Lezione #19

Introduzione ai Sistemi a Eventi Discreti

Sommario

- Introduzione ai sistemi a eventi discreti
- Modellistica dei SED con automi a stati finiti
- Esempi di controllo con automi a stati finiti

Introduzione

- Nelle lezioni precedenti, abbiamo visto che è possibile descrivere il comportamento di un **sistema dinamico** attraverso **equazioni (statiche o differenziali)** che legano variabili di **ingresso, stato** e **uscita**

- Abbiamo anche visto che, per questo tipo di sistemi, è possibile adottare **strategie di controllo modulante**

Introduzione

- Abbiamo inoltre studiato come sia possibile realizzare strategie di controllo di tipo **logico-sequenziale** per alcuni sistemi di automazione industriale

- I diversi tipi di controllo hanno tra loro una **relazione gerarchica** (e.g. modello CIM)

Introduzione

- **Controllo modulante:**
  - Presente al livello di campo
  - Agisce su variabili continue, a cui impone un andamento desiderato
  - (Spesso) basato sulla teoria del controllo classica
  - Gestisce situazioni non strettamente correlate tra loro (e.g. movimentazione di parti, gestione di livelli/temperature, ecc.)
  - Può includere algoritmi complessi (e.g. movimentazione degli assi di un robot)
  - Ha specifiche assegnate in termini di tempi di risposta, errore a regime, sovraelongazione…

Introduzione

- **Controllo logico sequenziale**
  - Livelli di controllo o di cella
  - Si occupa di integrare più elementi tra loro
    - Avviamento/spegnimento
    - Gestione di sequenze di lavorazione
    - Gestione di guasti/emergenze
  - Può sfruttare gli algoritmi implementati ai livelli inferiori
  - Le specifiche sono date in termini di comportamento desiderato del sistema, ad esempio tramite SFC funzionali

Sistemi a eventi
Motivazione

Per la progettazione di un buon sistema di controllo abbiamo bisogno di:

1. poter **definire le specifiche** in maniera non ambigua
2. verificare che tali specifiche siano **soddisfatte**

Sistemi a eventi
Motivazione

Nel caso del **controllo modulante**:

1. è possibile definire le specifiche in vari modi, ad esempio come **vincoli sulla risposta indiciale del sistema a ciclo chiuso**
2. Il **soddisfacimento** di tali specifiche può essere **verificato matematicamente** (e.g. analizzando la fdt di anello)

Sistemi a eventi
Motivazione

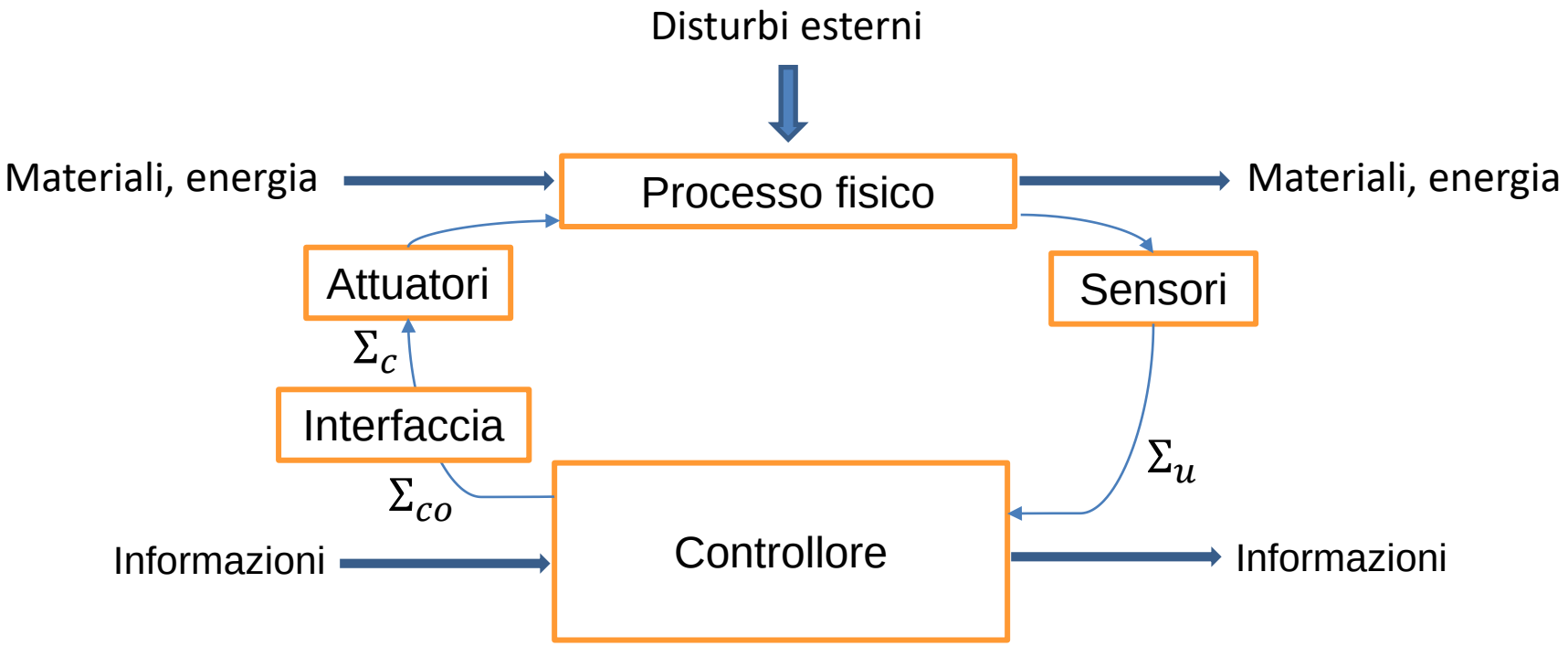
Nel caso del **controllo logico**:

1. è possibile definire le specifiche in maniera non ambigua ricorrendo, ad esempio, ad SFC funzionali
2. **serve uno strumento per verificare che tali specifiche siano soddisfatte**

Controllo a forzamento di eventi

- Idea:
  - Gli eventi che rappresentano l'inizio di un'attività vengono **generati da un dispositivo esterno in modo che vengano eseguite solo le sequenze desiderate**
  - Questi eventi sono detti **controllabili**
  - In generale, gli eventi controllabili saranno un **sottoinsieme** degli eventi generati dal controllore (al cui interno saranno generati anche eventi non controllabili, e.g. il termine di un intervallo temporale)

