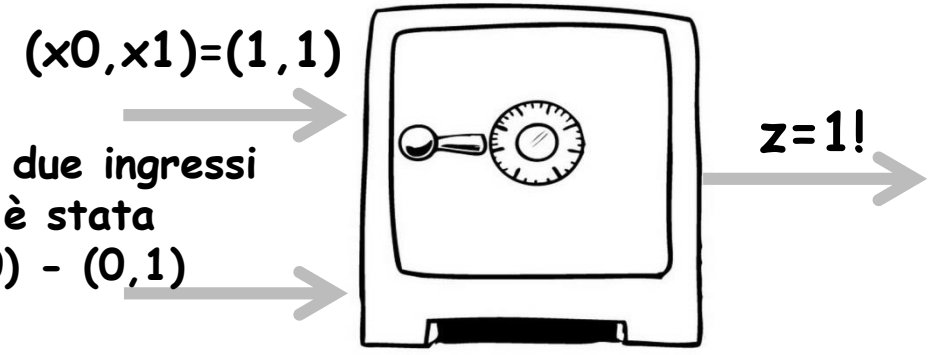
Reti Logiche T

Rete sequenziale

Es. **riconoscitore di sequenza:**
z=1 se e solo se la sequenza degli ingressi **(x0,x1)** è nell'ordine:
(0,0)-(0,1)-(1,1)



Gli ingressi non determinano univocamente l'uscita: è necessario portarsi dietro un *riassunto* della storia passata



Rete sequenziale

Definizione di «evento»:

- modifica di uno o più valori in ingresso
- scadere di un prefissato intervallo di tempo.

Indicati con $t_0, t_1, \ldots, t_{n-1}, t_n$ gli istanti in cui si sono verificati degli eventi, l'uscita u al generico istante t_n dipende

- dalla sequenza di ingresso $i(t_0) \Rightarrow i(t_1) \Rightarrow \ldots \Rightarrow i(t_{n-1}) \Rightarrow i(t_n)$
- dalla condizione iniziale della macchina $s(t_0)$.

$$u(t_n) = P\,(\, s(t_0), i(t_0), i(t_1), \ldots, i(t_{n-1}), i(t_n))$$

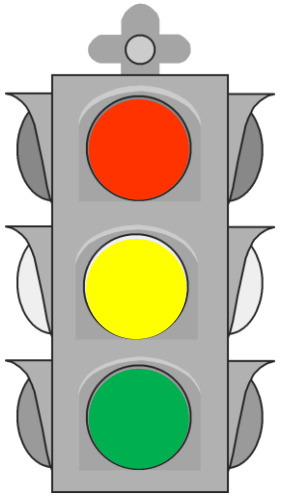
Una macchina che ha questo comportamento è detta

SEQUENZIALE

Rete sequenziale: lo stato iniziale

- Lo stato iniziale **$s(t_0)$** appartiene a un insieme discreto **S** di possibili stati della macchina
- Esso riassume tutta la storia della macchina precedente all'istante di tempo **t_0**

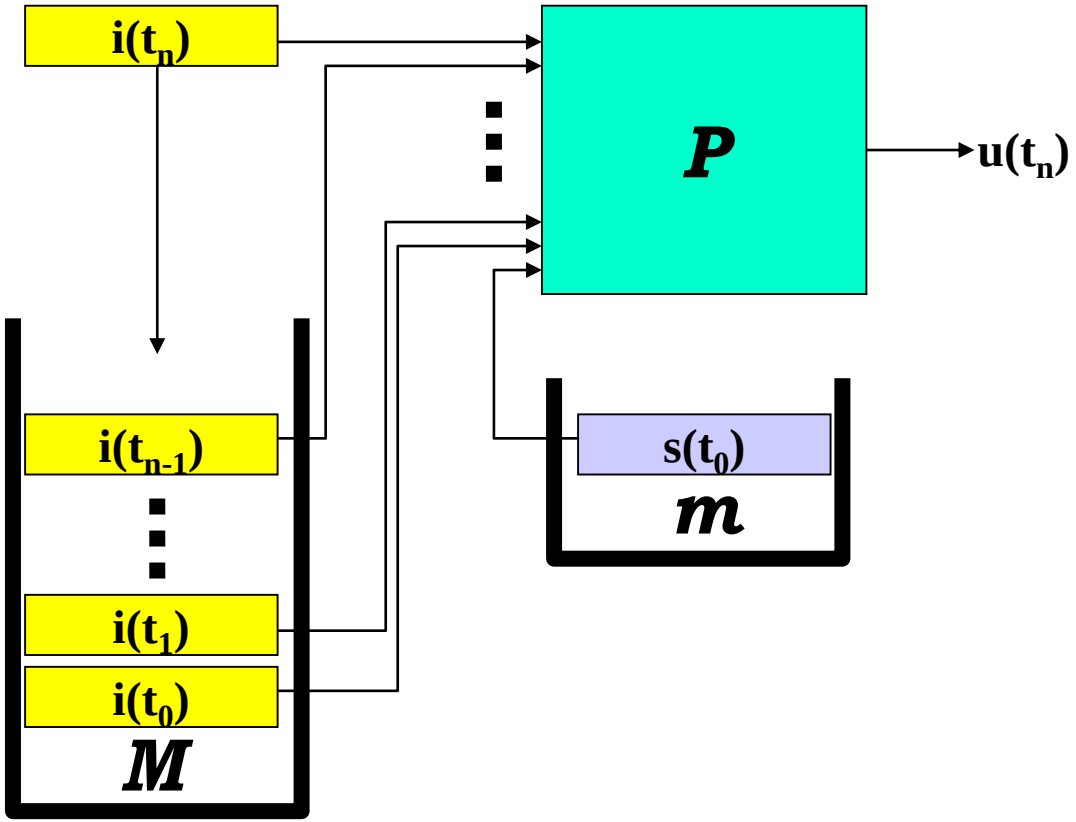
- **Esempio 1** : In un macchina che regola un **semaforo**,
  - Gli stati in cui la macchina si può trovare alla sua accensione sono 3: «rosso», «giallo», «verde»
  - la lampada attualmente accesa dipende non solo dalla temporizzazione associata a rosso, arancione e verde, ma anche dalla luce accesa nel momento in cui il semaforo è stato avviato.



- **Esempio 2**: Non basta caricare un orologio per avere l'ora esatta. L'ora indicata dipende infatti non solo dal n° di scatti che la molla ha dato alle lancette, ma anche dalla loro posizione iniziale.

