Dipartimento di Elettronica e Informazione
POLITECNICO DI MILANO

Automazione industriale
dispense del corso

5. Automi a stati finiti

Luigi Piroddi
piroddi@elet.polimi.it

Definizione e tipologie

Un automa (autonomo) è definito da una quadrupla di entità matematiche

$A = (X, E, f, x_0)$, dove:

- $X = \{x_0, x_1, x_2, x_3, \dots\}$ è l’insieme degli *stati*
- $E = \{e_1, e_2, e_3, \dots\}$ è l’insieme degli *eventi*
- $f : X \times E \rightarrow X$ è la *funzione di transizione* (o funzione dello stato prossimo) che ad ogni coppia (stato, evento) associa il prossimo stato dell’automa
  f può non essere definita per tutte le coppie $(x, e) \in X \times E$
  un evento e si dice ammissibile in x se $f(x, e)$ è definita
- x_0 è lo *stato iniziale*

A volte viene aggiunto anche un quinto argomento, $X_m \subseteq X$, che rappresenta il sottoinsieme di stati finali (*marcati*), che hanno un significato particolare nell’interpretazione del modello.

Ad esempio, se $X_m = \{x_0\}$ attribuiamo un valore particolare al ritorno nello stato iniziale (completamento di un ciclo produttivo).

L'evoluzione dell'automa è determinata dalla funzione di transizione:

- se l'automa si trova nello stato corrente x_i e l'evento e_k è accettabile da x_i (ovvero la coppia (x_i, e_k) appartiene al dominio di f), allora l'occorrenza dell'evento e_k fa cambiare stato all'automa e il nuovo stato è $f(x_i, e_k)$; altrimenti, non succede nulla
- ipotesi di *indivisibilità*: quando accade un evento che genera un cambio di stato, la transizione di stato si deve completare prima che accada un altro evento
- l'eventuale *rifiuto* di un evento (ovvero quando accade un evento che non è ammissibile nello stato corrente), può essere indice di qualche problema modellistico!
- ipotesi di *non concorrenza* degli eventi: due eventi non possono mai accadere contemporaneamente (a meno che non siano correlati)

Nella definizione di automa autonomo non si distingue tra variabili di ingresso e variabili di uscita (l'evoluzione non dipende da variabili esogene).

Gli eventi sono indistinguibili da questo punto di vista.

Un automa a stati finiti con ingressi e uscite è definito dalla sestupla

$A = (X, U, Y, f, h, x_0)$, dove:

- X e x_0 sono definiti come in precedenza
- $U = \{u_1, u_2, u_3, \ldots\}$ è l'insieme degli *eventi in ingresso*
- $Y = \{y_1, y_2, y_3, \ldots\}$ è l'insieme degli *eventi in uscita*
- $f : X \times U \to X$ è la *funzione di transizione* dello stato
- $h : X \times U \to Y$ è la *funzione di aggiornamento dell'uscita*

A una generica transizione dallo stato x_l allo stato x_k è associata una coppia (u_i, y_j) di eventi. L'interpretazione corrispondente è la seguente:

- se, quando l'automa si trova nello stato x_l, viene ricevuto l'evento di ingresso u_i, allora l'automa compirà la transizione allo stato x_k
- nel corso della transizione emetterà l'evento di uscita y_j

Tipi di automi:

- se l’insieme degli stati X è finito, l’automa si dice *a stati finiti* (*finite state machine*, *FSM*), altrimenti *a stati infiniti*
- se la funzione $f(\cdot,\cdot)$ è a più valori, allora l’automa si dice *non deterministico*, altrimenti *deterministico*

Ai fini del progetto del controllore logico, normalmente si usano automi *deterministici a stati finiti* (non avrebbe senso progettare un controllore non deterministico). Gli automi non deterministici si utilizzano nella valutazione delle prestazioni.

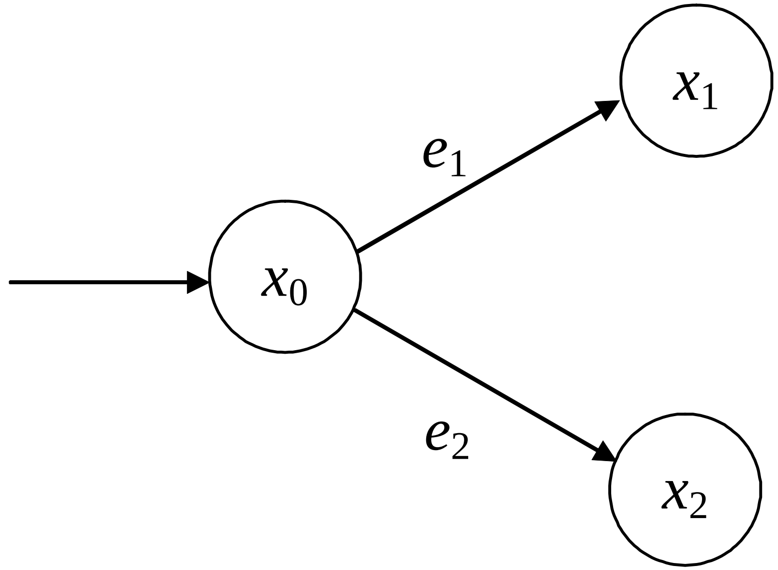
Automi di Mealy e di Moore:

- se la funzione di aggiornamento dell’uscita dipende sia dallo stato che dall’ingresso (come in un sistema dinamico non strettamente proprio) si parla di *automi di Mealy* $(h : X \times U \rightarrow Y)$
- se invece la funzione di aggiornamento dell’uscita dipende solo dallo stato e non dall’ingresso (sistema dinamico strettamente proprio), si parla di *automi di Moore* $(h : X \rightarrow Y)$
- la trasformazione di un automa di Moore in uno di Mealy (e viceversa) è sempre possibile, ma può comportare una complicazione della rappresentazione dell’automa

Rappresentazioni

Gli automi sono facilmente rappresentabili con un grafo orientato in cui:

- nodo $\leftrightarrow$ stato
- arco dal nodo x_i al nodo x_j $\leftrightarrow$ $x_j = f(x_i, e_k)$
- etichetta dell'arco dal nodo x_i al nodo x_j $\leftrightarrow$ e_k



x_0 = stato iniziale

$x_1 = f(x_0, e_1)$

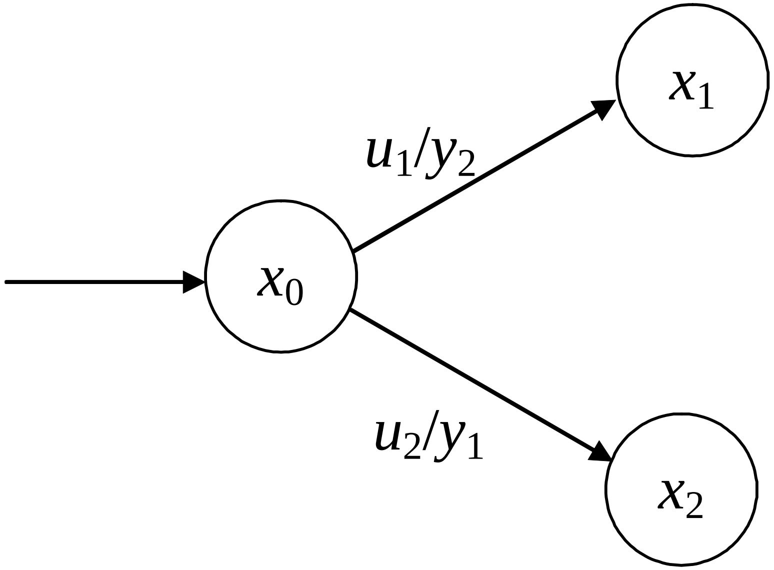
$x_2 = f(x_0, e_2)$

Una rappresentazione alternativa degli automi è quella in forma tabellare, nelle seguenti due versioni tipiche:

	evento			
stato	e_1	e_2	...	e_m
x_1	$f(x_1, e_1)$	$f(x_1, e_2)$	...	$f(x_1, e_m)$
x_2	$f(x_2, e_1)$	$f(x_2, e_2)$	...	$f(x_2, e_m)$
...	...	...	...	...
x_n	$f(x_n, e_1)$	$f(x_n, e_2)$	...	$f(x_n, e_m)$

	stato prossimo			
stato corrente	x_1	x_2	...	x_n
x_1	-	e_2	...	e_k
x_2	e_4	e_4	...	-
...	...	...	...	...
x_n	e_m	e_4	...	e_i