Controllo a forzamento di eventi



Σ_c: insieme degli eventi controllabili

Σ_u: insieme degli eventi non controllabili

$\Sigma_c \subseteq \Sigma_{co}$: insieme degli eventi generati dal controllore

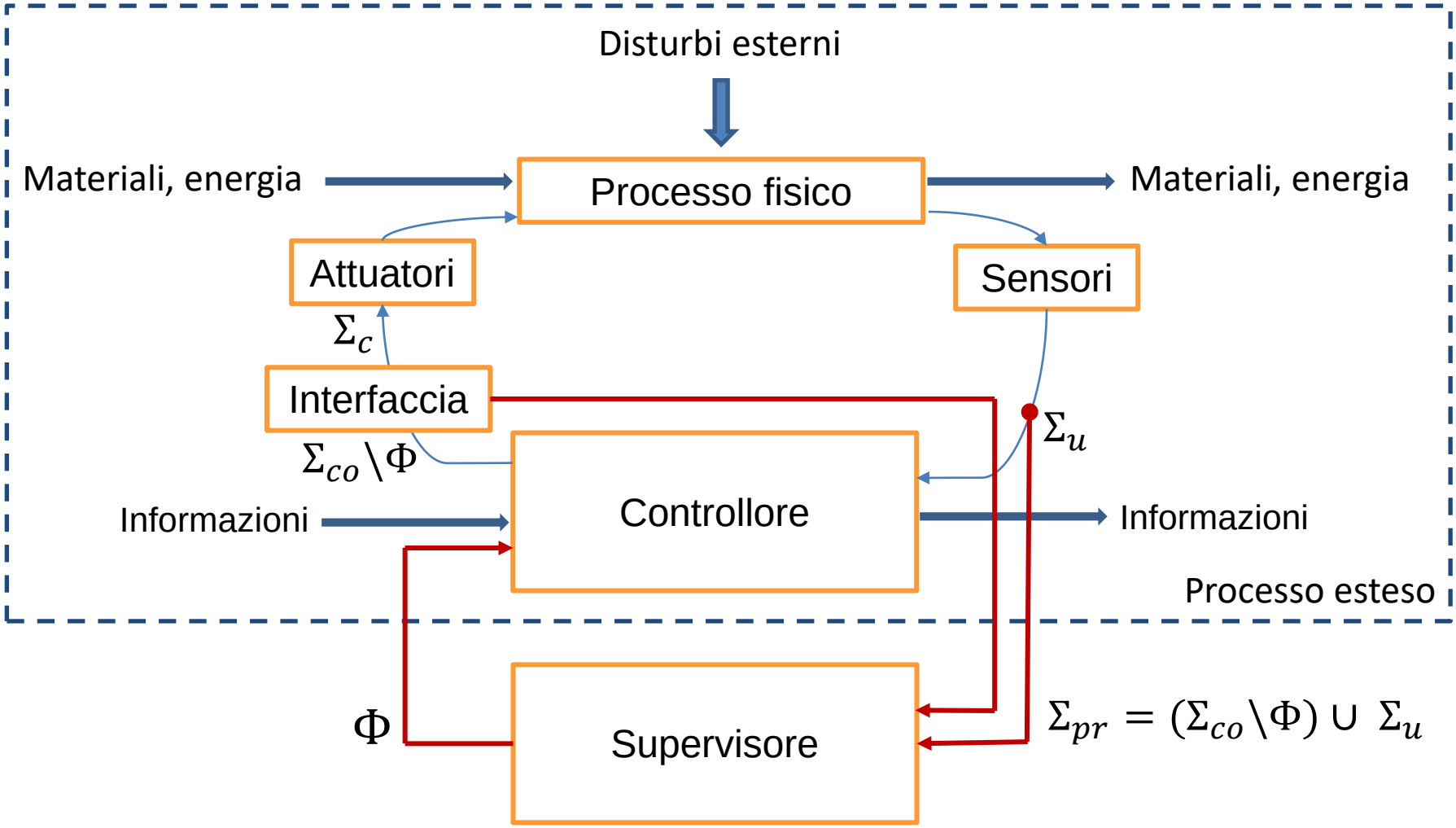
Controllo a forzamento di eventi

- Questo approccio è intuitivo, ma può condurre a **sistemi molto complessi**
  - Applicabile all'interno di un sottosistema
  - Può essere inadatto a coordinare più sottosistemi
- Un approccio in genere più efficiente è quello del **controllo di supervisione**

Controllo di supervisione

- Idea:
  - **Estendere il processo** fino a includere anche i dispositivi di controllo diretto
  - Al **controllore** viene lasciato il compito di **agire direttamente sul processo**
  - Un **supervisore** **può abilitare/disabilitare alcune delle azioni** che possono essere eseguite dal controllore

Controllo di supervisione



Φ: insieme degli eventi da disabilitare

Controllo di supervisione

- **Obiettivo**: disabilitare alcuni degli eventi controllabili in modo che il processo esteso rispetti le specifiche

- Dobbiamo:
  - **Modellare** il processo esteso
  - Trovare delle **strategie di sintesi** per il supervisore

Sistemi a eventi
Motivazione

- Dal punto di vista del **controllo logico**, l'evoluzione di un sistema di automazione può essere considerata come una **sequenza di eventi**

- Ciascun evento può corrispondere, ad esempio, all'inizio o al termine di una certa attività o fase di lavorazione

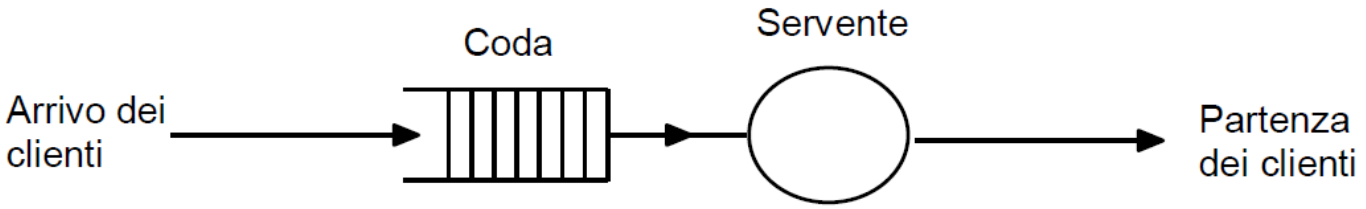
Sistemi a eventi
Motivazione

- Anche questo genere di sistemi appartiene alla classe dei **sistemi dinamici**, ma spesso ha uno stato non rappresentabile tramite un vettore di numeri reali

- I sistemi a eventi possono essere descritti tramite dei **formalismi** appositi

Sistemi a eventi
Esempio

- Esempio: consideriamo un **sistema server-client**
  - Le richieste vengono memorizzate in un buffer a capacità infinita
  - La coda è gestita in maniera FIFO



Sistemi a eventi
Esempio

- Esempio: consideriamo un **sistema server-client**
  - All'**arrivo di una richiesta**, il numero di elementi in coda viene incrementato di 1
  - Quando il server ha **gestito una richiesta**, il numero di elementi in coda viene decrementato di 1
  - Il server passa **da libero a occupato** quando inizia a elaborare una richiesta e **da occupato a libero** quando termina

Sistemi a eventi
Esempio

Si tratta di un **sistema dinamico**?

- Sì: il comportamento dipende dalla "storia passata" (e.g. numero delle richieste arrivate)
- La conoscenza della condizione iniziale e degli ingressi consente di determinare l'evoluzione

Sistemi a eventi
Esempio

Ma…

- L'evoluzione non è guidata dal tempo, ma dagli **eventi** asincroni che occorrono

- Le variabili di stato assumono valori in un insieme **non necessariamente numerico** (es. occupato/libero)

Sistemi a Eventi Discreti

Definiamo **sistema (dinamico) a eventi discreti (DEDS)** un sistema nel quale le variabili di stato sono **lessicali** e l'evoluzione dello stato è determinata dall'occorrenza di **eventi**

Sistemi a Eventi Discreti
Linguaggi

Variabili **lessicali:**

- Sono variabili **non rappresentabili numericamente**
- I possibili eventi e/o stati (se in numero finito) possono essere **enumerati**
- In particolare, posso considerare i possibili eventi come lettere di un **alfabeto**
- Una sequenza di eventi sarà una **stringa**
- Tutte le possibili sequenze formeranno un **linguaggio**

Sistemi a Eventi Discreti
Linguaggi

È possibile distinguere linguaggi (e modelli) **logici** e **temporali**

- **Linguaggio/modello logico**
  L'interesse è posto sull'ordine di occorrenza degli eventi
  - es. $e_1 e_2 e_3$ ...

- **Linguaggio/modello temporale**
  L'interesse è sia sull'ordine di occorrenza degli eventi che sull'istante in cui avvengono
  - es. $(e_1, t_1)(e_2, t_2)(e_3, t_3)$ ...

Sistemi a Eventi Discreti

- I sistemi a eventi discreti sono lo strumento naturale per descrivere i problemi di controllo logico

- Possono essere **complessi da modellare**

- È necessario rappresentare situazioni quali
  - **Concorrenza** di eventi (attività simultanee)
  - **Conflitti** (gestione di risorse comuni)
  - **Sincronizzazione** tra eventi (coordinamento)

Sistemi a Eventi Discreti

- Come già detto, cerchiamo un **formalismo** per descrivere e analizzare l'evoluzione di questi sistemi

- Potremo sfruttare lo stesso formalismo per la **sintesi di un supervisore**

Modellistica di SED

- Vedremo due strumenti per la modellistica di SED:
  - Automi a stati finiti
  - Reti di Petri

- Considereremo solo il caso di modelli logici (senza temporizzazione)

Automi a stati finiti

- Un **automa a stati finiti** è un sistema che può
  - Trovarsi in un dato **stato,** nel quale produce delle **uscite**
  - **Transitare** da uno stato a un altro all'accadere di determinati **eventi**

Automi a stati finiti

- **Esempio**: supponiamo di voler modellare la trasmissione sequenziale di pacchetti da un nodo A a un nodo B di una rete

- Il canale di comunicazione può trasmettere un pacchetto alla volta, che potrebbe non giungere a destinazione nel caso di problemi di trasmissione

Automi a stati finiti

- Supponiamo che la trasmissione avvenga secondo un semplice **protocollo *send-wait***
  - Quando invia un pacchetto, A avvia un timer
  - Quando riceve un pacchetto, B invia un *ack*

Automi a stati finiti

- Supponiamo che la trasmissione avvenga secondo un semplice **protocollo *send-wait***

  – Se A riceve l'*ack,* la trasmissione è avvenuta con successo e si può passare al pacchetto successivo

  – Se A non riceve l'*ack,* dopo un time-out rispedisce il dato

Automi a stati finiti

- Supponiamo che la trasmissione avvenga secondo un semplice **protocollo *send-wait***
  - Poiché anche il segnale di *ack* potrebbe andare perduto, A aggiunge al pacchetto un bit di intestazione
  - A ogni nuova trasmissione, il valore di questo bit cambia

Automi a stati finiti

- Supponiamo che la trasmissione avvenga secondo un semplice **protocollo *send-wait***
  - B si aspetta di ricevere pacchetti 0 e 1 in maniera alternata
  - Se riceve lo stesso bit di intestazione due volte, il dato potrebbe essere la ripetizione del precedente