Memoria e funzione

- Una rete sequenziale può essere impiegata solo quando l'uscita dipende da un numero **finito di eventi**
- non è infatti possibile disporre di una memoria M con capacità infinita e di una funzione P con infiniti ingressi.
- È necessario un *riassunto* della storia passata in un numero finito (e piccolo) di **stati**



Evoluzione del processo di elaborazione

- Si introduce **$s(t_n)$**, che rappresenta lo **stato corrente**, riassunto di $i(t_0),..,i(t_{n-1})$
- La funzione **F** determina l'uscita corrente $u(t_n)$ a partire dallo stato corrente $s(t_n)$ e dall'ingresso corrente $i(t_n)$
- La funzione ***G*** determina lo stato futuro $s(t_{n+1})$ a partire dall'ingresso corrente $i(t_n)$ e dallo stato presente $s(t_n)$

Funzione d'uscita: $S \times I \rightarrow U$

$$u(t_n) = F(s(t_n), i(t_n))$$

Funzione di aggiornamento dello stato: $S \times I \rightarrow S$

$$s(t_{n+1}) = G(s(t_n), i(t_n))$$

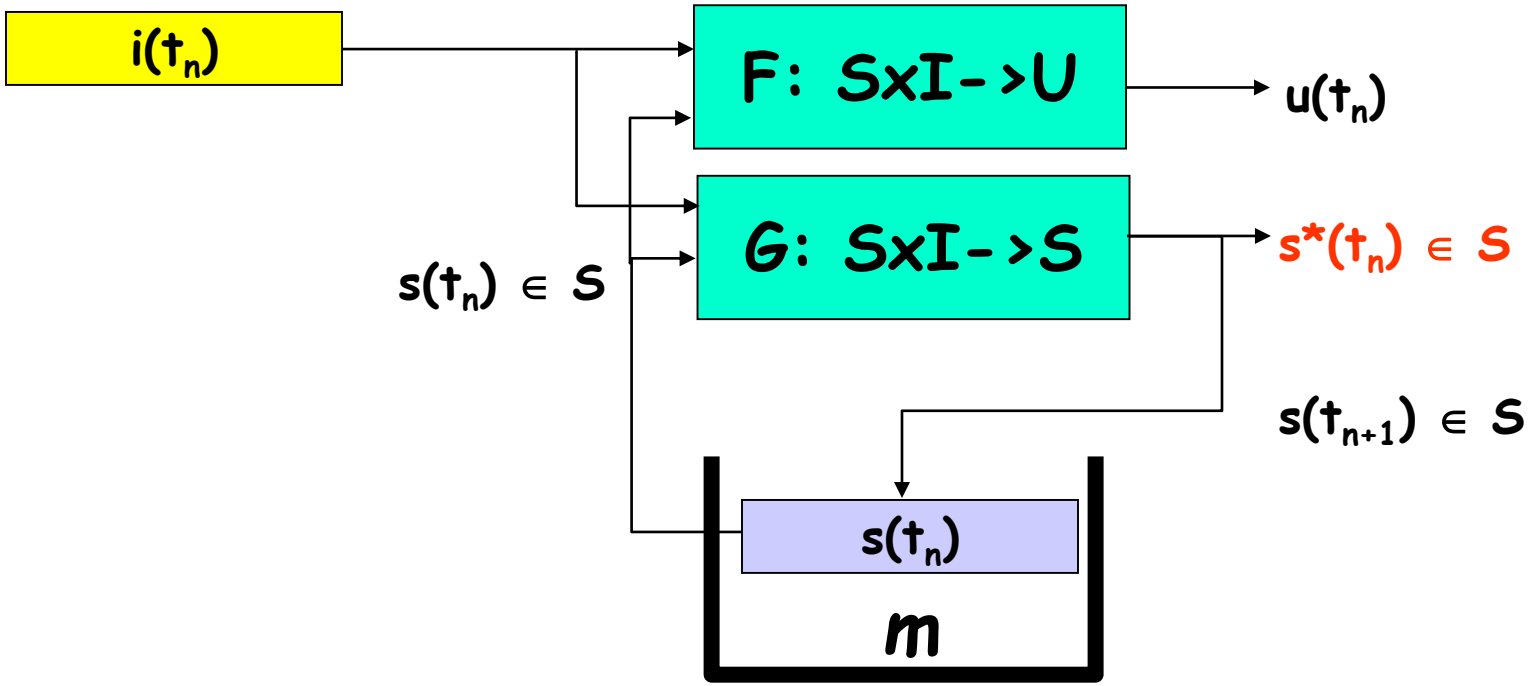
- Le funzioni F e G complessivamente realizzano la funzione P, richiedendo però solo la conoscenza dello **stato presente** e dell'**ingresso presente** (ovvero all'istante corrente t_n)

$$\begin{aligned} u(t_n) &= F(s(t_n), i(t_n)) \\ &= F(G(s(t_{n-1}), i(t_{n-1})), i(t_n)) \\ &\ldots \\ &= F(G\ldots(G(G(s(t_0), i(t_0)), i(t_1)), i(t_2)), \ldots), i(t_n)) \end{aligned}$$

$$u(t_n) = P\,(\, s(t_0),\ i(t_0),\ i(t_1),\ \ldots\ ,\ i(t_{n-1}),\ i(t_n)\,)$$

Stato interno presente e futuro

- La macchina impiega un insieme ***finito*** di stati interni per riassumere tutte le possibili sequenze d'ingresso.
- Le funzioni F e G dipendono solo da ingresso corrente e stato corrente -> si possono realizzare mediante due **reti combinatorie** (sfruttando le metodologie viste in precedenza)
- Più precisamente, la funzione G, tramite $i(t_n)$ e $s(t_n)$ calcola il nuovo riassunto della sequenza d'ingresso $s^*(t_n)$ (stato futuro), che viene depositato in memoria **m** (in **retroazione**)
- Il simbolo di stato futuro diventerà così lo stato presente al verificarsi dell'evento successivo (t_{n+1})



La FSM (Finite State Machine)

Automa a stati finiti:
Sistema matematico

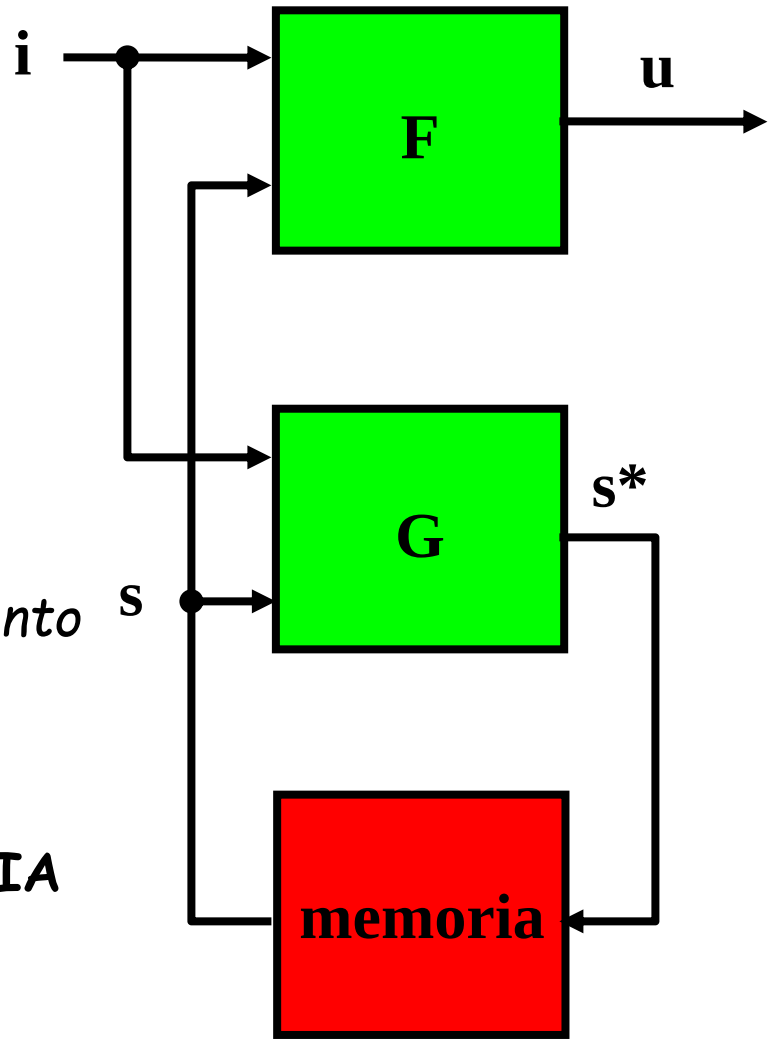
$\mathbf{M = \{I, U, S, F, G\}}$

formato da 3 INSIEMI

$\mathbf{I}$: $\{i_1, i_2, \ldots, i_n\}$ *alfabeto di ingresso*
$\mathbf{U}$: $\{u_1, u_2, \ldots, u_m\}$ *alfabeto di uscita*
$\mathbf{S}$: $\{s_1, s_2, \ldots, s_k\}$ *insieme degli stati*