Rappresentazioni grafica e tabellare nel caso di automi con ingressi e uscite:



x_0 = stato iniziale

$x_1 = f(x_0, u_1)$

$x_2 = f(x_0, u_2)$

$y_1 = h(x_0, u_2)$

$y_2 = h(x_0, u_1)$

	evento			
stato	u_1	u_2	...	u_m
x_1	$f(x_1, u_1), h(x_1, u_1)$	$f(x_1, u_2), h(x_1, u_2)$	...	$f(x_1, u_m), h(x_1, u_m)$
x_2	$f(x_2, u_1), h(x_2, u_1)$	$f(x_2, u_2), h(x_2, u_2)$	...	$f(x_2, u_m), h(x_2, u_m)$
...	...	...	...	...
x_n	$f(x_n, u_1), h(x_n, u_1)$	$f(x_n, u_2), h(x_n, u_2)$	...	$f(x_n, u_m), h(x_n, u_m)$

Modellizzazione con automi

Procedimento di modellizzazione diretta:

➊ elencare i componenti principali del sistema (macchine, manipolatori, nastri, ecc.)

➋ definire gli stati principali di questi componenti

➌ definire l’alfabeto di eventi ingresso/uscita associati all’evoluzione dello stato dei componenti

➎ definire gli stati del sistema complessivo, combinando gli stati dei componenti (non tutti i possibili stati ottenibili in questo modo sono poi effettivamente raggiunti: v. punto successivo)

➏ costruire il modello completo aggiungendo le transizioni opportune tra questi stati

➐ associare alle transizioni opportune funzioni degli ingressi e delle uscite

Questo metodo risulta poco agevole e facilmente soggetto a errori anche per sistemi di piccole/medie dimensioni (v. esempio seguente).

In alternativa, si possono modellizzare separatamente i vari componenti, per poi costruire il modello completo mediante il procedimento di *composizione sincrona* (o *composizione concorrente*).

Composizione sincrona

Dati due automi $A_1 = (X_1, E_1, f_1, x_{10})$ e $A_2 = (X_2, E_2, f_2, x_{20})$, si definisce *composizione sincrona* o *concorrente* l'automa $A = A_1 \| A_2 = (X, E, f, x_0)$, definito da:

- $X \subseteq X_1 \times X_2$ (X è costituito da coppie (x_1, x_2), con $x_1 \in X_1$, $x_2 \in X_2$, ma non nec. tutte)
- $E = E_1 \cup E_2$ (l'alfabeto dell'automa è l'unione dei due alfabeti di eventi)
- $f : X \times E \rightarrow X$ è definita come

  $f((x_1, x_2), e) = (x_1', x_2)$, se $e \in E_1$, $e \notin E_2$, e $f_1(x_1, e) = x_1'$

  $f((x_1, x_2), e) = (x_1, x_2')$, se $e \in E_2$, $e \notin E_1$, e $f_2(x_2, e) = x_2'$

  $f((x_1, x_2), e) = (x_1', x_2')$, se $e \in E_1 \cap E_2$, $f_1(x_1, e) = x_1'$ e $f_2(x_2, e) = x_2'$

  e non è definita altrimenti (un evento (di ingresso) può scattare nell'automa composto se può scattare in tutti gli automi *nel cui alfabeto compare*)
- lo stato iniziale $x_0 = (x_{10}, x_{20})$ è la combinazione degli stati iniziali dei singoli automi
- se gli automi A_1 e A_2 includono anche la definizione degli insieme di stati marcati X_{m1} e X_{m2}, l'automa composto è caratterizzato da $X_m \subseteq X_{m1} \times X_{m2}$ (non tutte le coppie di stati marcati dei singoli automi possono essere raggiunte dall'automa composto)

Procedimento di modellizzazione per composizione sincrona:

❶ elencare i componenti principali del sistema (macchine, manipolatori, nastri, ecc.)

❷ definire gli stati principali di questi componenti

❸ definire *per ogni componente* l’alfabeto di eventi ingresso/uscita associati all’evoluzione dello stato

❹ costruire gli automi relativi ai *singoli* componenti, aggiungendo le transizioni opportune tra gli stati definiti per ogni componente

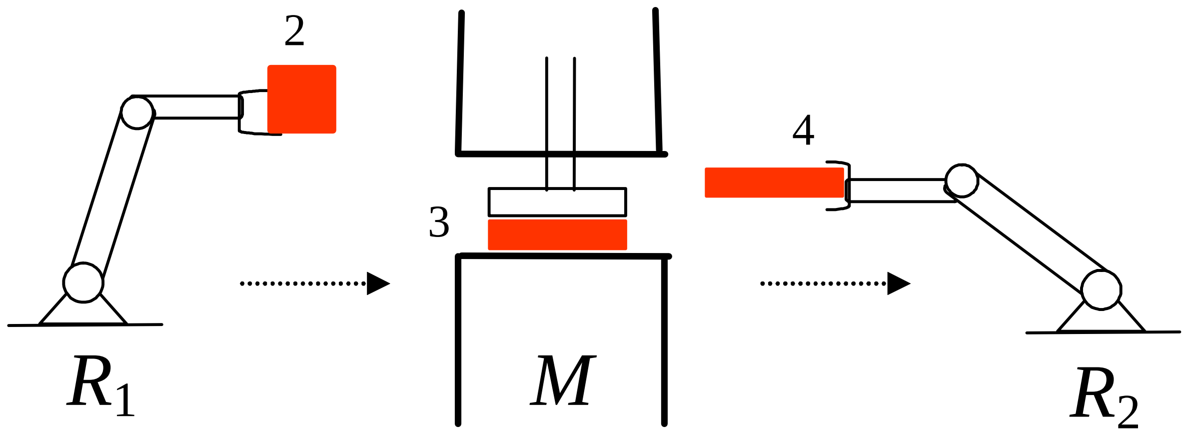
❺ costruire l’automa complessivo per composizione sincrona

Commenti:

- gli stati dell’automa sono stati *globali* del sistema da rappresentare
- le transizioni rappresentano variazioni dello stato globale
- il modello ad automi non è *modulare*
  - poiché lo stato è globale, una qualunque modifica del modello, relativa anche ad un solo componente (p.es. l’aggiunta o la modellizzazione di dettaglio di un componente) implica il cambiamento di tutti gli stati dell’automa!
- questo obbliga a considerare sempre il modello nel suo insieme

Esempio: modellizzazione logica di una macchina

Immaginiamo che una macchina *M* venga caricata con un pezzo grezzo da lavorare mediante un robot manipolatore R_1, che effettui una lavorazione specifica, al termine della quale un altro robot (R_2) prelevi il prodotto finito.