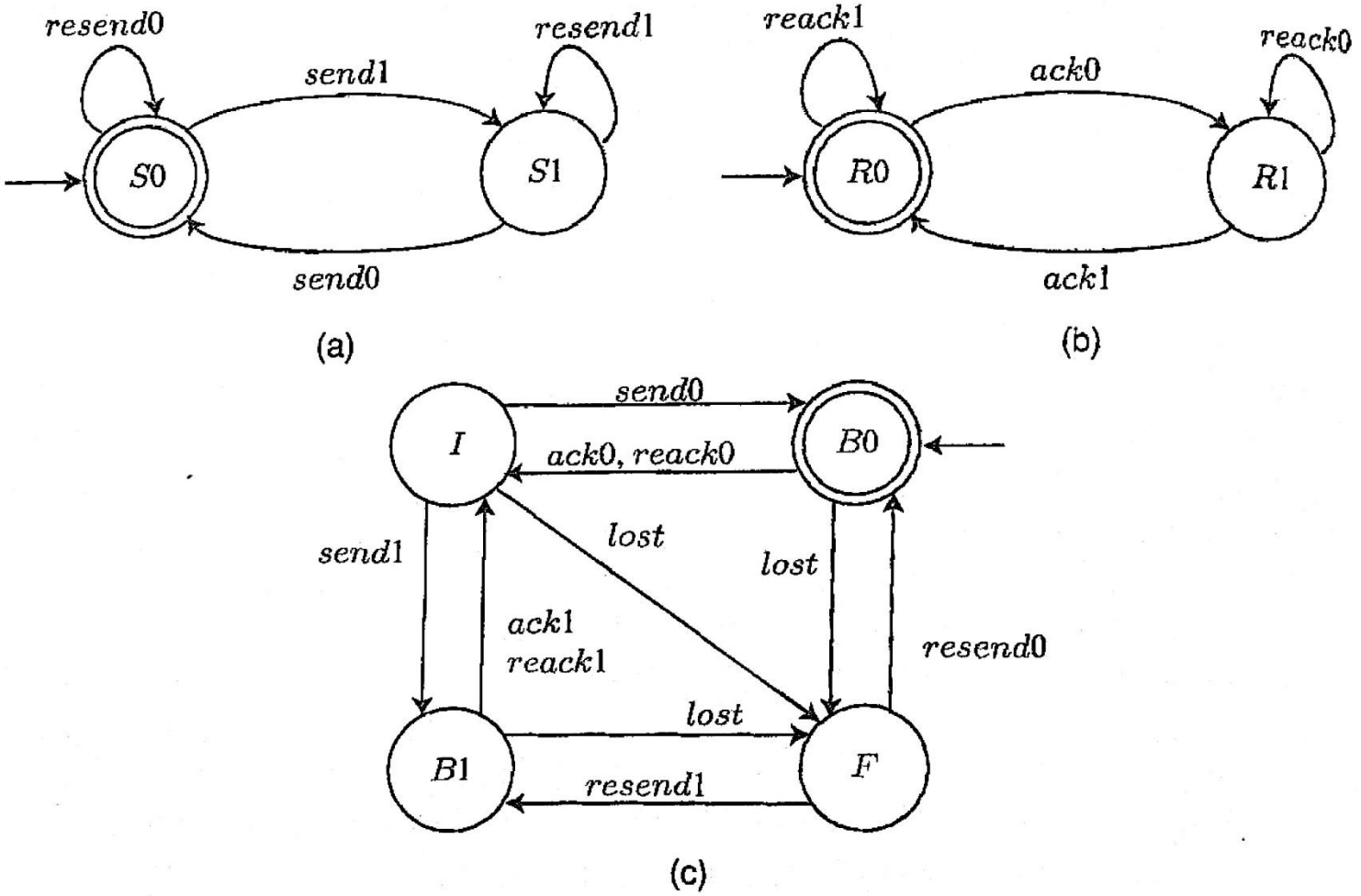
Automi a stati finiti

- Nodo A
  - Stato S0: stato in cui si trova dopo aver spedito il pacchetto-0 (dopo gli eventi *send0* o *resend0*)
  - Stato S1: stato in cui si trova dopo aver spedito il pacchetto-1 (dopo gli eventi *send1* o *resend1*)
- Nodo B
  - Stato R0: stato in cui si trova dopo aver ricevuto il pacchetto-0 (prima degli eventi *ack0* o *reack0*)
  - Stato R1: stato in cui si trova dopo aver ricevuto il pacchetto-1 (prima degli eventi *ack1* o *reack1*)

Automi a stati finiti

- Canale di comunicazione
  - Stato I: canale a riposo
  - Stato B0: sul canale c'è un pacchetto-0
  - Stato B1: sul canale c'è un pacchetto-1
  - Stato F: il pacchetto non può essere consegnato (evento *lost*)

Automi a stati finiti



Automi a stati finiti
Definizione

- Un **automa a stati finiti** è definito da
  - X insieme finito di stati
  - E insieme (o **alfabeto**) finito di eventi
  - $f: X \times E \rightarrow X$ funzione transizione di stato $f(x, e) = y$ se esiste una transizione etichettata con l'evento e che porta da x a y
  - $x_0 \in X$ stato iniziale
  - $X_m \subseteq X$ insieme degli stati finali (o marcati), stati con un particolare significato per chi realizza il modello

Automi a stati finiti
Definizione

- L'automa è definito dalla quintupla

$$A = (X, E, f, x_0, X_m)$$

- È a stati finiti se l'insieme X è finito

- La dinamica è descritta da

$$x' = f(x, e)$$

Automi a stati finiti
Definizione

- Posso inoltre associare a ciascuno stato delle uscite
  - **Macchina di Mealy**: uscite definite in corrispondenza delle transizioni di stato

$$y = h(x, e)$$

  - **Macchina di Moore**: uscite funzione solo dello stato

$$y = h(x)$$

Automi a stati finiti
Rappresentazione grafica

- Un'**automa a stati finiti** può essere rappresentato un **grafo orientato** in cui
  - i **nodi** rappresentano i possibili **stati** del sistema
  - Gli **archi** rappresentano le transizioni collegate ai possibili **eventi**
  - Lo **stato iniziale** è indicato da una freccia entrante
  - Gli **stati finali** sono doppiamente cerchiati (o in nero)

Automi a stati finiti
Composizione

- Il modello complessivo sarà la **composizione** degli automi associati alle singole parti del sistema

- Tale composizione deve rappresentare il fatto che i sistemi lavorano in **concorrenza**
  - **Operazioni private** (che cioè comportano la partecipazione di un solo sistema) possono essere eseguite in maniera asincrona
  - **Operazioni comuni** (che cioè comportano la partecipazione di più sistemi) vanno sincronizzate

Composizione concorrente (o parallela):

- Dati gli automi $A_1 = (X_1, E_1, f_1, x_{0,1}, X_{m,1})$ e $A_2 = (X_2, E_2, f_2, x_{0,2}, X_{m,2})$, la loro composizione $A_1 || A_2$ è data da

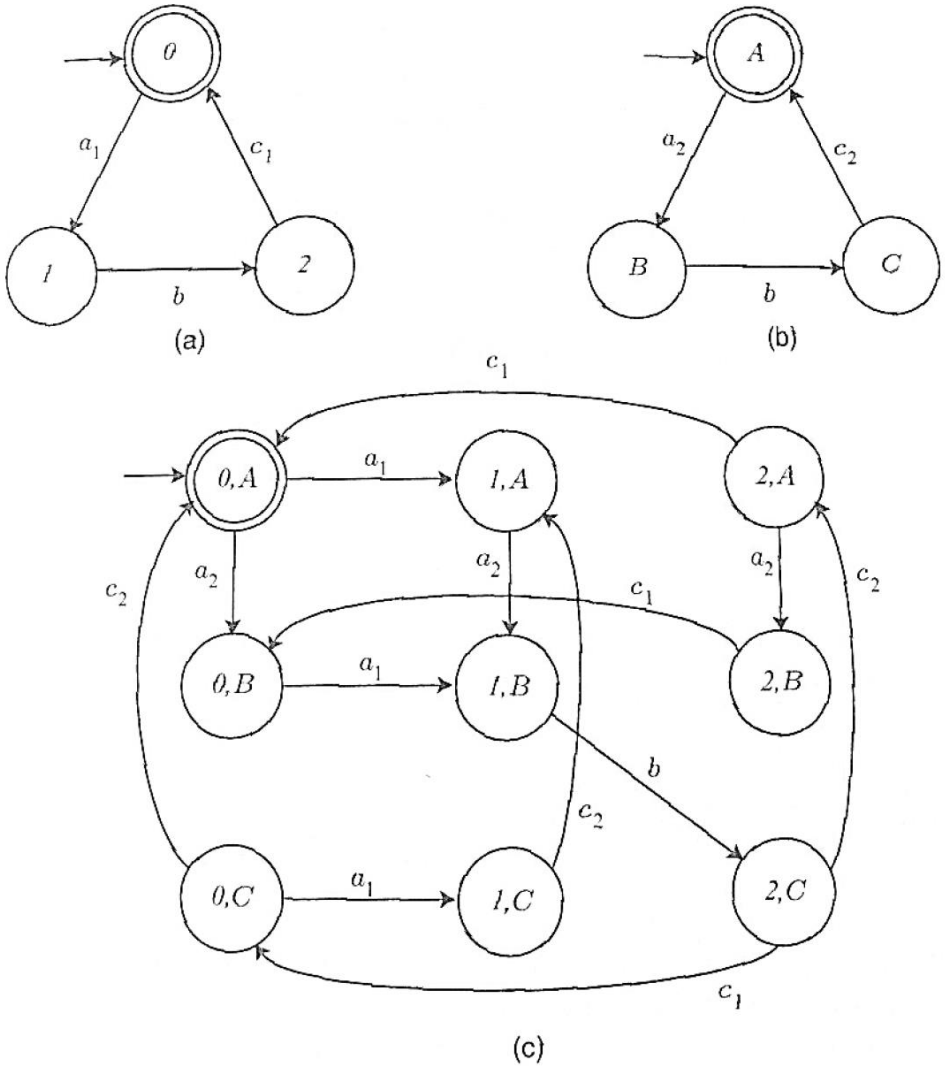
  - $X \subseteq X_1 \times X_2$ (considero solo il sottoinsieme di stati *accessibile* [3])
  - $E = E_1 \cup E_2$
  - $x_0 = (x_{0,1}, x_{0,2})$
  - $X_m \subseteq X_{m,1} \times X_{m,2}$

