

# Automazione

12 Febbraio 2018

## Esercizio 1

Si consideri un sistema di automazione operante a livello di coordinamento per il controllo di un robot pick-and-place (PNP1) così caratterizzato:

Task  $A_1$ ) ogni 10 t.u. è eseguito un processo di identificazione dei pezzi sul nastro, che impiega 4 t.u.

Task  $A_2$ ) ogni 20 t.u. è eseguita la movimentazione di un pezzo con il robot  $PNP_1$ , impiegando 5 t.u.

C'è poi un task addizionale aperiodico di verifica dell'avvenuta movimentazione, caratterizzato da:

Task  $A_3$ ) un minimo tempo di attivazione di 30 t.u. un tempo di esecuzione massimo di 6 t.u.

Questo task aperiodico è gestito come un task periodico equivalente. I tre task sopra menzionati sono gestiti con una modalità di scheduling hard real time con algoritmo RMPO. Il problema sorge con la necessità di gestire un secondo robot pick-and-place ( $PNP_2$ ) così caratterizzato:

Task  $A_4$ ) ogni 20 t.u. è eseguita la movimentazione di un pezzo con il robot  $PNP_2$ , impiegando 3 t.u.

I quattro task suddetti devono essere gestiti con una modalità di scheduling hard real time. Si ipotizzi che tutti i task siano indipendenti. Si tenga conto che il costo di aggiornamento del sistema di controllo con algoritmo RMPO (necessario per gestire il nuovo task  $A_4$ ) è inferiore al costo di riprogrammazione del sistema di controllo esistente con algoritmo EDF, che a sua volta è inferiore al costo di acquisto e messa in esercizio di un nuovo e ulteriore sistema di controllo. Utilizzando RMPO, in caso di uguale priorità tra i task  $A_2$  e  $A_4$ , indicare a quale dei due robot ( $PNP_1$  o  $PNP_2$ ) si vuole dare priorità.

1. Verificare se sussiste la condizione necessaria di schedulabilità dei quattro task periodici.
2. Verificare se sussiste almeno una delle condizioni sufficienti di schedulabilità dei quattro task periodici, utilizzando l'algoritmo RMPO.
3. Eseguire lo scheduling RMPO dei quattro task periodici.
4. Eseguire lo scheduling EDF dei quattro task periodici.
5. Al fine di minimizzare i costi per la messa in esercizio di un sistema di controllo funzionante, quale soluzione conviene adottare?

## Esercizio 2

Si consideri la rete di Petri in Fig. 1.

- Calcolare i  $P$ -invarianti canonici della rete. La rete è conservativa? Giustificare la risposta.
- Calcolare i  $T$ -invarianti della rete.
- La rete è viva? E' reversibile? Può andare in blocco? Giustificare le risposte.

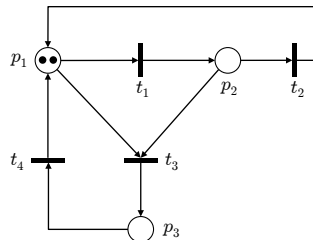


Figura 1: Una rete di Petri con marcatura iniziale  $\mathbf{x}_0 = (2 \ 0 \ 0)^T$ .

[altri esercizi sul retro]

### Esercizio 3

Una linea deve gestire la produzione di cinque prodotti  $\{A, B, C, D, E\}$  che necessitano di due lavorazioni in sequenza, la prima effettuata sulla macchina #1 e la seconda sulla macchina #2. La Tabella 1 riporta i tempi necessari per le lavorazioni dei diversi prodotti su ciascuna macchina. Sequenziare le lavorazioni sulle due macchine in modo da minimizzare il tempo totale di completamento dei cinque prodotti, descrivendo i singoli passi dell'algoritmo applicato. Riportare il risultato graficamente su un diagramma di Gantt. Mostrare anche un sequenziamento con il peggiore tempo di completamento possibile, nell'ipotesi che la seconda macchina non rimanga mai in attesa alla fine di una lavorazione quando è disponibile un prodotto già lavorato dalla prima.

Prodotto	A	B	C	D	E
Macchina #1	4	1	5	2	5
Macchina #2	3	2	4	3	6

Tabella 1: Tempi di lavorazione (minuti) dei cinque prodotti sulle due macchine.