Vogliamo modellizzare con un automa (di Mealy) il gestore logico del ciclo di lavorazione della macchina. Il modello più semplice introduce due stati per ogni componente del sistema (manipolatori robotici e macchina):

- il componente è *occupato* in una lavorazione o in un trasporto
- il componente è *disponibile*, in attesa di nuove lavorazioni

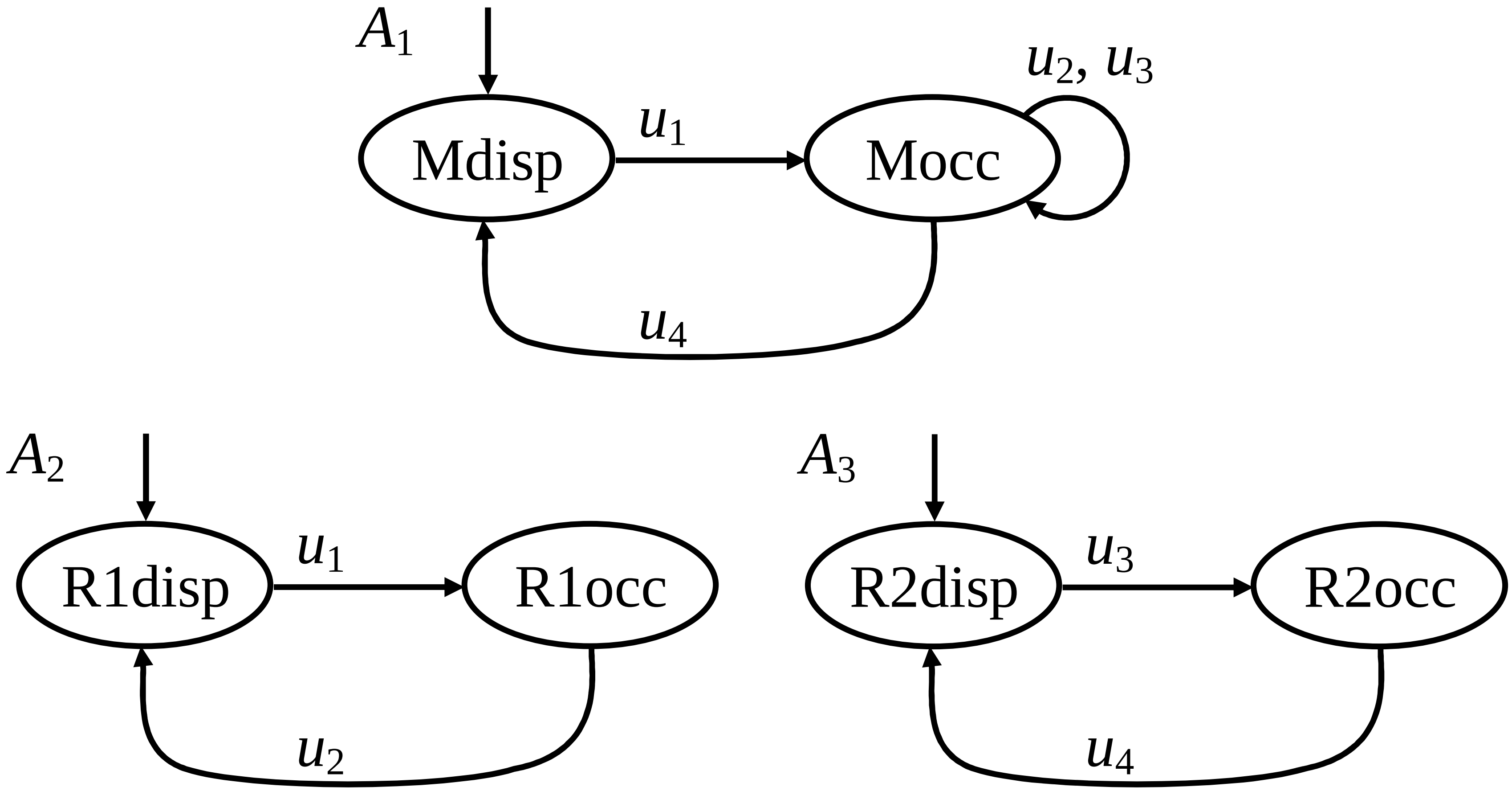
Gli stati corrispondenti ai vari componenti del sistema sono:

componente del sistema	stati
macchina M	Mdisp, Mocc
robot R_1	R1disp, R1occ
robot R_2	R2disp, R2occ

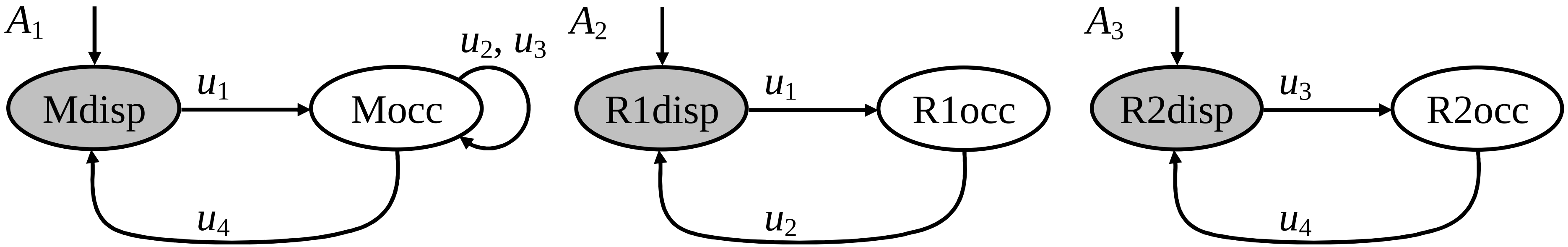
Gli eventi (di ingresso) sono:

	simbolo	significato
ingressi	u_1	inizio ciclo (fine attesa)
	u_2	fine carico
	u_3	fine lavorazione
	u_4	fine scarico

Esempio: modellizzazione sintetica errata (dettaglio insufficiente)



Stato iniziale x_1 ({Mdisp, R1disp, R2disp})

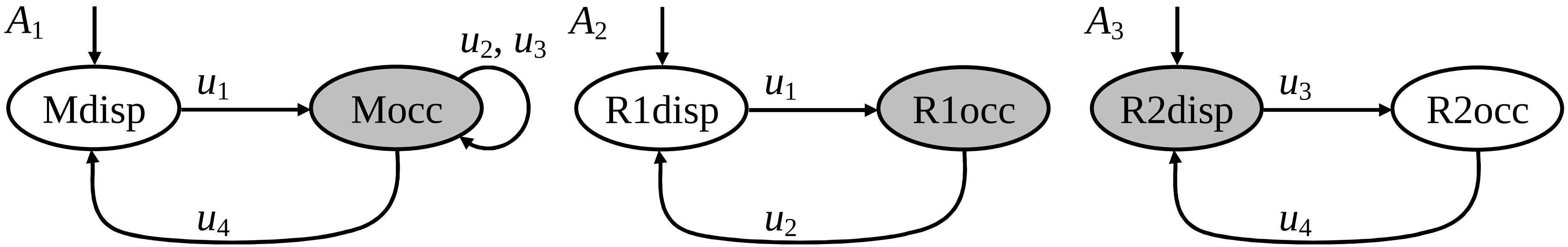


- l’unico evento accettabile è u_1

eventi accettabili da	u_1	u_2	u_3	u_4
A_1	✓	✗	✗	✗
A_2	✓	✗		
A_3			✓	✗
automa complessivo	✓	✗	✗	✗