Composizione concorrente (o parallela):

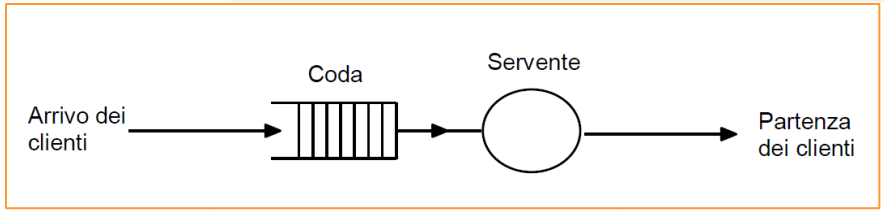
- Dati gli automi $A_1 = (X_1, E_1, f_1, x_{0,1}, X_{m,1})$ e $A_2 = (X_2, E_2, f_2, x_{0,2}, X_{m,2})$, la loro composizione $A_1 || A_2$ è data da

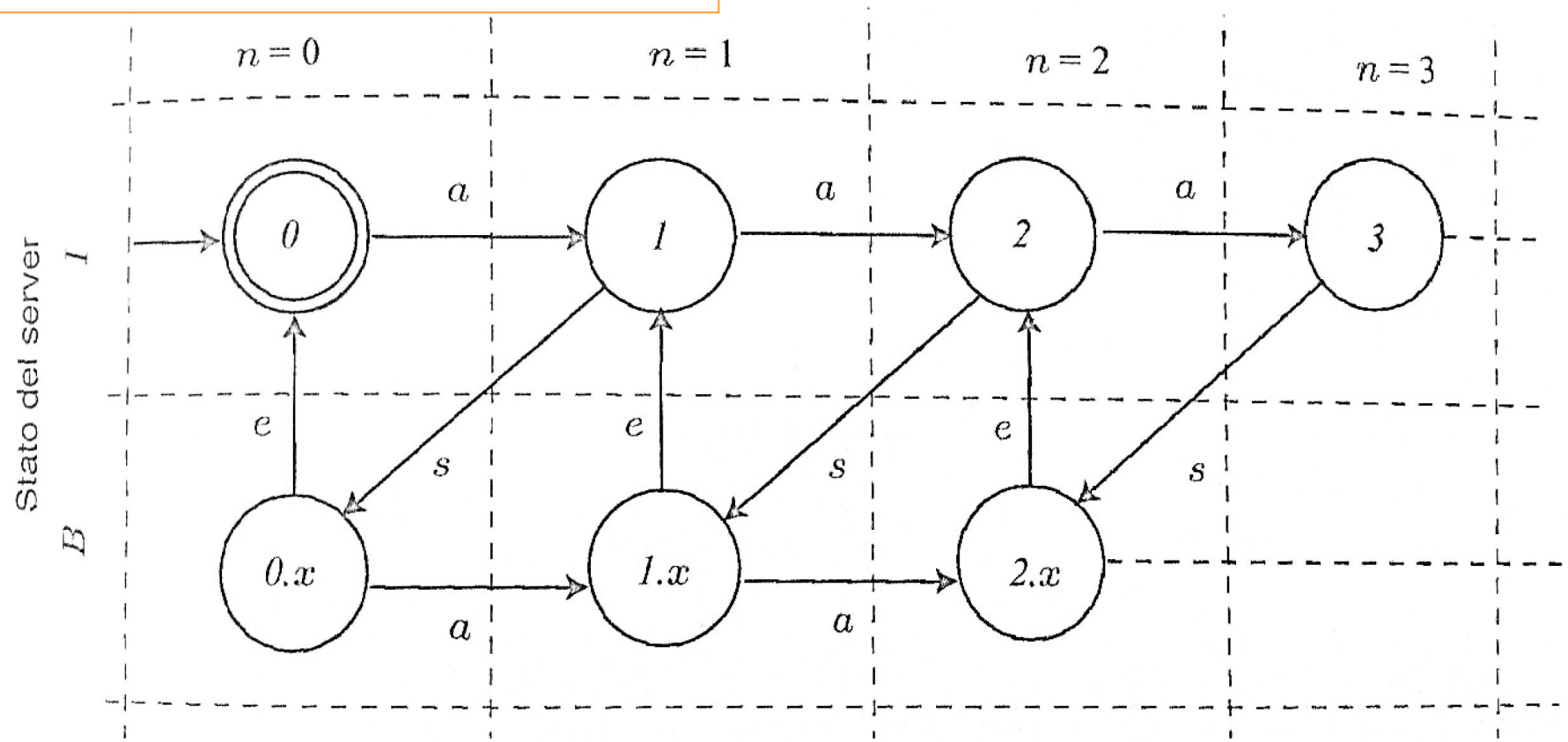
  - $f\big((x_1, x_2), e\big) = \big(f_1(x_1, e), f_2(x_2, e)\big)$ se $e \in E_1 \cap E_2$
  - $f\big((x_1, x_2), e\big) = \big(f_1(x_1, e), x_2\big)$ se $e \in E_1, e \notin E_2$
  - $f\big((x_1, x_2), e\big) = \big(x_1, f_2(x_2, e)\big)$ se $e \notin E_1, e \in E_2$
  - Non definita altrimenti

Automi a stati finiti
Composizione



Automi a stati finiti
Composizione





Automi a stati finiti
Composizione

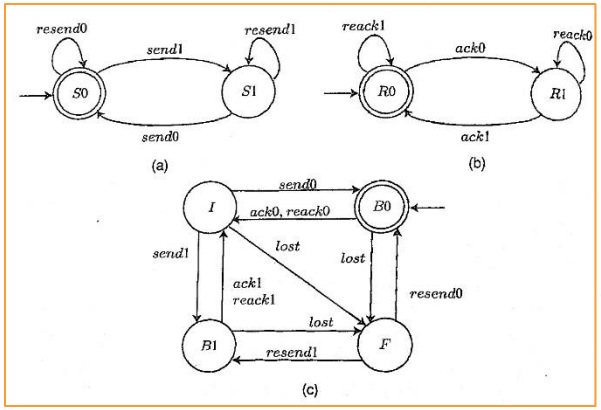
- In questo caso abbiamo uno stato a cardinalità finita (server) e uno a cardinalità infinita numerabile (client)