### Esercizio 4

Si consideri la cella di lavorazione in Fig. 2. Nella cella entrano due tipi diversi di pezzi grezzi ( $A$  e  $B$ ) mediante due nastri trasportatori separati  $NA$  e  $NB$ . Tali nastri possono portare al massimo tre pezzi. Un robot manipolatore  $R$  carica i pezzi su due macchine utensili  $M1$  e  $M2$  che effettuano una lavorazione (ognuna lavora pezzi grezzi sia di tipo  $A$  sia di tipo  $B$ ). Tali macchine non sono dotate di buffer di ingresso e uscita, quindi possono accettare un solo pezzo alla volta. Quando entrambe le macchine hanno terminato la lavorazione, il manipolatore  $R$  le scarica e carica i pezzi nella zona  $Z$  (anch'essa priva di buffer) dove avviene l'assemblaggio. Al termine il robot  $R$  depone il pezzo finito e assemblato su un nastro trasportatore di uscita  $NU$  che può contenere un solo prodotto. Tale prodotto viene infine prelevato dall'esterno. Si considerino le operazioni di trasporto istantanee e quindi descritte da una sola transizione. Si rappresenti con una rete di Petri il comportamento del sistema indicando il significato fisico di posti e transizioni.

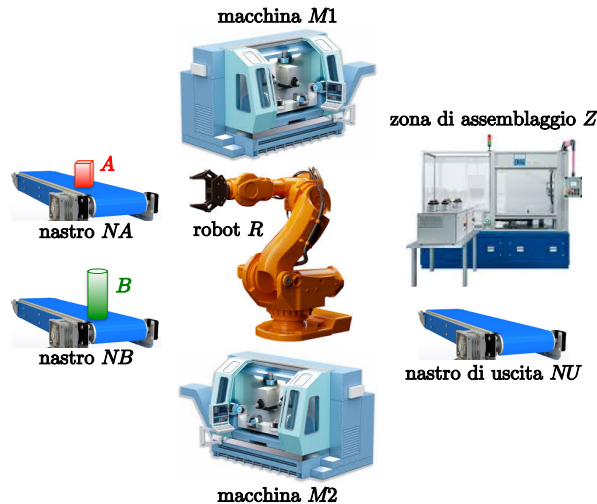


Figura 2: Una cella di lavorazione con un robot manipolatore, due macchine utensili, tre nastri di trasporto e una zona di assemblaggio.

[150 minuti; libri aperti]

# Soluzioni

12 Febbraio 2018

## Esercizio 1

Il problema di scheduling di task misti può essere trasformato in un problema equivalente caratterizzato dai seguenti quattro task periodici:

1.  $T_1 = 10$  t.u.,  $C_1 = 4$  t.u.;
2.  $T_2 = 20$  t.u.,  $C_2 = 5$  t.u.;
3.  $T_3 = 30$  t.u.,  $C_3 = 6$  t.u.;
4.  $T_4 = 20$  t.u.,  $C_4 = 3$  t.u.

**Punto 1.** Per verificare la condizione necessaria, si calcola il fattore di utilizzazione:

$$U = \frac{4}{10} + \frac{5}{20} + \frac{6}{30} + \frac{3}{20} = \frac{24 + 15 + 12 + 9}{60} = \frac{60}{60} = 1.$$

**Punto 2.** Verificata la condizione necessaria, controlliamo se esiste almeno una condizione sufficiente per lo scheduling secondo l'algoritmo RMPO:

$$U_{lsm}(\text{RMPO}, n) = n \left( 2^{1/n} - 1 \right) = 4 \left( 2^{1/4} - 1 \right) \simeq 0.76.$$

Dato che  $U > U_{lsm} > \ln 2$  e inoltre i quattro task non sono legati tra loro da relazioni armoniche, non è verificata alcuna condizione sufficiente. Non possiamo quindi dire a priori se RMPO sia in grado di schedulare i task.