- *M* passa nello stato occupato così come R_1, mentre R_2 rimane disponibile (stato x_2)

Stato x_2 ({Mocc, R1occ, R2disp})

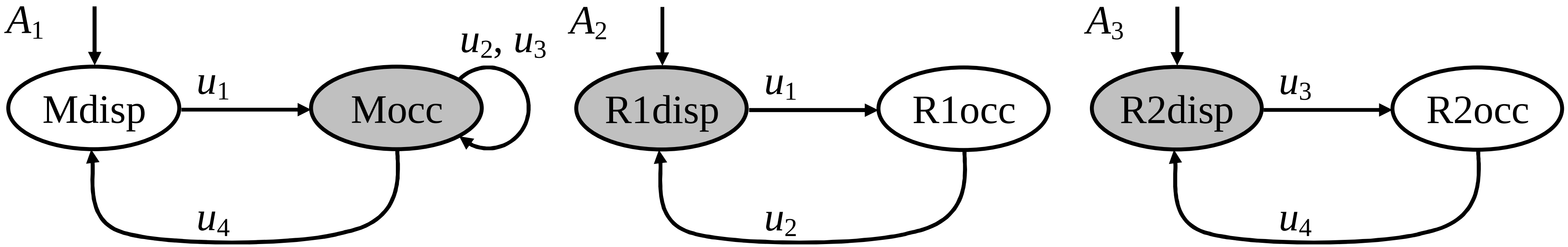


- sono accettabili sia u_2 che u_3

eventi accettabili da	u_1	u_2	u_3	u_4
A_1	✗	✓	✓	✓
A_2	✗	✓		
A_3			✓	✗
automa complessivo	✗	✓	✓	✗

- se scatta u_2 cambia stato solo R_1, che torna disponibile (stato x_3)
- se scatta u_3 cambia stato solo R_2, che diventa occupato (stato x_5)
  - equivale a iniziare l’operazione di scarico prima che sia arrivata l’informazione di corretta conclusione della fase di carico e prima della lavorazione!

Stato x_3 ({Mocc, R1disp, R2disp})

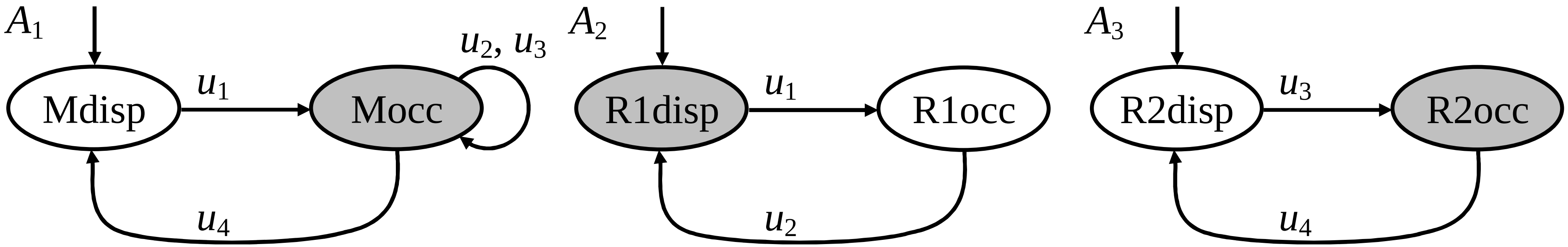


- l’unico evento accettabile da tutti e 3 gli automi è u_3

eventi accettabili da	u_1	u_2	u_3	u_4
A_1	✗	✓	✓	✓
A_2	✓	✗		
A_3			✓	✗
automa complessivo	✗	✗	✓	✗

- cambia stato solo R_2, che diventa occupato (stato x_4)

Stato x_4 ({Mocc, R1disp, R2occ})

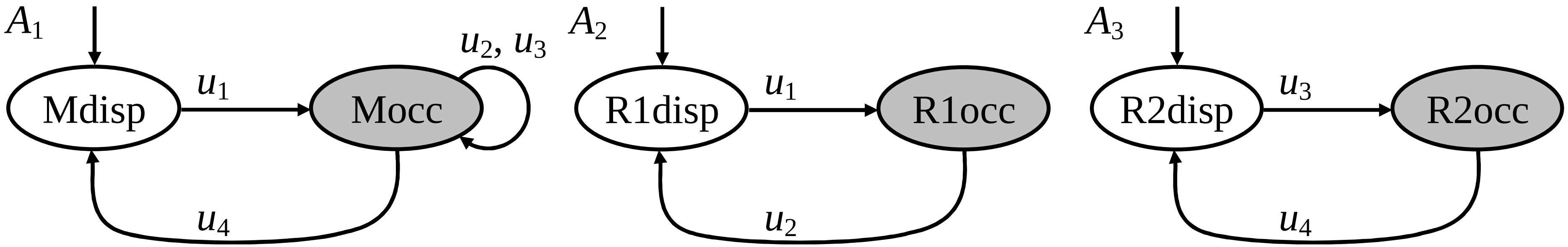


- l'unico evento accettabile da tutti e 3 gli automi associati al sistema è u_4

eventi accettabili da	u_1	u_2	u_3	u_4
A_1	✗	✓	✓	✓
A_2	✓	✗		
A_3			✗	✓
automa complessivo	✗	✗	✗	✓

- M e R_2 tornano disponibili (si ritorna nello stato x_1)

Stato x_5 ({Mocc, R1occ, R2occ})

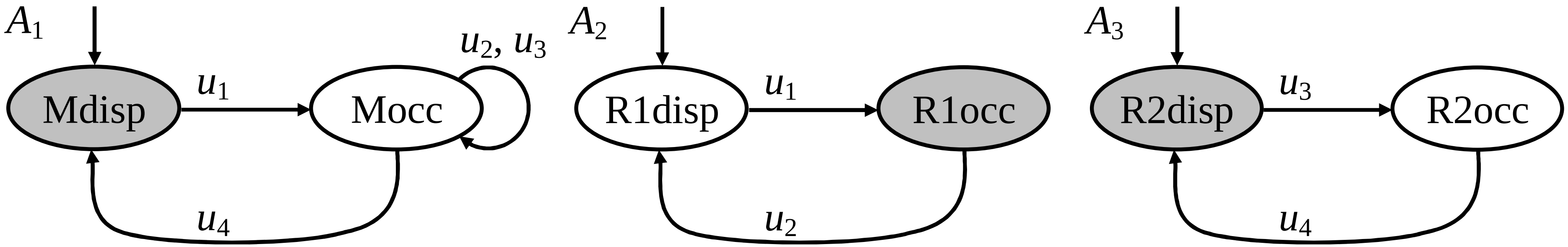


- sono accettabili sia u_2 che u_4

eventi accettabili da	u_1	u_2	u_3	u_4
A_1	✗	✓	✓	✓
A_2	✗	✓		
A_3			✗	✓
automa complessivo	✗	✓	✗	✓

- se scatta u_2 cambia stato solo R_1, che torna disponibile (stato x_4)
- se scatta u_4 cambiano stato M e R_2, che tornano disponibili (stato x_6)