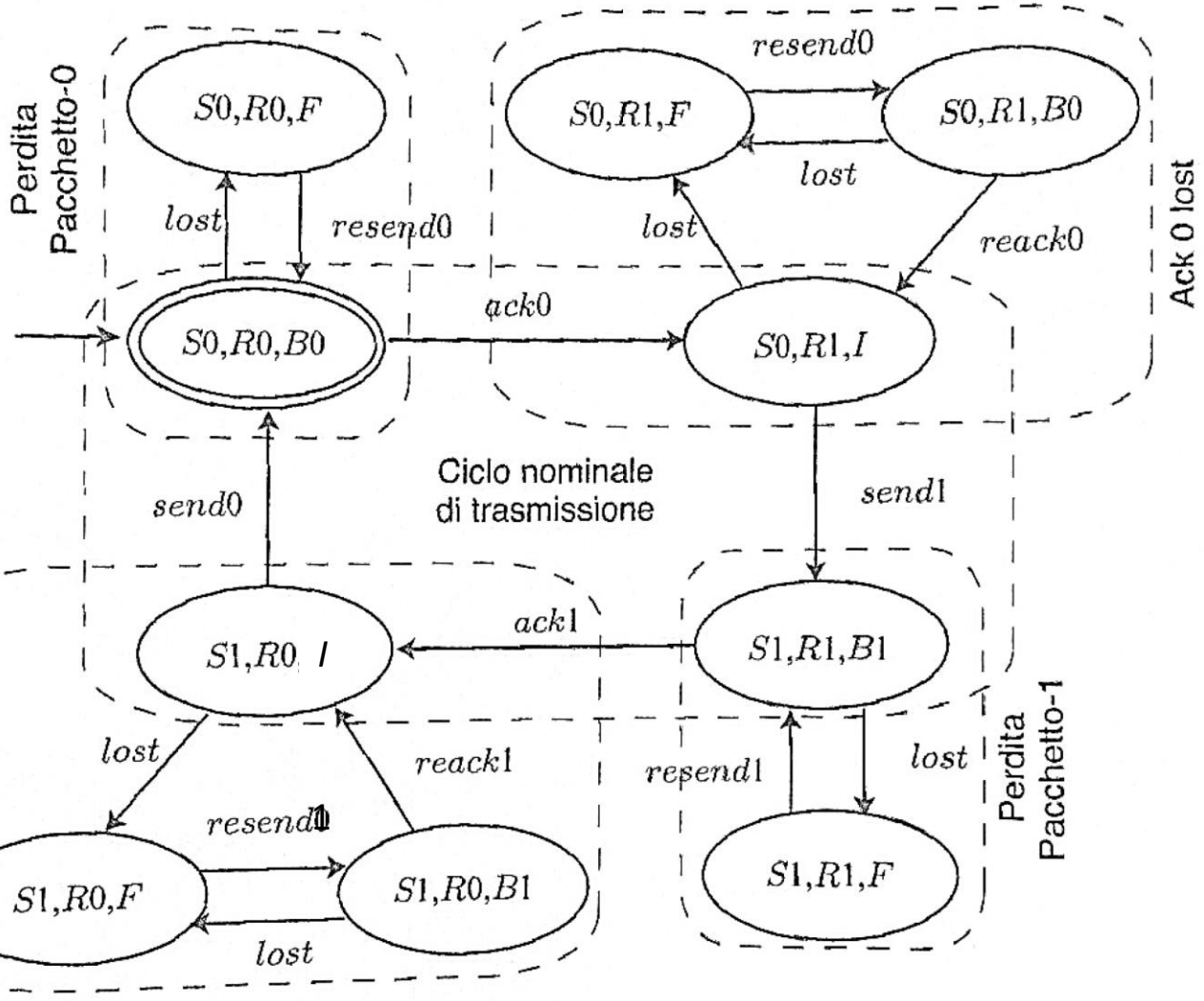
- $\Sigma_{pr} = \{a, s, e\}$
  - a = *arrival*, aumenta di uno la coda
  - s = *start of service*, mette il buffer su occupato e riduce di uno la coda
  - e = *end of service*, libera il buffer

- Stato server: {I (*Idle),* B (*Busy)*}

- Stato complessivo: $X = \{I, B\} \times \mathbb{N}$

Automi a stati finiti
Composizione





Controllo con automi

- È possibile utilizzare gli automi a stati finiti per progettare e implementare **supervisori**

- Due approcci possibili:
  - **forzamento di eventi**
  - **controllo di supervisione**

Controllo con automi
Esempio

- Supponiamo di avere un sistema composto da due macchine operatrici **M1** e **M2** e un buffer **B**:
  - La prima lavora dei pezzi e poi li pone in un buffer di capacità 1
  - La seconda preleva i semilavorati dal buffer ed effettua una seconda lavorazione

Controllo con automi
Esempio

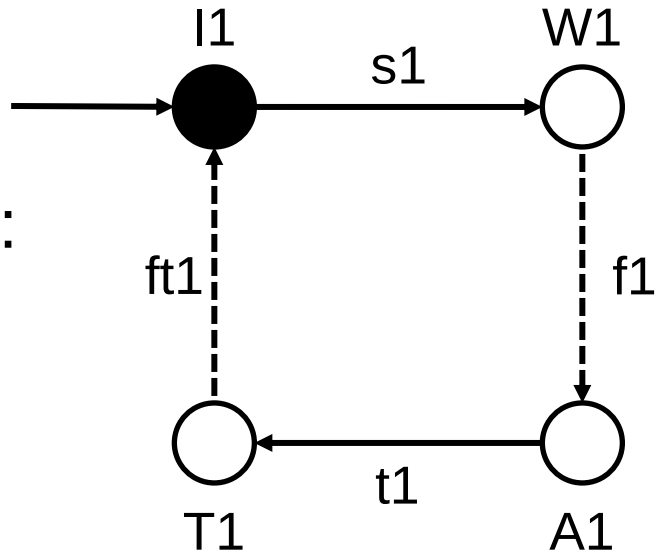
- La macchina **M1** ha 4 stati:
  - Libera (I1)
  - Pezzo in lavorazione (W1)
  - Termine lavorazione del pezzo e attesa buffer (A1)
  - Posizionamento del pezzo semilavorato nel buffer (T1)

Controllo con automi
Esempio

- La macchina **M1** può cambiare stato a fronte dei seguenti eventi:
  - Inizio lavorazione (s1)
  - Fine lavorazione (f1), *non controllabile*
  - Inizio trasferimento (t1)
  - Fine trasferimento (ft1), *non controllabile*

Controllo con automi
Esempio

- Automa per la macchina **M1**
  - $X_{M1} = \{I1, W1, A1, T1\}$
  - $\Sigma_{M1} = \{s1, f1, t1, ft1\}$
  - Funzione transizione di stato:
    - $\delta_{M1}(I1, s1) = W1$
    - $\delta_{M1}(W1, f1) = A1$
    - $\delta_{M1}(A1, t1) = T1$
    - $\delta_{M1}(T1, ft1) = I1$
  - $x_{0,M1} = \{I1\}$
  - $X_{m,M1} = \{I1\}$ (associo al ritorno in I1 il significato di terminazione di un ciclo)



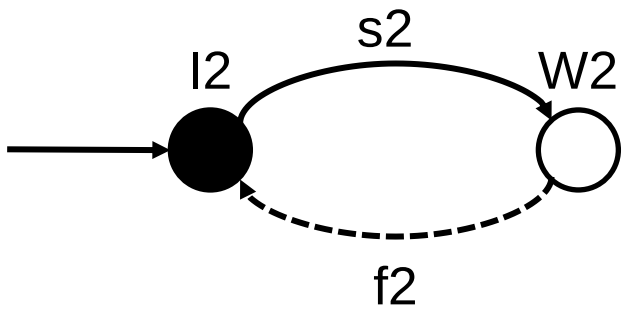
Controllo con automi
Esempio

- La macchina **M2** ha 2 stati:
  - Libera (I2)
  - Prelievo pezzo e lavorazione (W2)

- Le transizioni avvengono a fronte degli eventi:
  - Inizio lavorazione (s2)
  - Termine lavorazione (f2), *non controllabile*

Controllo con automi
Esempio

- Automa per la macchina **M2**
  - $X_{M2} = \{I2, W2\}$
  - $\Sigma_{M2} = \{s2, f2\}$
  - Funzione transizione di stato:
    - $\delta_{M2}(I2, s2) = W2$
    - $\delta_{M2}(W2, f2) = I2$
  - $x_{0,M2} = \{I2\}$
  - $X_{m,M2} = \{I2\}$



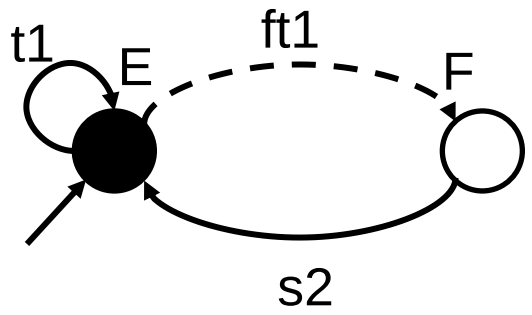
Controllo con automi
Esempio

- Il buffer B, invece, **non è controllato**

- Vogliamo però che
  - la macchina M1 attenda che il buffer sia libero prima di depositare un nuovo pezzo
  - la macchina M2 attenda che il buffer sia pieno prima di provare a prelevare un pezzo

Controllo con automi
Esempio

- Possiamo **descrivere il comportamento desiderato** del buffer con un ulteriore automa
- Stati:
  - buffer vuoto (E)
  - buffer pieno (F)
- Eventi:
  - inizio trasferimento da M1 (t1)
  - fine trasferimento da M1 (ft1)
  - prelievo pezzo da M2 (s2)