**Punto 3.** La soluzione dello scheduling RMPO richiede una scelta sulla priorità da dare al task  $A_2$  e al task  $A_4$  durante la quinta t.u.. Nell'ipotesi che venga data priorità al robot  $PNP_1$ , la soluzione si blocca dopo la trentesima t.u. (Fig. 3). Anche nell'ipotesi che venga data priorità al robot  $PNP_2$ , la soluzione si blocca dopo la trentesima t.u. (Fig. 4). Nella trentesima t.u. viene data priorità a uno dei due robot e termina la deadline del task  $A_3$ . Da ciò si evince che RMPO non è in grado di eseguire lo scheduling dei quattro task periodici.

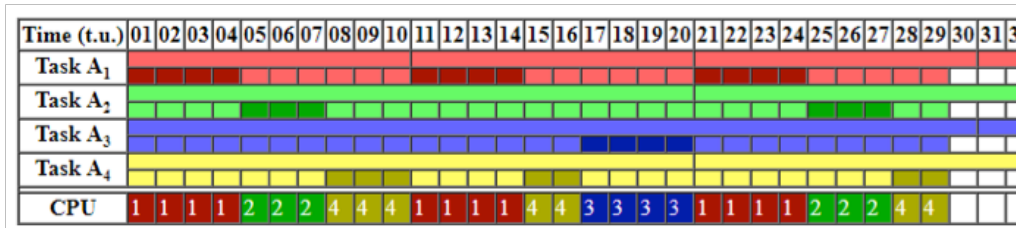


Figura 3: Scheduling RMPO con priorità al robot  $PNP_1$ .

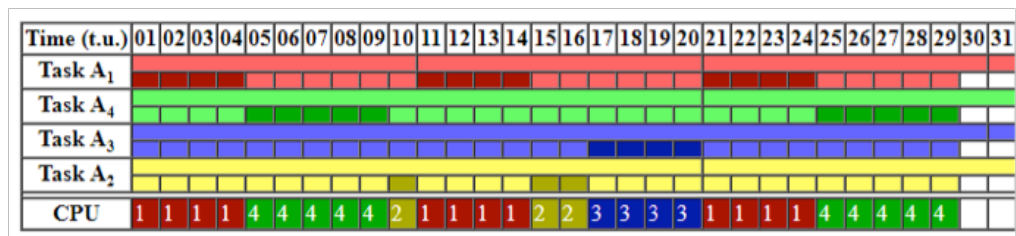


Figura 4: Scheduling RMPO con priorità al robot  $PNP_2$ .

**Punto 4.** La soluzione dello scheduling secondo l'algoritmo EDF è riportata in Fig. 5.

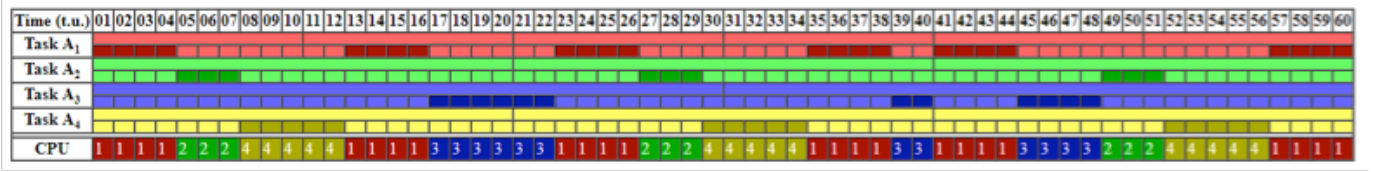


Figura 5: Scheduling con l'algoritmo EDF.

**Punto 5.** La soluzione più economica sarebbe stata quella di aggiornare l'algoritmo RMPO per gestire i quattro task. Ma come visto RMPO non è in grado di gestire i quattro task. Rimarrebbero pertanto solo due soluzioni tecniche: acquistare un sistema di controllo addizionale oppure aggiornare quello esistente con EDF. Poiché EDF fornisce una soluzione al problema, la scelta più conveniente è quella di aggiornare il sistema di controllo esistente con EDF.