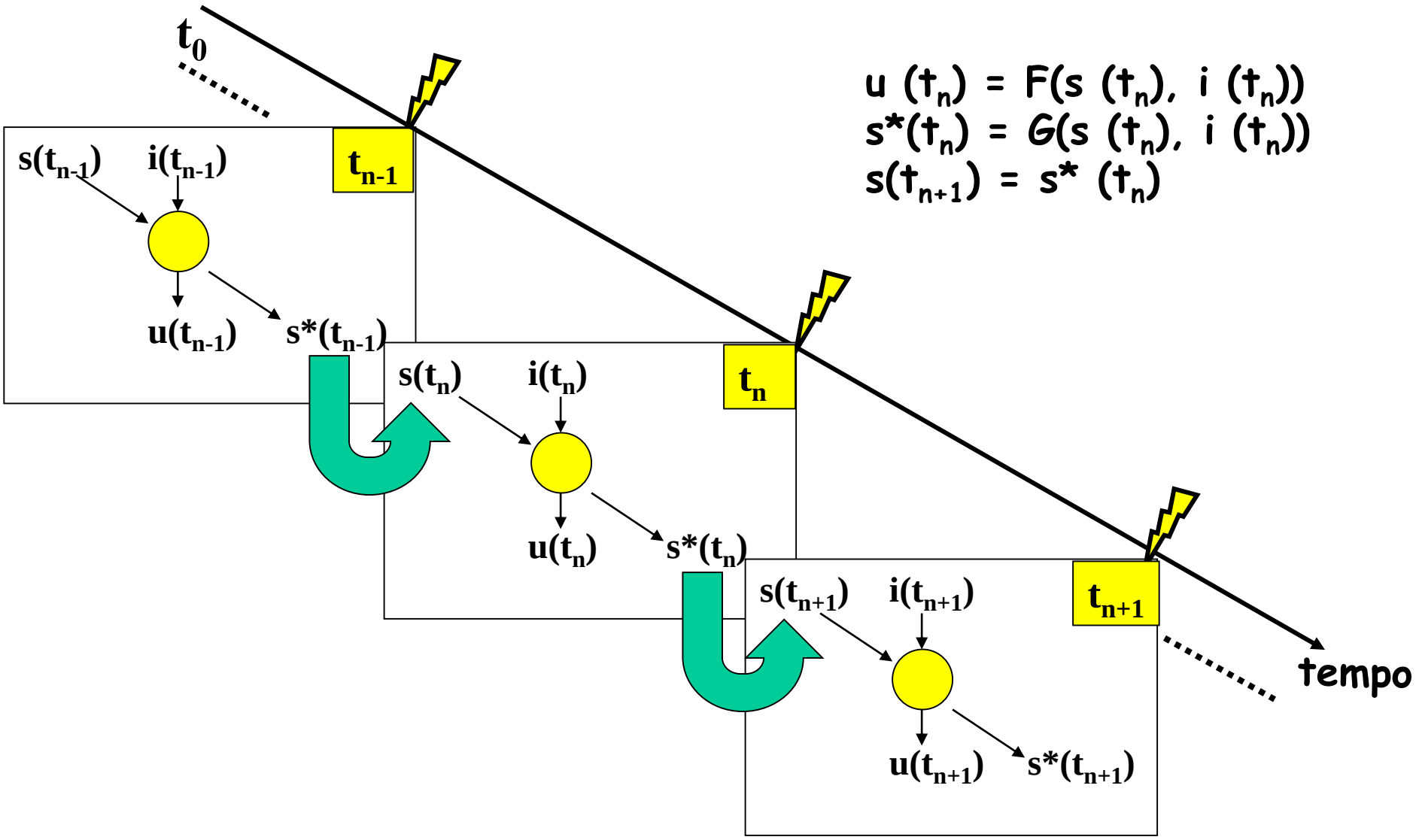
e da 2 FUNZIONI

$\mathbf{F : S \times I \rightarrow U}$ *funzione di uscita*
$\mathbf{G : S \times I \rightarrow S}$ *funzione di aggiornamento dello stato interno*



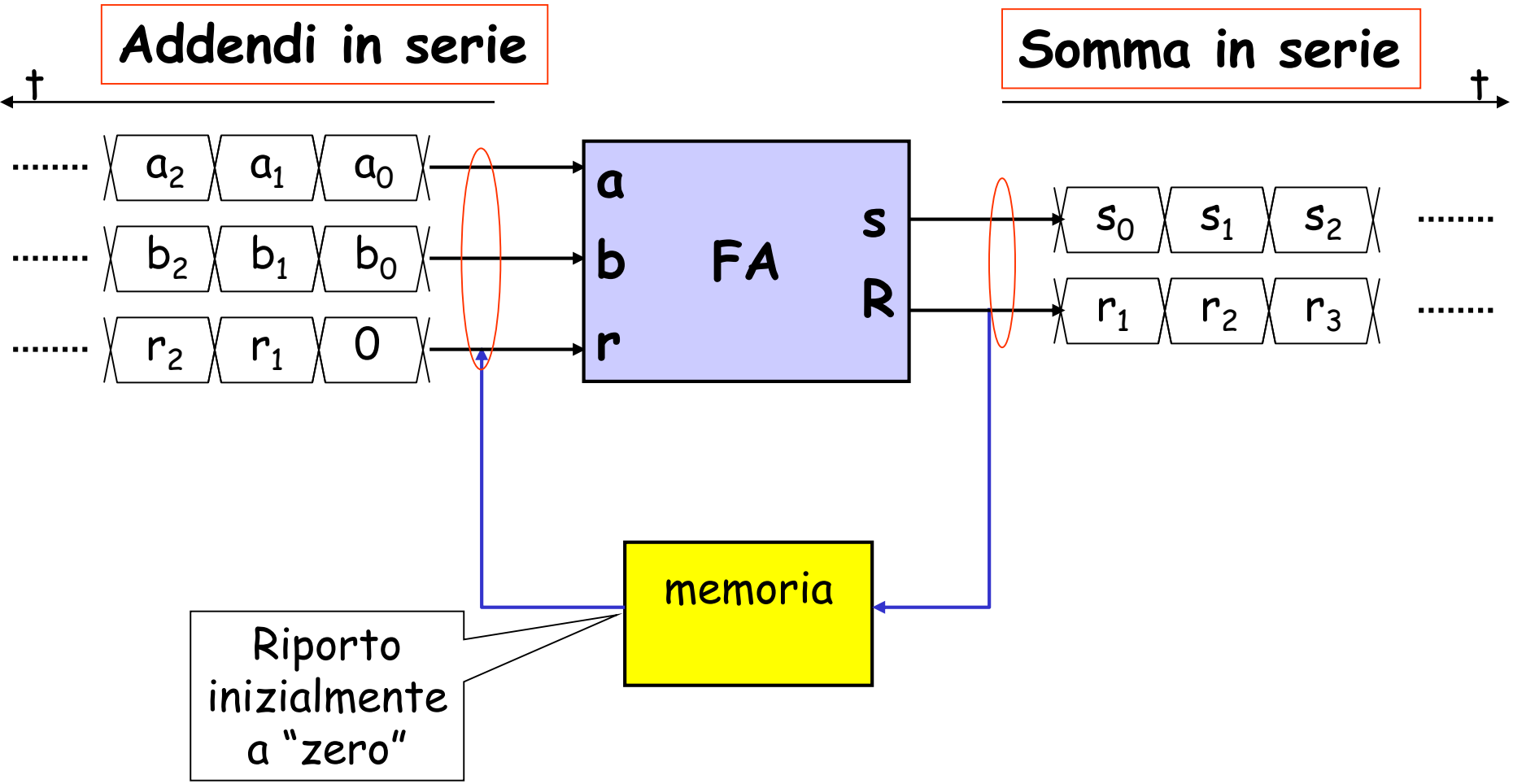
Nella realizzazione occorre una MEMORIA
che mantenga il "vecchio stato" s
fino a quando non è necessario
*sostituirlo con il "nuovo stato" s**