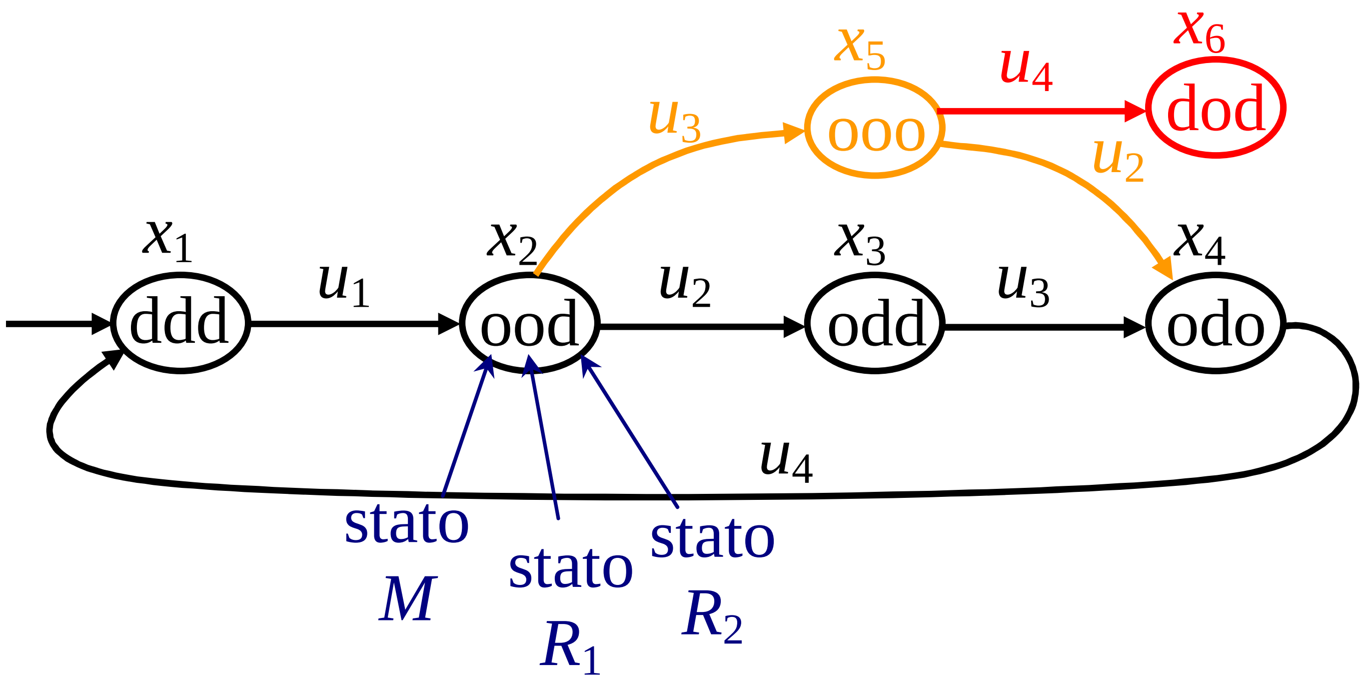
Stato x_6 ({Mdisp, R1occ, R2disp})



- non è accettabile alcun evento (stato *morto* o *deadlock*)

eventi accettabili da	u_1	u_2	u_3	u_4
A_1	✓	✗	✗	✗
A_2	✗	✓		
A_3			✓	✗
automa complessivo	✗	✗	✗	✗

La rappresentazione grafica dell’automa è la seguente:

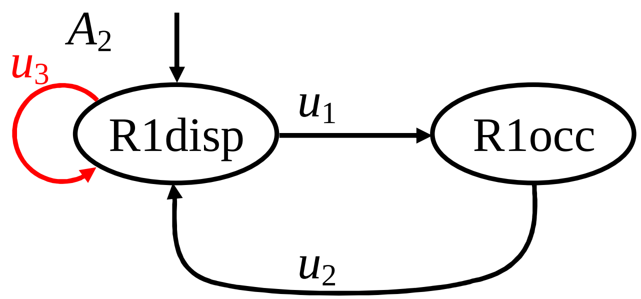


Gli stati x_5 e x_6 sono chiaramente spurii e indesiderati.

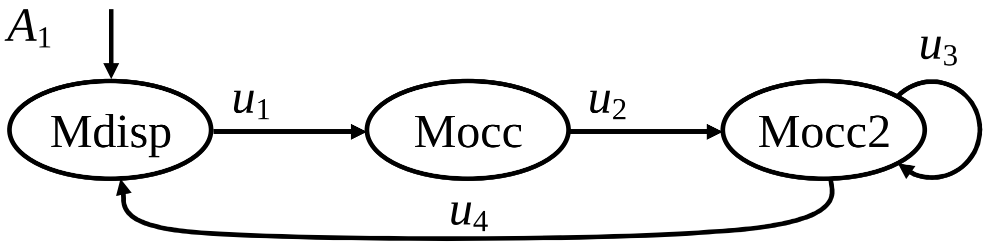
Il problema nasce dal fatto che lo stato Mocc rappresenta in modo troppo compatto il funzionamento della macchina e non riporta correttamente la precedenza logica tra u_2 e u_3 (la sequenza corretta è carico – lavorazione – scarico).

Si può ovviare in due modi diversi:

❶ estendere l'alfabeto di A_2 o A_3, come per esempio in figura, per evitare che l'evento u_3 risulti accettabile in x_2 (ma è un artificio, perché u_3 non dovrebbe avere alcuna rilevanza per il comportamento del robot R_1)

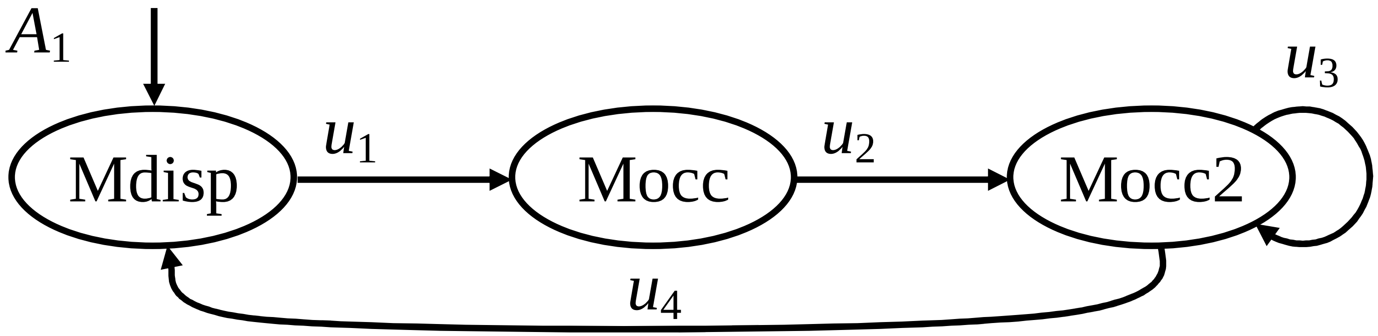


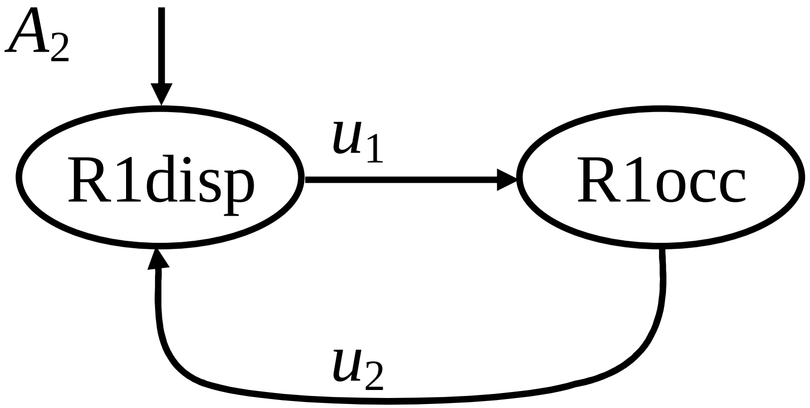
❷ dettagliare lo stato di occupato della macchina M, correggendo anche il relativo automa, p.es. come in figura