### Esercizio 2

- La matrice di incidenza della rete di Petri in Fig. 1 è

$$\mathbf{C} = \begin{pmatrix} -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix},$$

il cui rango è  $\rho = 3$ , come si può facilmente verificare. Eliminando infatti la prima colonna, che è pari all'opposto della seconda, la matrice quadrata  $3 \times 3$  residua ha determinante  $+1$ . Lo spazio nullo di  $\mathbf{C}^T$  ha quindi dimensione  $|\mathcal{P}| - \rho = 3 - 3 = 0$ , ossia non esiste alcun  $P$ -invariante (l'equazione  $\boldsymbol{\gamma}^T \mathbf{C} = \mathbf{0}^T$  ha solo la soluzione banale  $\boldsymbol{\gamma} = \mathbf{0}$ ). La rete pertanto non è conservativa, poiché non è ricoperta da  $P$ -invarianti positivi.

- Lo spazio nullo di  $\mathbf{C}$  è invece mono-dimensionale ( $|\mathcal{T}| - \rho = 4 - 3 = 1$ ). Il calcolo dei  $T$ -invarianti fornisce

$$\mathbf{C} \boldsymbol{\eta} = \mathbf{0} \quad \Rightarrow \quad \begin{cases} -\eta_1 + \eta_2 - \eta_3 + \eta_4 = 0 \\ \eta_1 - \eta_2 - \eta_3 = 0 \\ \eta_3 - \eta_4 = 0 \end{cases} \quad \Rightarrow \quad \begin{cases} \eta_1 = \eta_2 \\ \eta_3 = \eta_4 = 0. \end{cases}$$

Pertanto, come previsto, esiste il solo  $T$ -invariante canonico:

$$\boldsymbol{\eta} = (1 \quad 1 \quad 0 \quad 0)^T.$$

Come vettore delle occorrenze,  $\boldsymbol{\eta}$  è associato all'unica sequenza ammissibile di scatti  $\{t_1, t_2\}$ , sequenza che riporta la rete alla marcatura iniziale.

- La rete non è viva, come si vede bene dall'albero di raggiungibilità riportato in Fig. 6 (si ricordi sempre che l'albero deve essere costruito con una logica 'depth first' e con ordinamento lessicografico). Infatti, dopo la sequenza ammissibile di scatti  $\{t_1, t_3, t_4\}$ , le transizioni  $t_3$  e  $t_4$  non sono più vive (abilitabili). La rete inoltre non è reversibile, perché non può tornare alla marcatura iniziale a partire da tutte le possibili marcature raggiungibili. Non appena scatta la transizione  $t_3$ , nella rete rimarrà solo un token. Infine la rete non va mai in blocco in quanto è sempre e comunque ammissibile il ciclo  $p_1 \rightarrow p_2 \rightarrow p_1$  (con la sequenza di scatti  $\{t_1 t_2\}$ ).

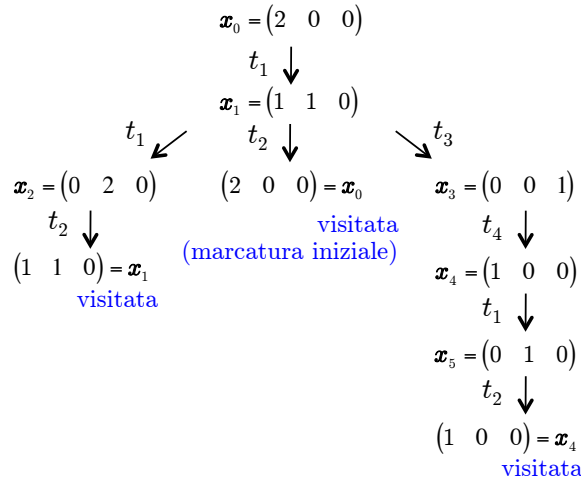


Figura 6: Albero di raggiungibilità della rete di Petri di Fig. 1.

### Esercizio 3

Si tratta di un problema di flow shop scheduling con  $n = 5$  lavori su  $m = 2$  macchine, per il quale è possibile applicare l'algoritmo di Johnson che minimizza il tempo totale di completamento (*makespan*).