Evoluzione dell'elaborazione



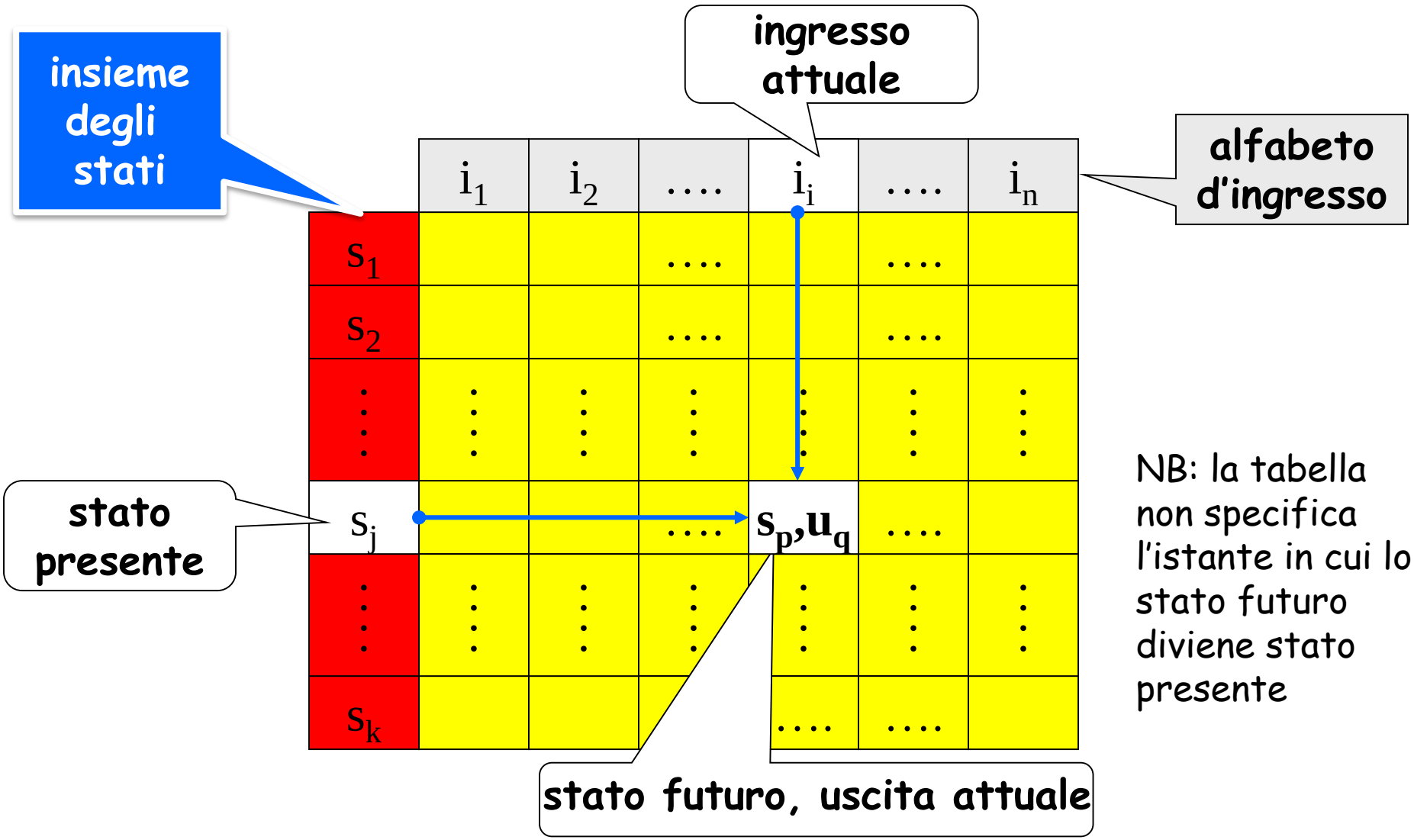
Esempio: addizione in serie

- I coefficienti dell'addizione (a,b,r) sono introdotti in sequenza a partire da quelli di minor peso
- Non è necessario memorizzarli tutti: il bit di riporto riassume ciò che è sufficiente sapere della storia passata per generare in sequenza le cifre del risultato (1 unico bit di stato)



Descrizione con tabella di flusso

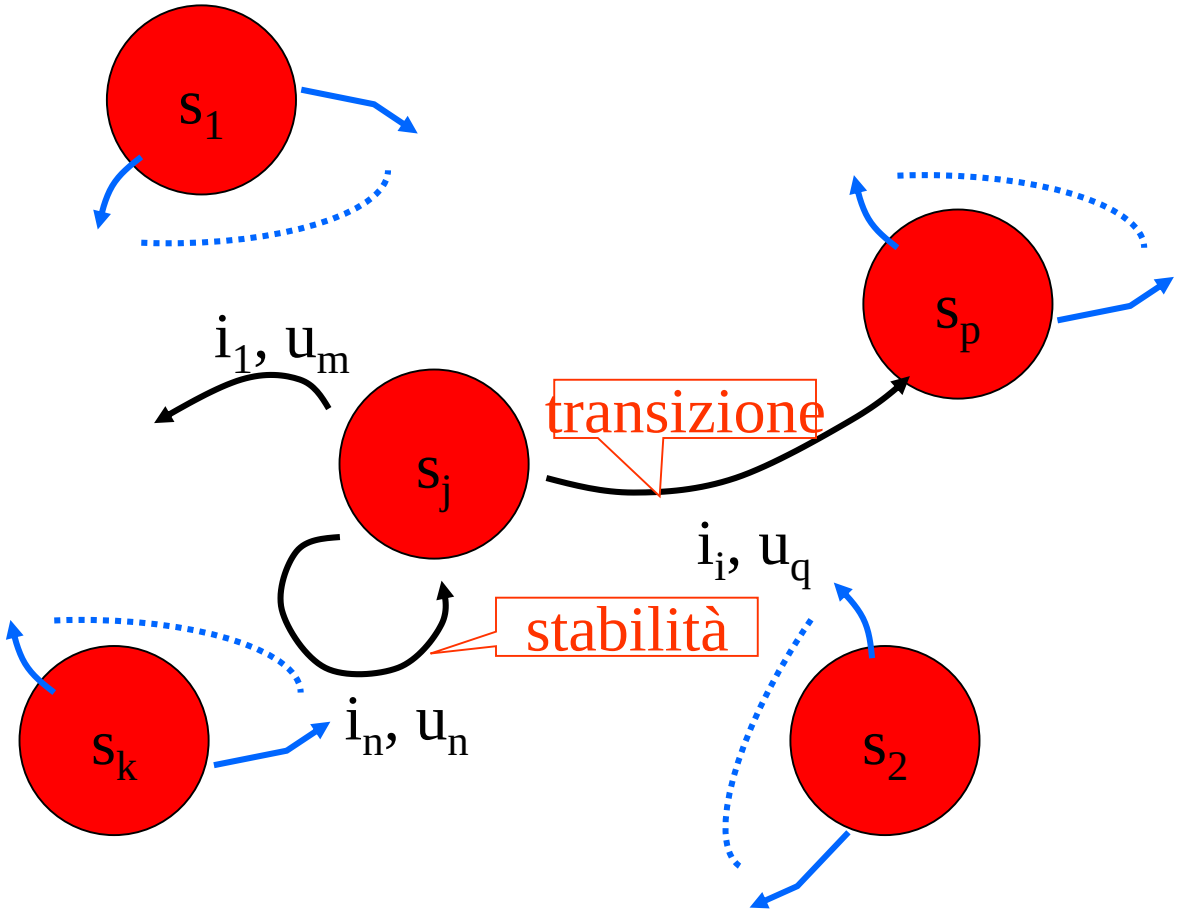
- **Tabella di flusso:** descrive il *comportamento* di una macchina a stati finiti
- In ogni casella vengono riportati i valori delle funzioni F,G per il rispettivo ingresso e stato presente