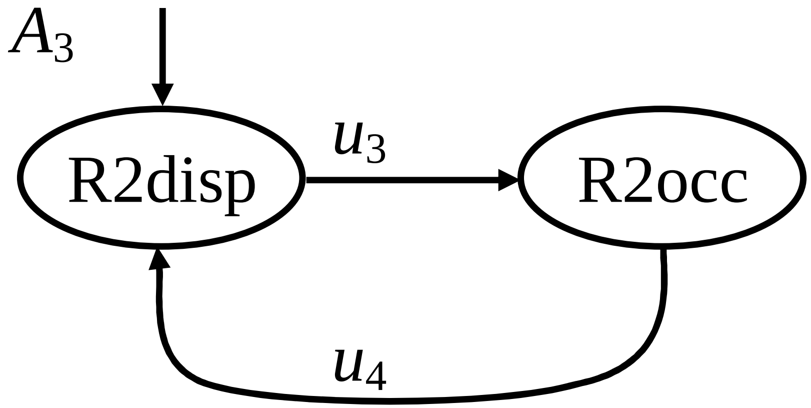


NB. Questo modello non riporta correttamente la precedenza logica tra u_3 e u_4, ma ci pensa il modello A_3 a garantire tale precedenza nel modello complessivo.

Esempio: modellizzazione sintetica corretta





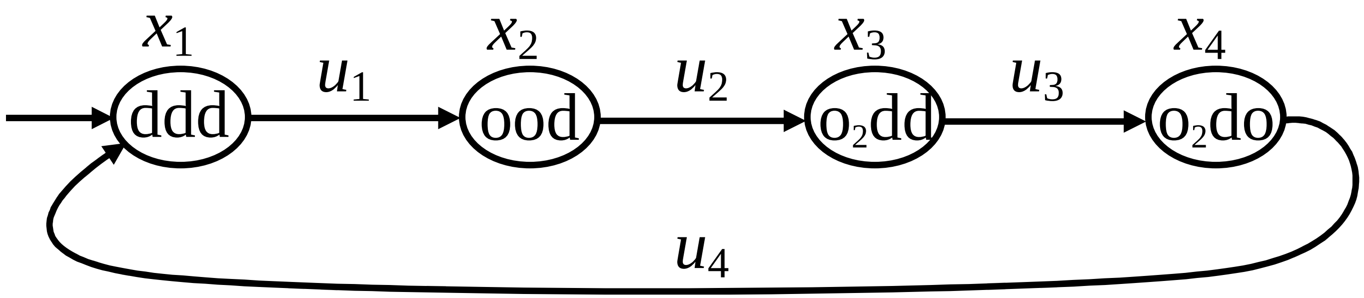


NB. Si poteva scegliere di rappresentare A_1 con 4 stati, dettagliando ulteriormente il significato di "macchina occupata", ma il numero minimo di stati che consente di rappresentare correttamente il comportamento del sistema è pari a 3.

Evoluzione:

- stato iniziale x_1 ({Mdisp, R1disp, R2disp})
  - l’unico evento accettabile è u_1 (può scattare in A_1 e A_2 e non compare nell’alfabeto di A_3)
  - M passa nello stato occupato così come R_1, mentre R_2 rimane disponibile (stato x_2)
  - non può invece scattare u_3 che non è abilitato in A_1
- stato x_2 ({Mocc, R1occ, R2disp})
  - è accettabile solo u_2
  - se scatta u_2 cambiano stato R_1, che torna disponibile, e M che passa nello stato Mocc2 (stato x_3)
- stato x_3 ({Mocc2, R1disp, R2disp})
  - l’unico evento accettabile da tutti e 3 gli automi è u_3
  - cambia stato solo R_2, che diventa occupato (stato x_4)
- stato x_4 ({Mocc2, R1disp, R2occ})
  - l’unico evento accettabile da tutti e 3 gli automi associati al sistema è u_4
  - M e R_2 tornano disponibili (si ritorna nello stato x_1)

L’automa che descrive il comportamento complessivo del gestore logico ha solo 4 stati:



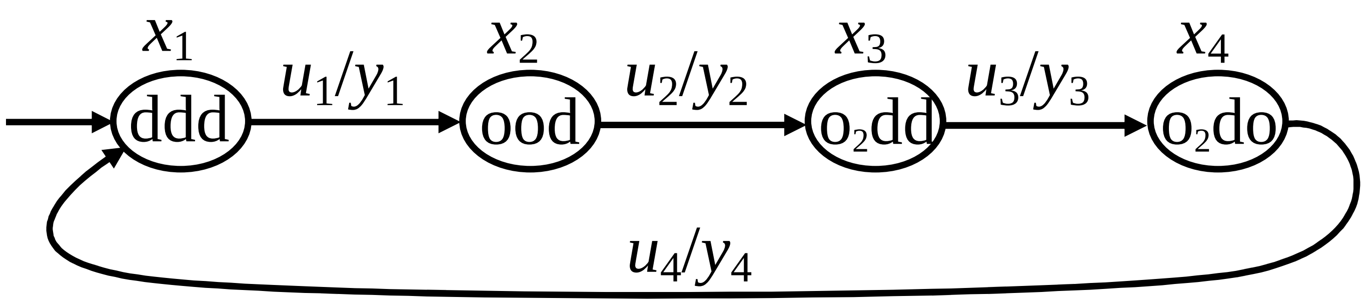
che corrispondono alle seguenti condizioni di funzionamento dei componenti:

stato	stato di M	stato di R_1	stato di R_2	codifica	descrizione
x_1	Mdisp	R1disp	R2disp	ddd	M in attesa di lavorazione
x_2	Mocc	R1occ	R2disp	ood	R_1 carica M
x_3	Mocc2	R1disp	R2disp	o_2dd	M lavora
x_4	Mocc2	R1disp	R2occ	o_2do	R_2 scarica M

L'automa va completato con gli eventi di uscita (comandi):

	simbolo	significato
uscite	y_1	inizio carico
	y_2	inizio lavorazione
	y_3	inizio scarico
	y_4	fine ciclo (inizio attesa)

cosicché esso risulta un automa non autonomo (di Mealy):



Esempio: modellizzazione dettagliata

Le operazioni di trasporto dei robot manipolatori sono rappresentate in forma troppo compatta:

- non si distingue tra effettivo trasporto e operazioni di carico / scarico
- ad esempio, nello stato x_2 la macchina M è occupata inutilmente per gran parte dell’operazione corrispondente (*uso inefficiente delle risorse*)!

Questo non consente nessuna forma di parallelismo tra i vari dispositivi che compongono il sistema, che potrebbe essere invece desiderabile nel caso in cui si volessero gestire più prodotti contemporaneamente.

Occorre quindi:

- dettagliare gli stati dei dispositivi
- consentire parallelismo

NB. Quando due componenti svolgono operazioni concorrenti (p.es. un robot si sposta mentre la macchina lavora) non si sa a priori chi finisce prima! Di conseguenza, sono necessari più stati per rappresentare tutti i possibili funzionamenti.