Controllo con automi
Esempio – controllo a forzamento di eventi

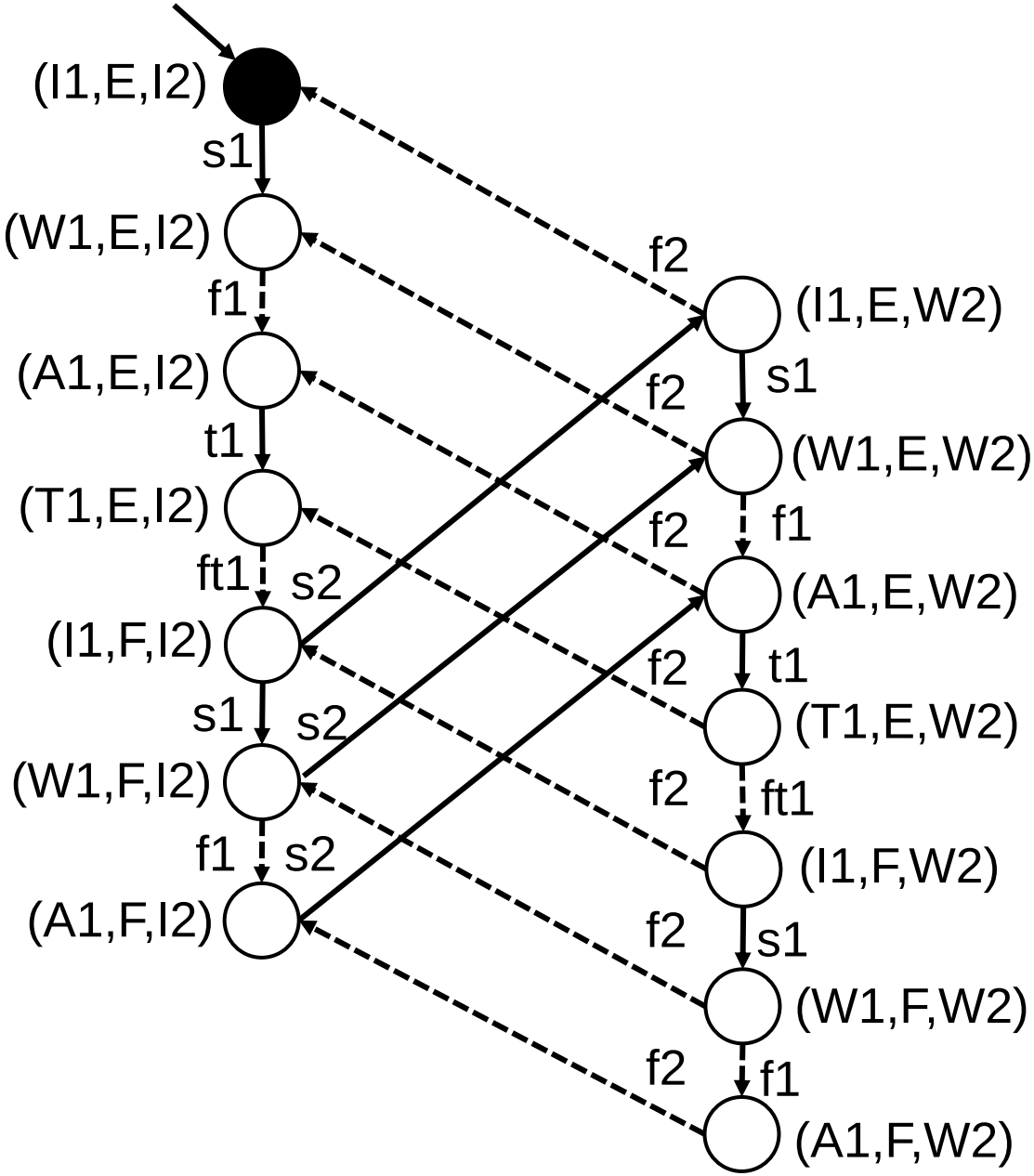
- Posso considerare la composizione concorrente dei tre automi visti

- M1||B||M2 **includerà il comportamento desiderato per il buffer**

  → **è un modello del sistema controllato!**

- Stati:
  - $X = X_{M1} \times X_{M2} \times X_B$

- Eventi:
  - $\Sigma_c = \{s1, t1, s2\}$
  - $\Sigma_u = \{f1, ft1, f2\}$


(I1,E,I2)
s1
(W1,E,I2)
f1
(A1,E,I2)
t1
(T1,E,I2)
ft1
s2
(I1,F,I2)
s1
s2
(W1,F,I2)
f1
s2
(A1,F,I2)
f2
(I1,E,W2)
s1
f2
(W1,E,W2)
f1
f2
(A1,E,W2)
t1
f2
(T1,E,W2)
ft1
f2
(I1,F,W2)
s1
f2
(W1,F,W2)
f1
f2
(A1,F,W2)

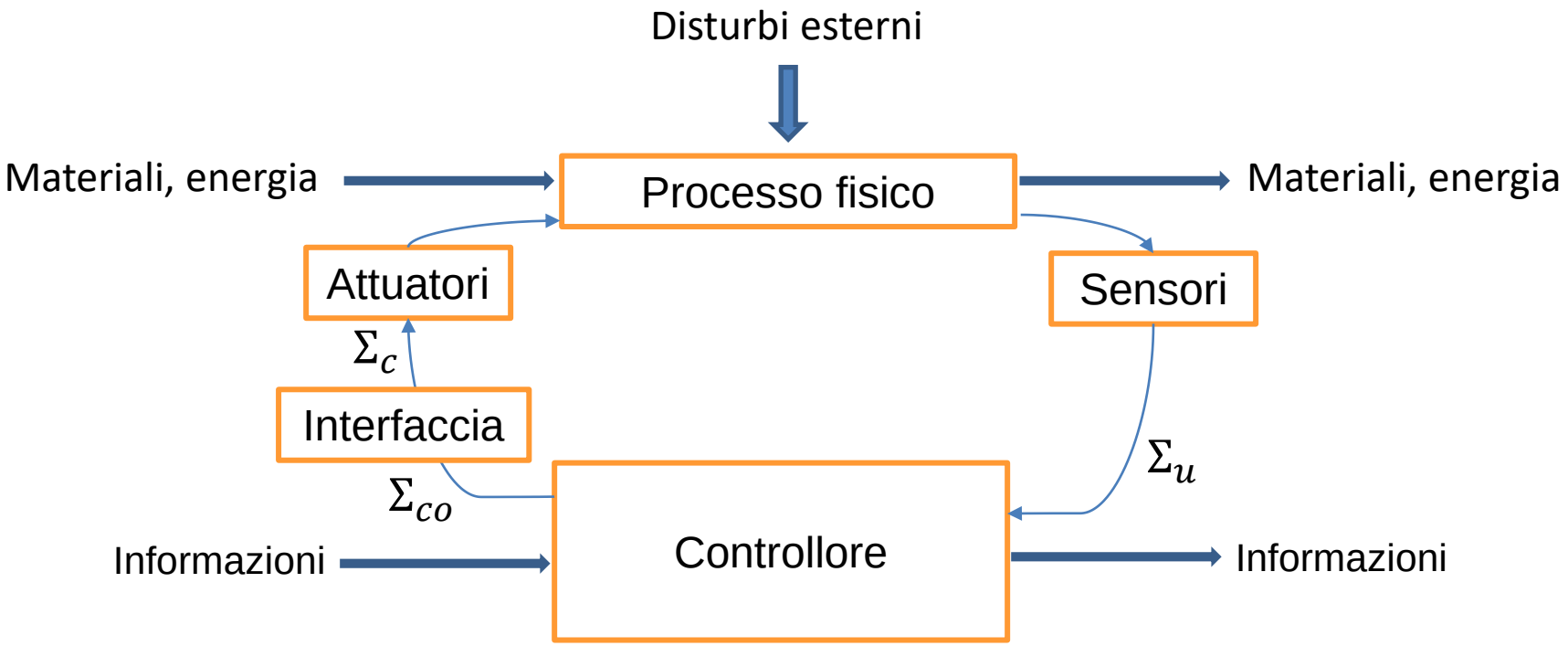
Controllo con automi
Esempio – controllo a forzamento di eventi

- Approccio a **forzamento di eventi**: gli eventi controllabili (= inizio di un'attività) vengono generati da un **dispositivo esterno**

- Si può associare a ogni stato un'uscita consistente in un vettore di variabili booleane, per indicare **quali degli eventi Σ_c vengono forzati, ovvero quali azioni vengono effettuate sul processo**
  → si otterrà una macchina di Moore