NB: la tabella non specifica l'istante in cui lo stato futuro diviene stato presente

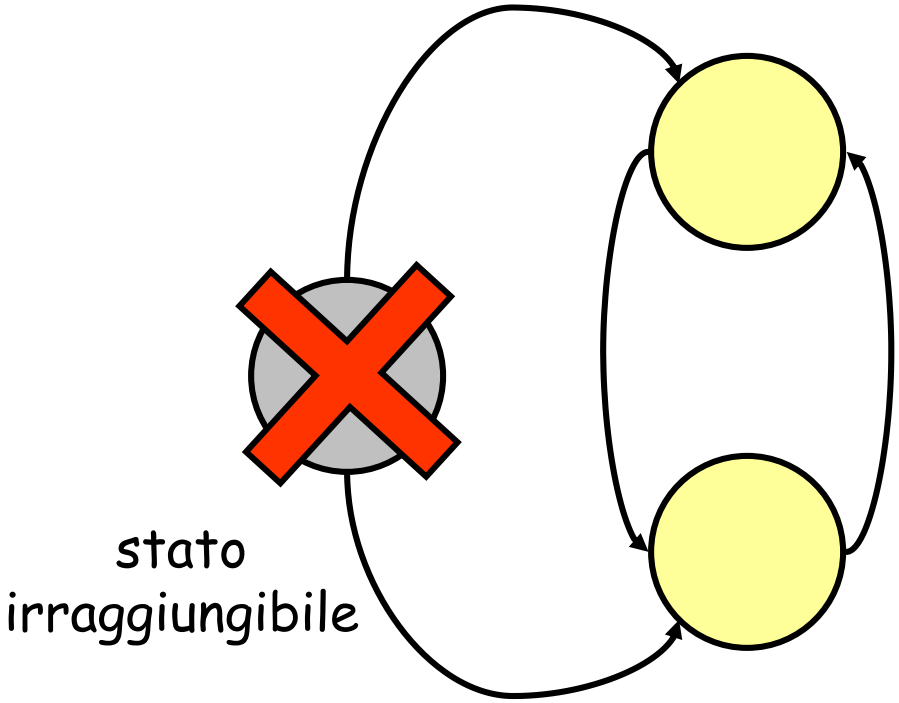
Descrizione con grafo degli stati

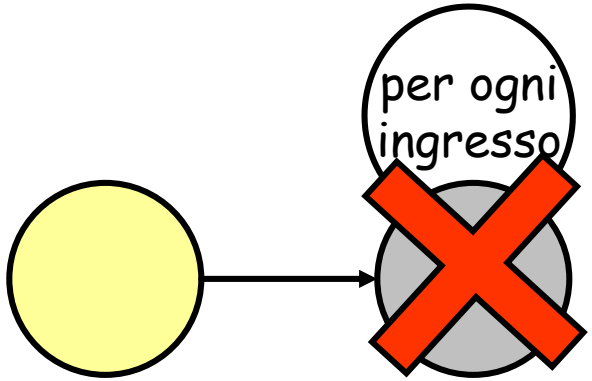
- Grafo (o diagramma) degli stati: grafo a rami orientati in cui ogni nodo rappresenta uno stato ed ogni ramo una transizione da stato presente a stato futuro
- Da ogni nodo devono uscire tanti rami quanti sono i simboli d'ingresso
- Stato stabile: ramo che si richiude sul nodo da cui diparte
- L'indicazione dell'uscita su ciascuna transizione definisce il **modello di Mealy**



Grafi strettamente connessi

- Per progettare macchine che non si fermano mai
  - occorre che esista sempre almeno un percorso per passare da un nodo arbitrariamente scelto a un altro (grafo **strettamente connesso**)
  - occorre evitare **stati iniziali** (dotati di sole frecce uscenti, irraggiungibili) o **stati finali** (dotati di sole frecce entranti, assorbenti)