$\Rightarrow$ *l’uso di operazioni concorrenti determina una moltiplicazione degli stati*

Scriviamo ora un modello più dettagliato del sistema introducendo i seguenti stati per i componenti del sistema:

componente del sistema	stati
macchina M	Mdisp, Mcar, Mlav, Msca
robot R_1	R1disp, R1pre, R1car, R1rit
robot R_2	R2disp, R2sca, R2rit

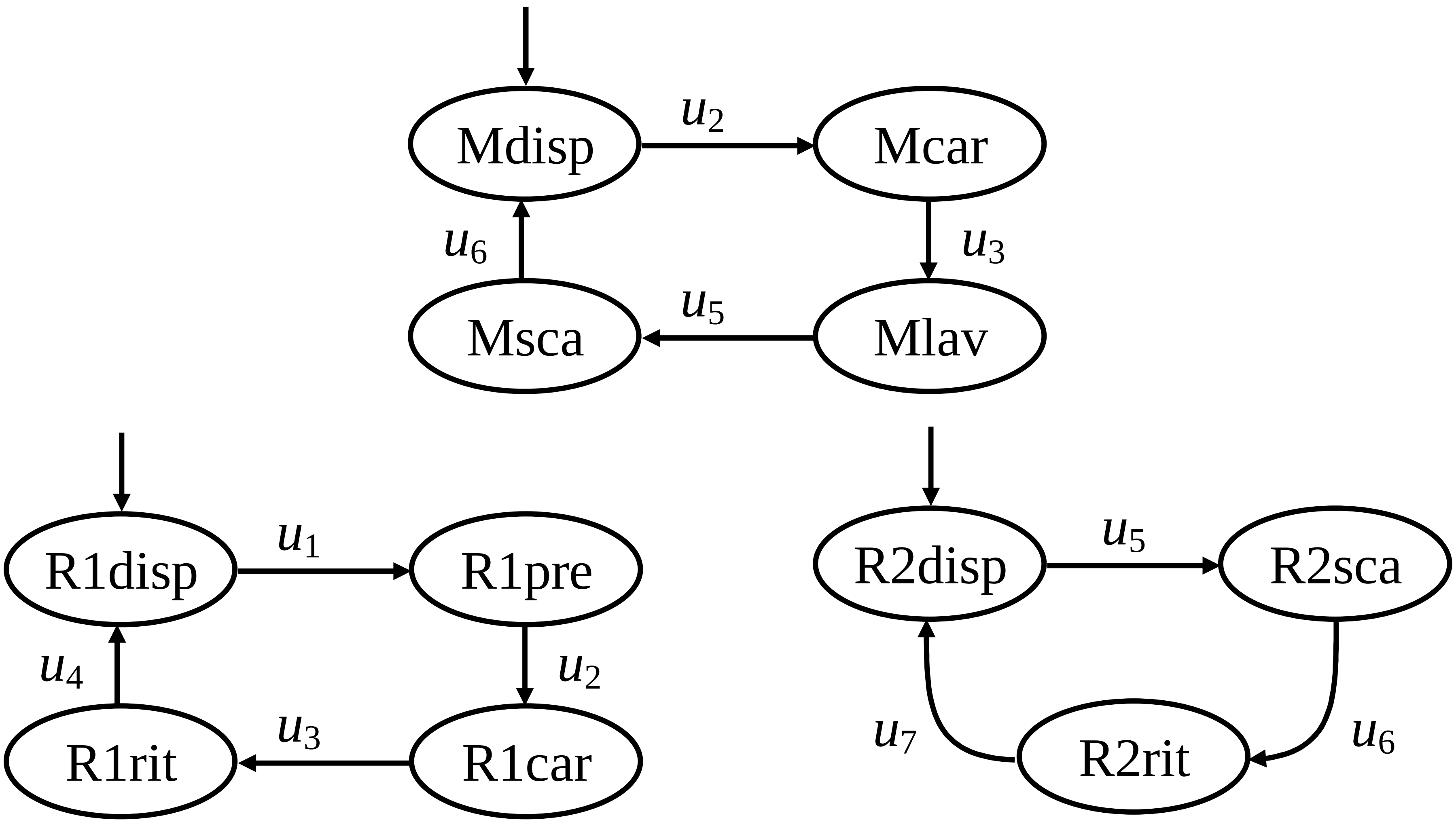
- il robot R_1 preleva un pezzo (R1pre), lo carica su M (R1car), e infine torna nella posizione di riposo (R1rit)
  - solo nello stato R1car è necessario che ci sia sincronismo tra R_1 e M
  - negli altri stati, il funzionamento di R_1 è indipendente da quello degli altri componenti
- il robot R_2 è inizialmente in attesa di un pezzo lavorato su M da prelevare; quando ce n’è uno disponibile, R_2 scarica M (R2sca); successivamente lo porta alla coda di uscita, dove lo rilascia; infine torna nella posizione di riposo (R2rit)
  - solo nello stato R2sca è necessario che ci sia sincronismo tra R_2 e M

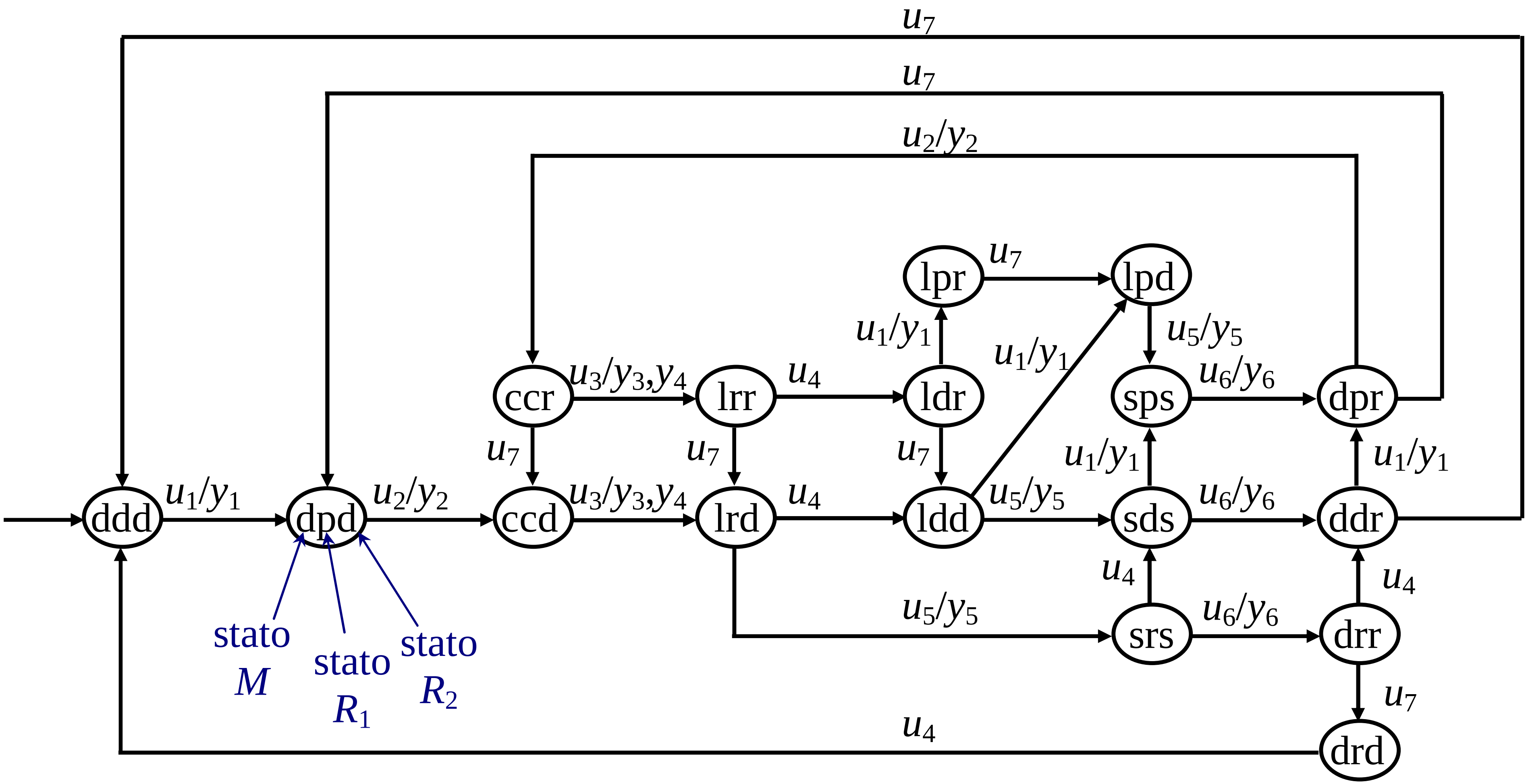
Gli eventi (di ingresso) necessari a gestire questi stati aggiuntivi crescono di conseguenza:

	simbolo	significato
ingressi	u_1	pezzo disponibile per prelievo
	u_2	fine prelievo pezzo con R_1
	u_3	fine carico di M con R_1
	u_4	fine ritorno R_1
	u_5	fine lavorazione M
	u_6	fine scarico di M con R_2
	u_7	fine ritorno R_2

Componendo i 3 sotto-modelli dettagliati, il modello si complica notevolmente rispetto a prima: gli stati sono dell’ordine di $4\times4\times3 = 48$ (quelli effettivamente raggiungibili sono 17).

Inoltre, non c’è alcun legame tra il modello precedente e questo: bisogna generare l’automa da zero!


Mdisp
Mcar
Mlav
Msca
u_2
u_3
u_5
u_6
R1disp
R1pre
R1car
R1rit
u_1
u_2
u_3
u_4
R2disp
R2sca
R2rit
u_5
u_6
u_7



LEGENDA: stato x_i = jkl dove j (= d, c, l, s), k (= d, p, c, r) ed l (= d, s, r) sono associati agli stati di M, R_1, R_2, rispettivamente.

Nel grafico sono stati aggiunti i comandi riportati nella tabella seguente, trasformando l’automa autonomo in un automa non autonomo (di Mealy).

	simbolo	significato
uscite	y_1	inizio prelievo pezzo con R_1
	y_2	inizio carico di M con R_1
	y_3	inizio ritorno R_1
	y_4	inizio lavorazione M
	y_5	inizio scarico di M con R_2
	y_6	inizio ritorno R_2

stato	stato di M	stato di R_1	stato di R_2	codifica	descrizione
x_1	Mdisp	R1disp	R2disp	ddd	M in attesa di lavorazione
x_2	Mdisp	R1pre	R2disp	dpd	R_1 preleva un pezzo
x_3	Mcar	R1car	R2disp	ccd	R_1 carica M
x_4	Mlav	R1rit	R2disp	lrd	M lavora R_1 torna in posizione di riposo
x_5	Mlav	R1disp	R2disp	ldd	M lavora
x_6	Msca	R1disp	R2sca	sds	R_2 scarica M
x_7	Mdisp	R1disp	R2rit	ddr	R_2 trasporta pezzo e ritorna in posizione di riposo
x_8	Msca	R1rit	R2sca	srs	R_1 torna in posizione di riposo R_2 scarica M
x_9	Mdisp	R1rit	R2rit	drr	R_1 torna in posizione di riposo R_2 trasporta pezzo e ritorna in posizione di riposo
x_{10}	Mdisp	R1rit	R2disp	drd	R_1 torna in posizione di riposo
x_{11}	Msca	R1pre	R2sca	sps	R_1 preleva un pezzo R_2 scarica M
x_{12}	Mdisp	R1pre	R2rit	dpr	R_1 preleva un pezzo R_2 trasporta pezzo e ritorna in posizione di riposo
x_{13}	Mcar	R1car	R2rit	ccr	R_1 carica M R_2 trasporta pezzo e ritorna in posizione di riposo
x_{14}	Mlav	R1rit	R2rit	lrr	M lavora R_1 torna in posizione di riposo R_2 trasporta pezzo e ritorna in posizione di riposo
x_{15}	Mlav	R1disp	R2rit	ldr	M lavora R_2 trasporta pezzo e ritorna in posizione di riposo
x_{16}	Mlav	R1pre	R2rit	lpr	M lavora R_1 preleva un pezzo R_2 trasporta pezzo e ritorna in posizione di riposo
x_{17}	Mlav	R1pre	R2disp	lpd	M lavora R_1 preleva un pezzo

Criticità del problema dimensionale

Modifica 1 – due linee uguali indipendenti

- è facile verificare che l’estensione del modello più semplice (l’automa a 4 stati), non richiede 8 stati (= 4+4), ma 16 (= 4x4), perché occorre combinare insieme tutti i possibili stati dei due sotto-sistemi!

$$\begin{array}{ccccc} R_1 & \dashrightarrow & M_1 & \dashrightarrow & R_2 \\ R_3 & \dashrightarrow & M_2 & \dashrightarrow & R_4 \end{array}$$

Modifica 2 – due linee uguali che condividono soltanto il robot per lo scarico delle macchine

- in questo caso sono necessari 15 stati

$$\begin{tikzcd} R_1 \arrow[r, dashed] & M_1 \arrow[dr, dashed] & \\ & & R_2 \\ R_3 \arrow[r, dashed] & M_2 \arrow[ur, dashed] & \end{tikzcd}$$

Modifica 3 – una linea sola, ma con la macchina con 2 unità

- servono 8 stati invece di 4

$$R_1 \dashrightarrow \underset{\text{2 unità}}{M} \dashrightarrow R_2$$

Vantaggi e svantaggi della modellizzazione con automi

Vantaggi:

- semplicità di definizione (descrizione grafica)
- semplicità di interpretazione (in termini di evoluzione per transizioni di stato)
- semplicità di analisi (è facile capire se ci sono stati di blocco e, in generale, se si finisce in stati desiderati oppure no)

In generale, lo stato di un automa può essere pensato come l’insieme dei valori di *tutte* le variabili di stato, che sono tipicamente gli stati di funzionamento di ogni dispositivo/risorsa del sistema.

Quante più variabili di stato ci sono e quanti più valori hanno, tanti più sono gli stati potenziali (poi, magari, alcuni non vengono mai raggiunti e si possono eliminare).

Ciò comporta i seguenti svantaggi:

- basso potere rappresentativo
  parallelismo, condivisione di risorse, ecc.
- assenza modularità
  anche fare semplici modifiche è complicato e richiede di rifare da zero l’automa
- dimensioni notevoli anche in casi semplici
- stato del sistema = globale (localizzato in un nodo)
  interpretazione (uso dei dispositivi, sequenze di operazioni) = distribuita
  non c’è leggibilità locale, nel senso che non c’è corrispondenza tra le parti di un automa e quelle del sistema fisico

A noi serve uno strumento di modellizzazione con le caratteristiche opposte, che ci consenta di ottenere modelli:

- facilmente modificabili, con l’aggiunta di altre variabili o con la modifica dell’insieme di valori assumibili da una o più variabili, senza che sia necessario rifare tutta la modellizzazione e senza che la complessità esploda
- modulari, ovvero costruiti “assemblando” sottomodelli relativi a parti del sistema fisico (ad esempio le singole macchine di una fabbrica), eventualmente sostituibili
- facilmente interpretabili in termini dell’evoluzione dello stato delle singole parti del sistema fisico $\rightarrow$ stato distribuito nel modello, con interpretazione locale