Controllo con automi

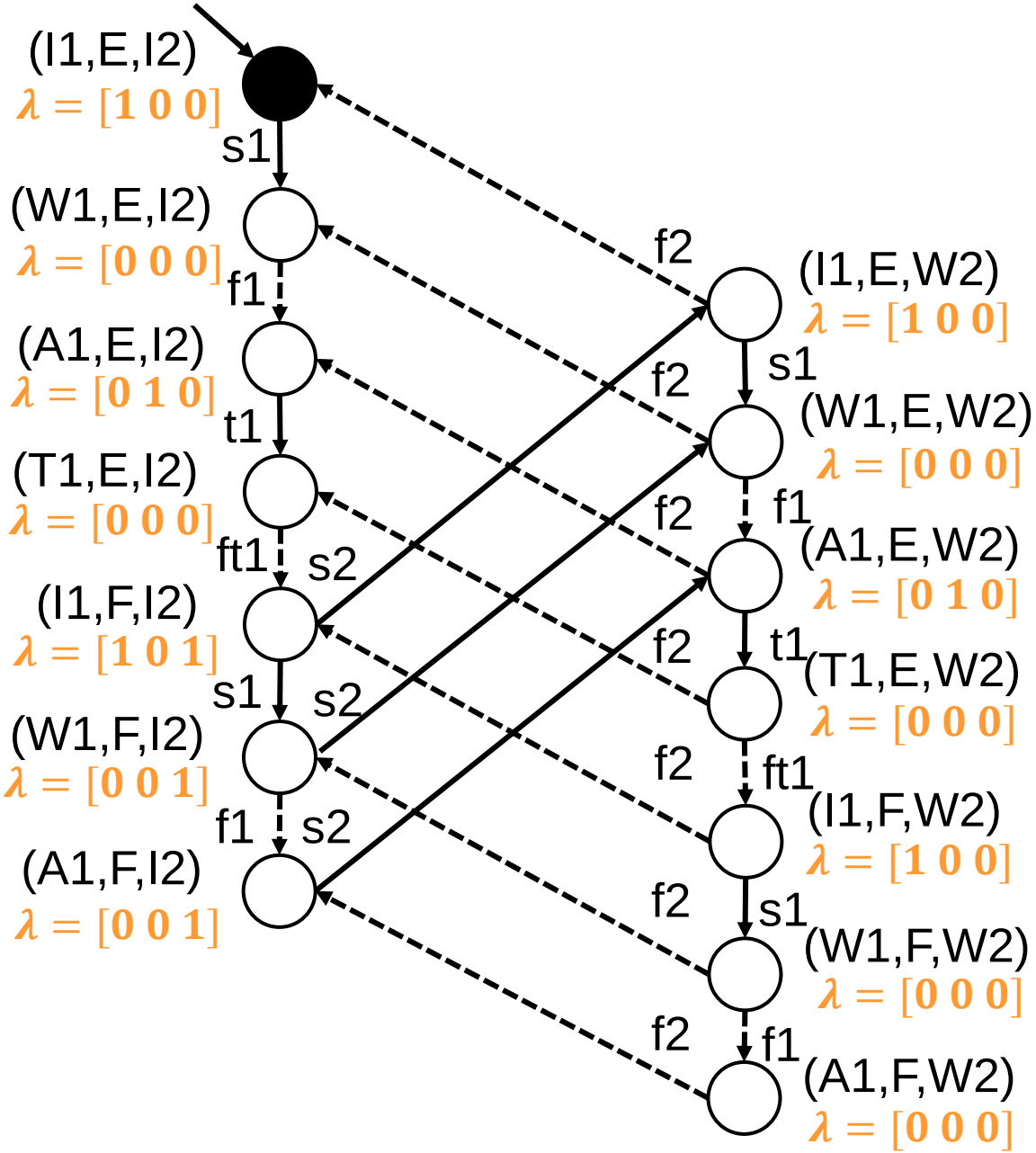
Esempio – controllo a forzamento di eventi



Σ_c: insieme degli eventi controllabili

Σ_u: insieme degli eventi non controllabili

$\Sigma_c \subseteq \Sigma_{co}$: insieme degli eventi generati dal controllore


(I1,E,I2)
$\lambda = [1\ 0\ 0]$
s1
(W1,E,I2)
$\lambda = [0\ 0\ 0]$
f1
(A1,E,I2)
$\lambda = [0\ 1\ 0]$
t1
(T1,E,I2)
$\lambda = [0\ 0\ 0]$
ft1
s2
(I1,F,I2)
$\lambda = [1\ 0\ 1]$
s1
s2
(W1,F,I2)
$\lambda = [0\ 0\ 1]$
f1
s2
(A1,F,I2)
$\lambda = [0\ 0\ 1]$
f2
(I1,E,W2)
$\lambda = [1\ 0\ 0]$
s1
f2
(W1,E,W2)
$\lambda = [0\ 0\ 0]$
f1
f2
(A1,E,W2)
$\lambda = [0\ 1\ 0]$
t1
f2
(T1,E,W2)
$\lambda = [0\ 0\ 0]$
ft1
f2
(I1,F,W2)
$\lambda = [1\ 0\ 0]$
s1
f2
(W1,F,W2)
$\lambda = [0\ 0\ 0]$
f1
f2
(A1,F,W2)
$\lambda = [0\ 0\ 0]$

Controllo con automi

Esempio – controllo a forzamento di eventi

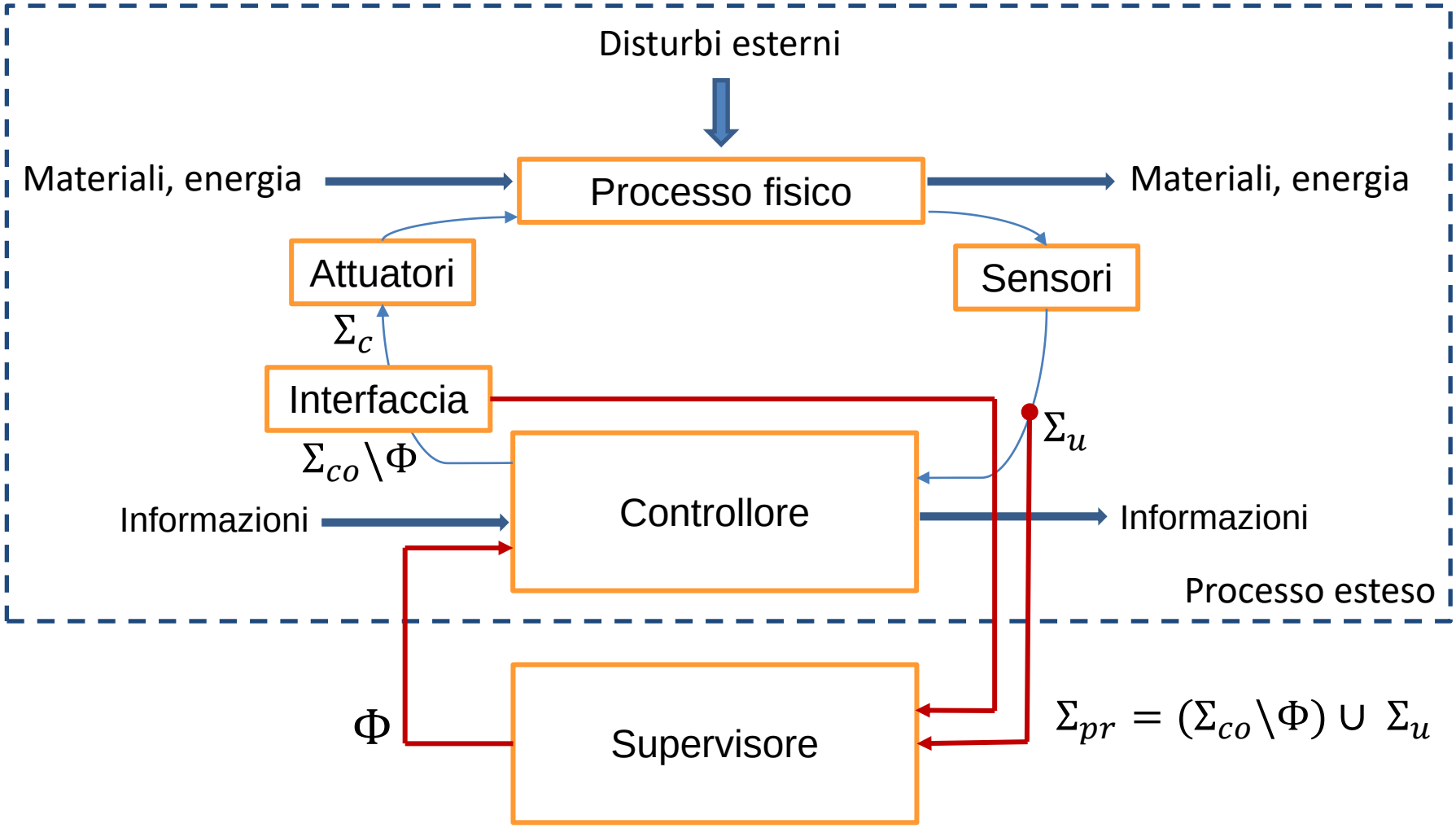
- In questo caso, le uscite dell'automa sono da interpretare come **azioni direttamente effettuate sul processo**