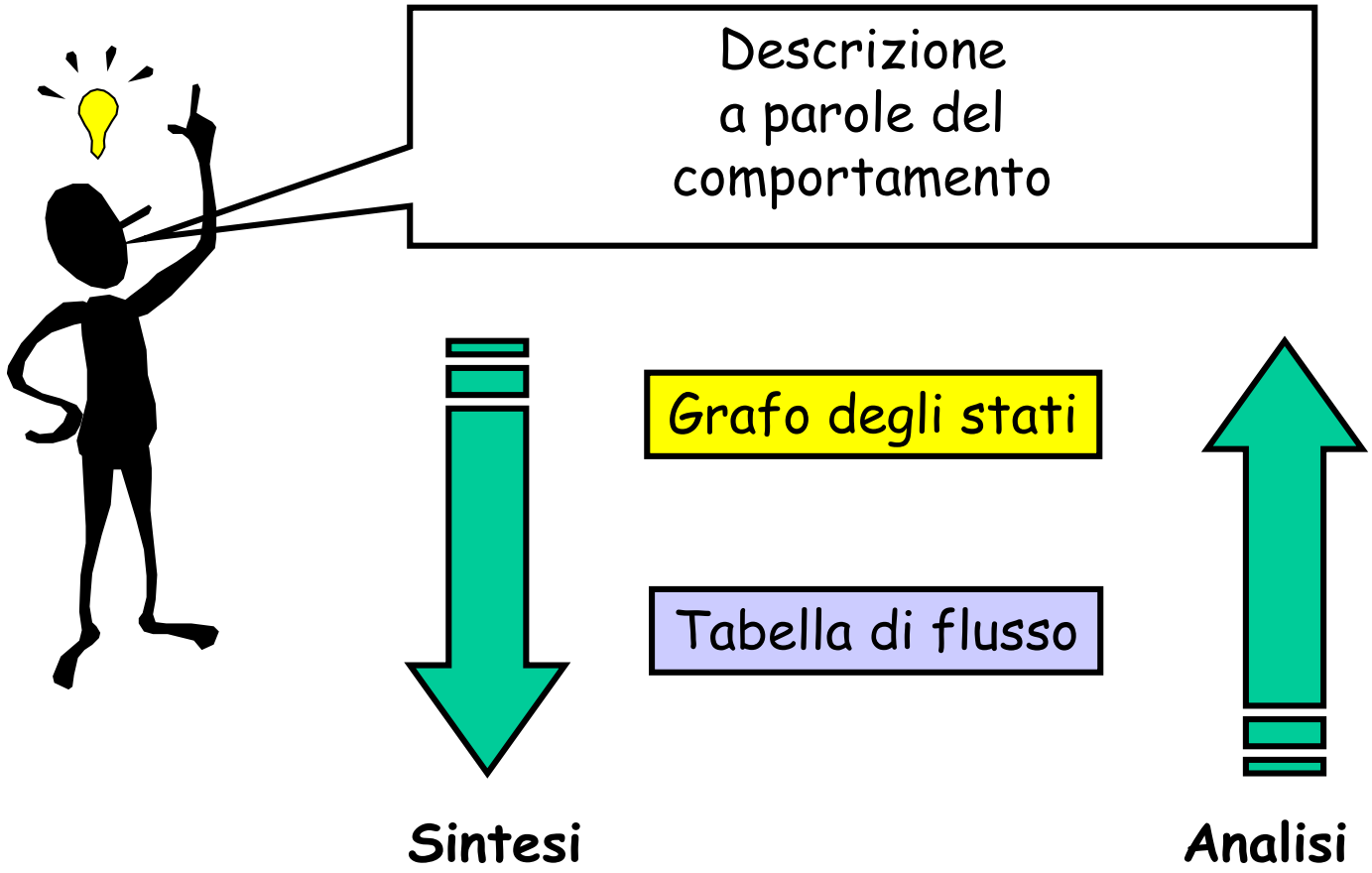




stato
assorbente

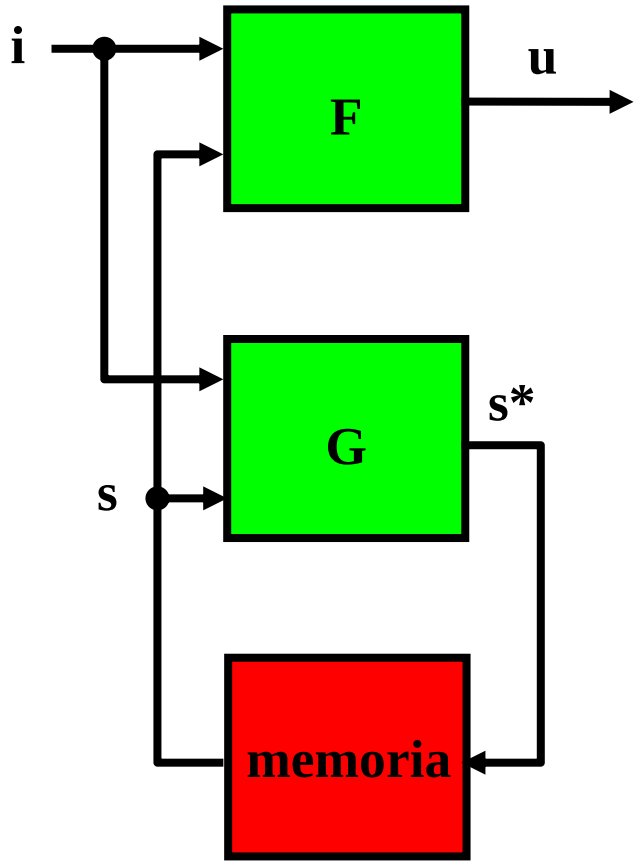
Analisi e Sintesi

- Grafo degli stati e tabella di flusso sono due strumenti diversi per descrivere una stessa macchina
- Il grafo degli stati è utile generalmente all'inizio del processo di **sintesi** (quando non è noto il numero di stati) e alla fine del processo di **analisi**
- La tabella di flusso(rappresentazione più ordinata) viene impiegata generalmente nel secondo passo della sintesi e nel penultimo passo dell'analisi

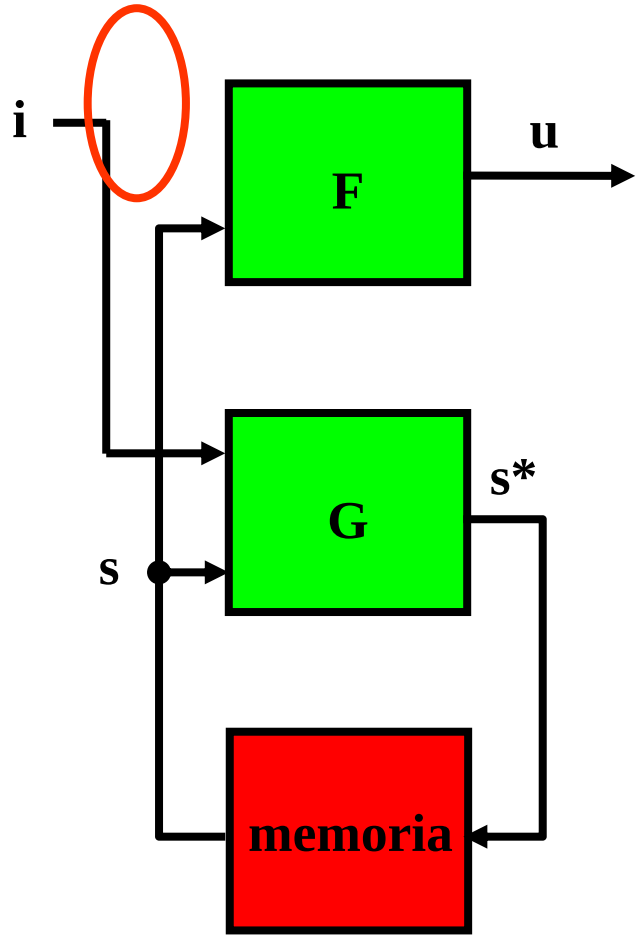


Automa di Mealy e Automa di Moore

- Automa di **Moore**: caso particolare dell'automa di Mealy in cui l'uscita dipende solo dallo **stato interno presente**



Mealy
$F : S \times I \rightarrow U$



Moore
$F : S \rightarrow U$

Dall'automa di Mealy all'automa di Moore

GRAFO:
simbolo d'uscita è all'interno del nodo

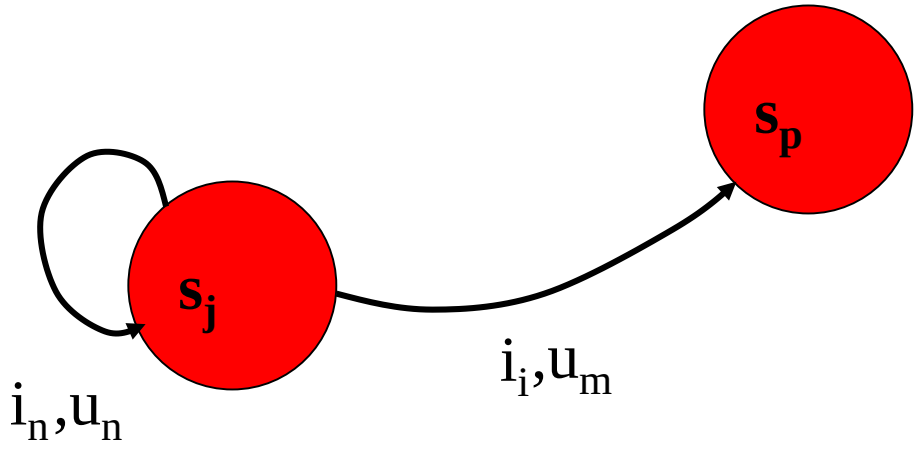
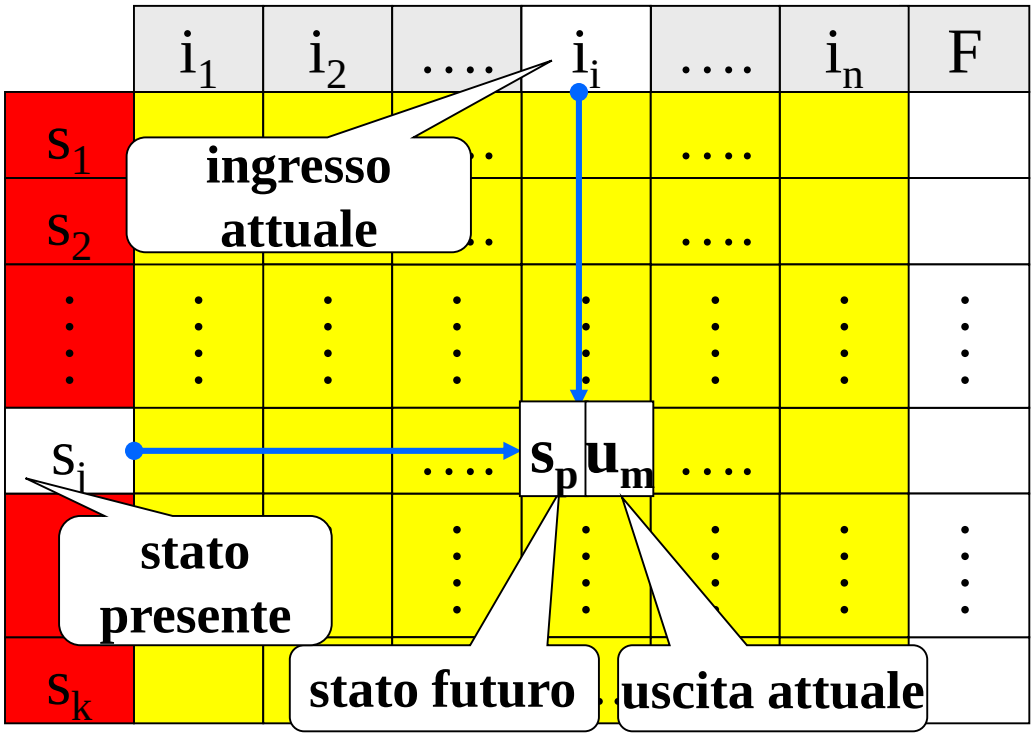


TABELLA:
ulteriore colonna per specificare la $F : S \rightarrow U$



Dall'automa di Mealy all'automa di Moore

GRAFO:
simbolo d'uscita è all'interno del nodo

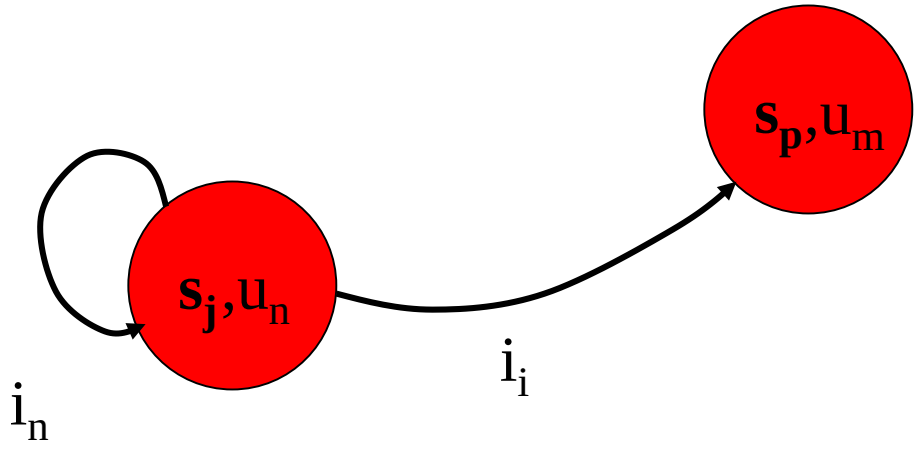
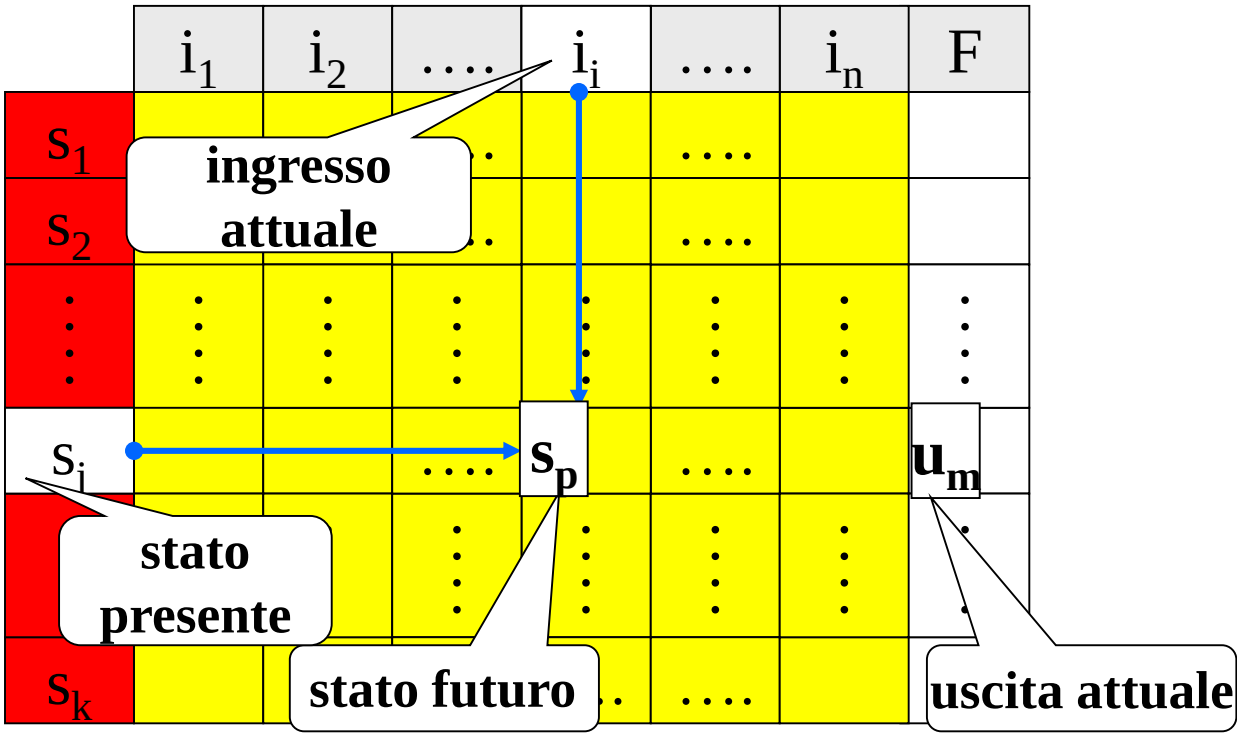