- L'approccio a **forzamento di eventi** comporta alcuni svantaggi:

  – Non c'è **separazione** tra le varie parti che compongono il processo
  – L'automa risultante può essere **complesso**

Controllo con automi

Esempio – controllo di supervisione

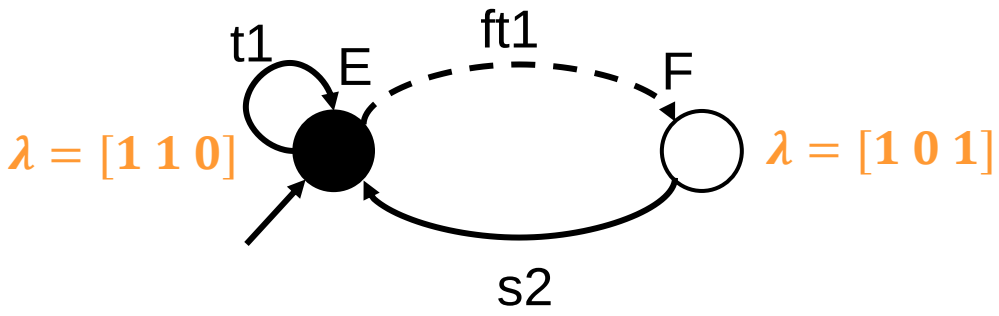


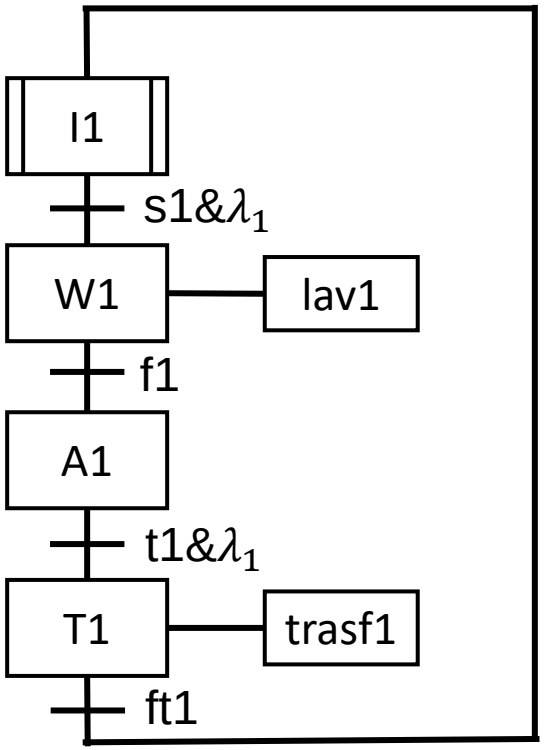
Φ: insieme degli eventi da disabilitare

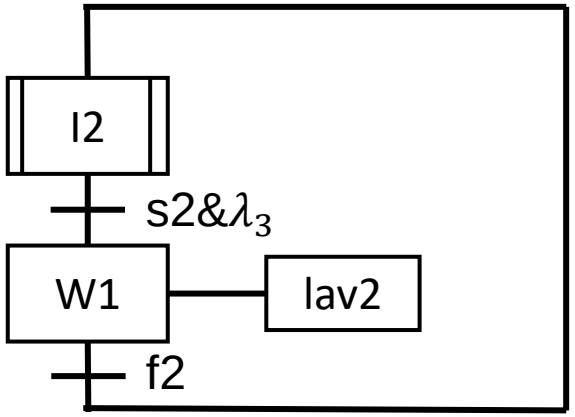
Controllo con automi
Esempio – controllo di supervisione

- Con l'approccio del **controllo supervisivo**, avrò ancora una macchina di Moore le cui uscite rappresentano quali eventi controllabili sono abilitati/disabilitati

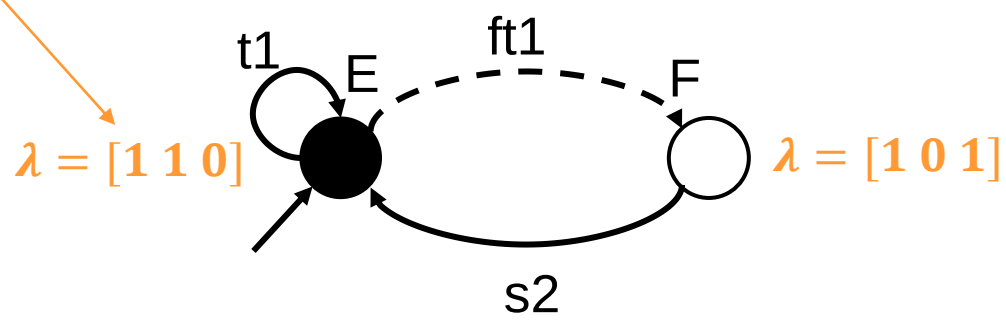
- Le transizioni negli automi che rappresentano i vari processi saranno condizionate alle uscite dell'automa supervisore

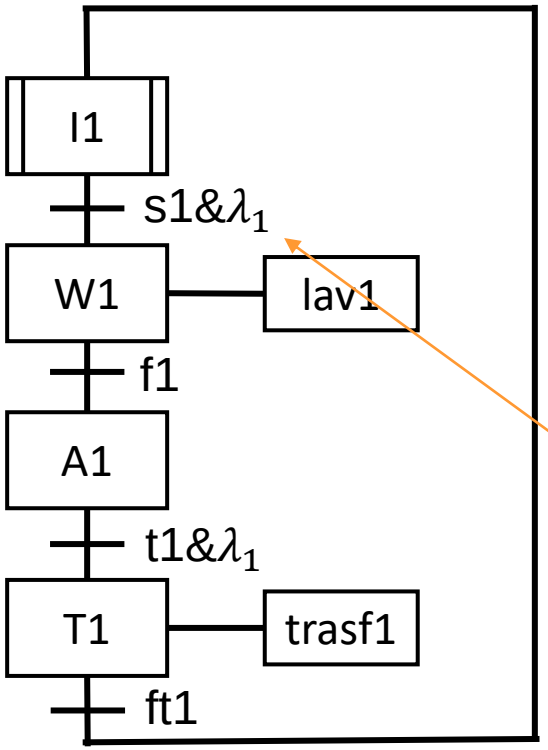

t1
E
ft1
F
$\lambda = [1\ 1\ 0]$
$\lambda = [1\ 0\ 1]$
s2

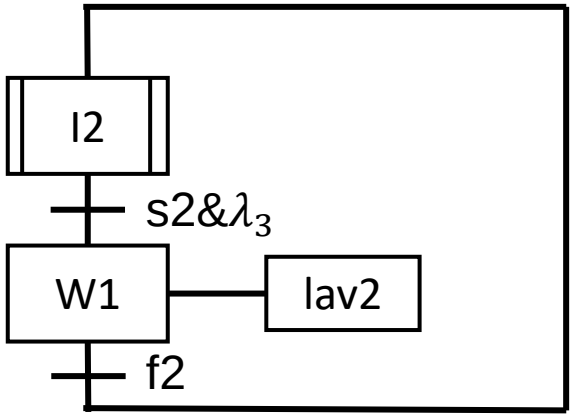

I1
s1&λ_1
W1
lav1
f1
A1
t1&λ_1
T1
trasf1
ft1


I2
s2&λ_3
W1
lav2
f2

s1 e t1 abilitati, s2 disabilitato







Posso passare da I1 a W2 solo se l'evento s1 è abilitato dal supervisore

Controllo con automi
Esempio – controllo di supervisione

- Questa soluzione è molto più efficiente della precedente
  - Gli stati del supervisore passano da 14 a 2!
- Inoltre, c'è una chiara distinzione tra eventi generati dal processo ed eventi necessari alla supervisione
- Gli SFC di controllo per M1 e M2 restano praticamente immutati
  - Nel caso del controllo a forzamento di eventi, devo calcolare un nuovo automa per tutto il sistema di controllo

Automi a stati finiti
Svantaggi

- La modellistica mediante automi a stati finiti può portare a modelli **complessi**, soprattutto se sono presenti **molti stati** possibili

- Non consentono di **separare le diverse entità fisiche** del modello

- Non consentono di descrivere sistemi con un **numero di stati infinito**

→ Nella prossima lezione, vedremo come è possibile superare alcuni di questi problemi utilizzando le **reti di Petri**

Esercizi

Realizzare un sistema di controllo (a forzamento di eventi e di supervisione) per l'esempio delle due macchine con buffer quando il buffer è di capacità 3

Risorse e Riferimenti

- [1] § 8.1-8.3
- [3] § 16.1- 16.3

Fine Lezione #19

Introduzione ai Sistemi a Eventi Discreti