TABELLA:
ulteriore colonna per specificare la **F : S → U**



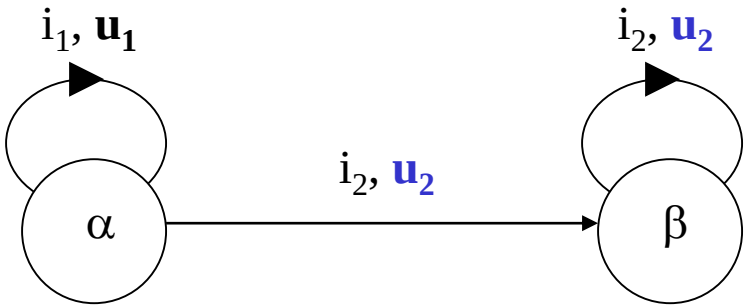
Mealy vs. Moore

- Come si vedrà in seguito, l'automa di Mealy, a fronte di una (apparentemente) maggiore complessità progettuale, ha diversi vantaggi rispetto all'automa di Moore:
  - permette di aggiornare l'uscita in modo più **rapido**
  - permette spesso una **riduzione del numero di stati** necessario alla realizzazione della rete


Reti asincrone vs.
reti sincrone

Classificazione dei comportamenti

1)



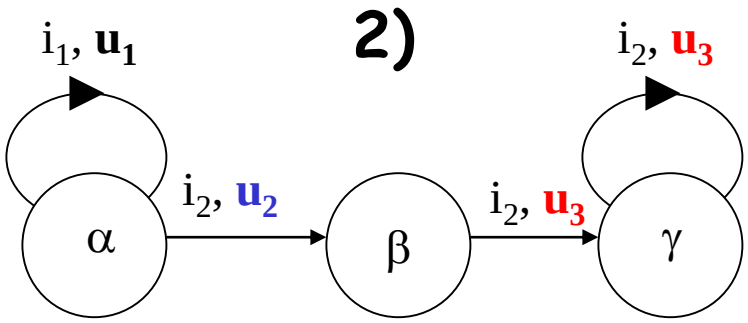
•Nel comportamento di **tipo 1** la macchina passa sempre da una condizione di stabilità all'altra, seguendo le varazioni dell'ingresso. Gli "eventi" che determinano le variazioni dell'uscita sono SOLO le variazioni dell'ingresso.

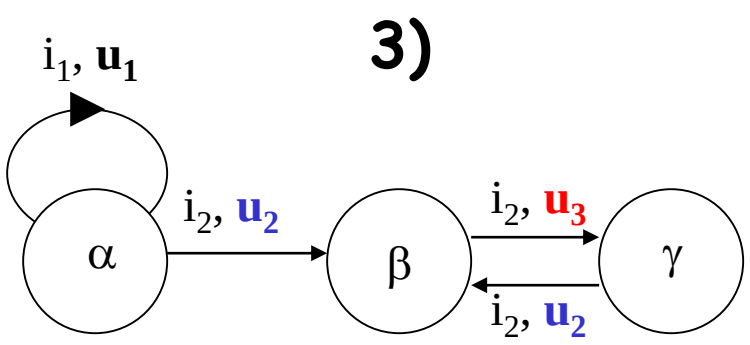
•Il comportamento di tipo 1 quindi non tiene conto **esplicitamente al trascorrere del tempo**

• Per tale ragione, le macchine sequenziali che realizzano comportamenti di tipo 1 sono dette ASINCRONE ("prive di tempo").

• Si osservi che le macchine asincrone tengono conto esplicitamente solo della *sequenza* dei simboli di ingresso ma NON della loro *durata*.

Classificazione dei comportamenti





• Nei comportamenti di **tipo 2 e 3** l'uscita cambia pur in presenza di ingresso costante. Quindi, gli eventi che determinano le variazioni dell'uscita sono anche dovuti **esplicitamente al trascorrere del tempo** (es., nel caso 2), in un istante di tempo con ingresso costante i_2 l'uscita passa da u_2 a u_3)

•Poichè si vuole poter controllare con precisione la durata dei simboli di uscita ogni transizione di stato deve svolgersi in un prefissato intervallo di tempo (*l'unità di misura del tempo*). In questo modo la durata desiderata per ciascun simbolo d'uscita può essere ottenuta mediante una opportuna sequenza di stati instabili.

•Lo stato e l'uscita possono cambiare solo allo scadere di tale intervallo di tempo

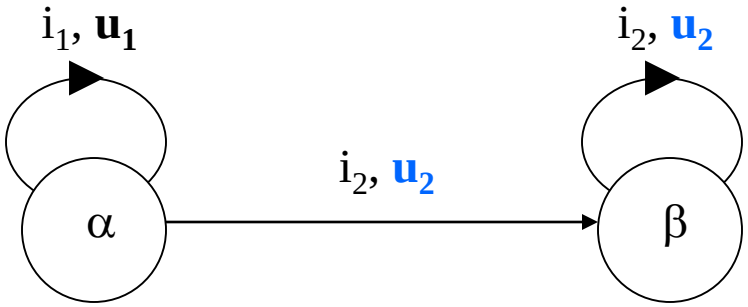
•Tali reti sequenziali sono dette SINCRONE. Possono realizzare comportamenti che tengono conto esplicitamente sia della successione dei simboli d'ingresso sia della durata di ciascun simbolo. E' possibile anche realizzare macchine prive di ingressi che tengono conto SOLO del trascorrere del tempo (es: il semaforo).

Riassumendo..

Macchina asincrona (1): ogni nuovo ingresso produce subito una stabilità e genera quindi un solo nuovo simbolo d'uscita

Evento: modifica dell'ingresso

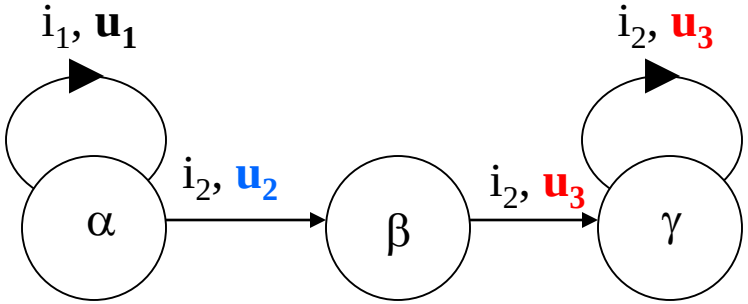
1)



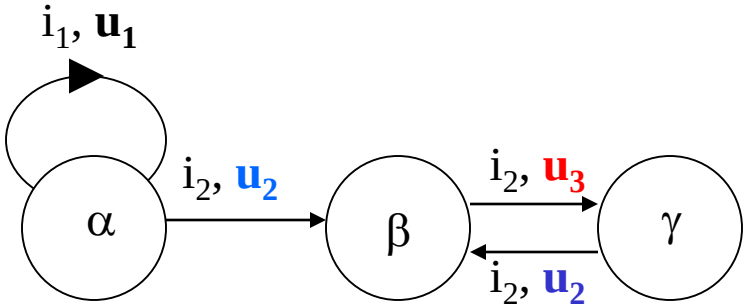
Macchina sincrona (2,3): un nuovo ingresso produce una sequenza, finita o periodica, di transizioni di stato e di simboli d'uscita

Evento: modifica dell'ingresso e trascorrere del tempo

2)



3)

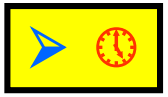


Classificazione delle macchine

Macchina combinatoria

Macchina sequenziale asincrona

Macchina sequenziale sincrona



Evento che può modificare l'uscita

- **modifica dell'ingresso**
- **trascorrere del tempo**

- D'ora in avanti ci occuperemo prima delle macchine asincrone e successivamente delle macchine sincrone