L'algoritmo processa per primo il prodotto  $i$  ( $i = 1, \dots, 5$ ) con il minimo tempo di lavorazione  $t_{ij}$ . In questo caso,  $i = B$  (il minimo tempo nella tabella è  $t_{B1} = 1$ . Poiché  $t_{B1} = 1 < 2 = t_{B2}$ , il prodotto  $B$  verrà schedulato per primo. Rimosso  $B$  dalla lista dei prodotti, il minimo tempo di lavorazione sui restanti è  $t_{D1} = 2$ . Poiché  $t_{D1} = 2 < 3 = t_{D2}$ , il prodotto  $D$  verrà posto come secondo, ossia immediatamente dopo il primo. Rimosso anche  $D$  dalla lista dei prodotti, il minimo tempo di lavorazione sui restanti è  $t_{A2} = 3$ . Poiché  $t_{A1} = 4 > 3 = t_{A2}$ , il prodotto  $A$  verrà invece schedulato per ultimo. Tra i due rimanenti lavori, quello con il minore tra i tempi di lavorazione è  $C$ , con  $t_{C2} = 4$ . Poiché  $t_{C1} = 5 > 4 = t_{C2}$ , il prodotto  $C$  verrà invece sequenziato come penultimo. Di conseguenza  $E$  è in terza posizione. La soluzione ottima fornita dall'algoritmo è quindi  $\{B, D, E, C, A\}$ , illustrata nel diagramma di Gantt in Fig. 7. Il tempo totale è pari a  $T = 21$  minuti. Si noti che questo sequenziamento ottimo è unico (non è stato infatti necessario operare scelte tra tempi di lavorazione uguali).

La somma dei tempi delle lavorazioni sulla macchina #1 è pari a 17 minuti. La prima macchina rimane quindi a riposo per gli ultimi quattro minuti. La somma dei tempi delle lavorazioni sulla macchina #2 è pari a 18 minuti. La seconda macchina è ferma quindi per un totale di tre minuti, un minuto all'inizio della sequenza e due minuti in attesa del termine della lavorazione del prodotto  $E$  da parte della prima macchina. I prodotti  $C$  e  $A$  attendono ciascuno un minuto prima di poter iniziare la loro lavorazione sulla macchina #2.

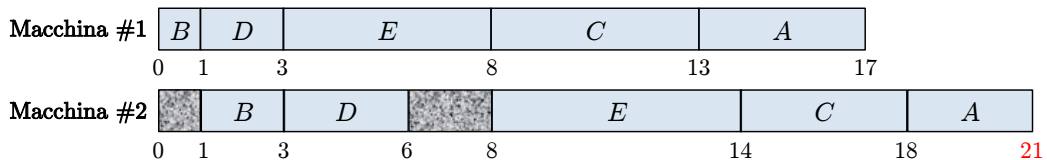


Figura 7: Diagramma di Gantt con la soluzione del problema di flow shop scheduling.

La Fig. 8 mostra invece una sequenza con il peggior tempo totale di completamento possibile sotto le ipotesi di lavoro fatte, pari a  $T = 25$  minuti (quasi il 20% peggiore di quella ottima). La sequenza  $\{A, C, E, D, B\}$  non è altro che quella con ordine inverso rispetto alla sequenza ottima. Il tempo in cui la macchina #2 è ferma è di 7 minuti, quattro iniziali e poi due minuti (per attendere il prodotto  $C$ ) e un minuto (per attendere il prodotto  $E$ ).

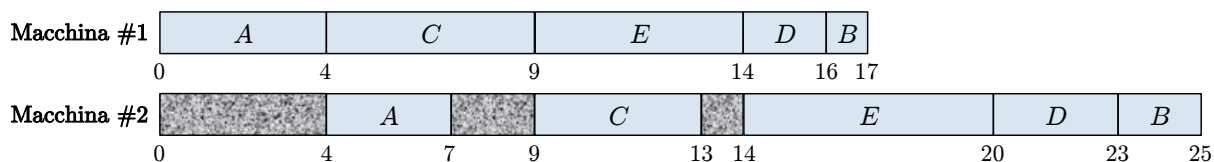


Figura 8: Diagramma di Gantt di uno scheduling con il peggior tempo di completamento possibile per il problema considerato ( $T = 25$  minuti).

#### Esercizio 4

La rete di Petri con 14 posti e 9 transizioni in Fig. 9 descrive il comportamento stati/eventi della cella di lavorazione considerata.

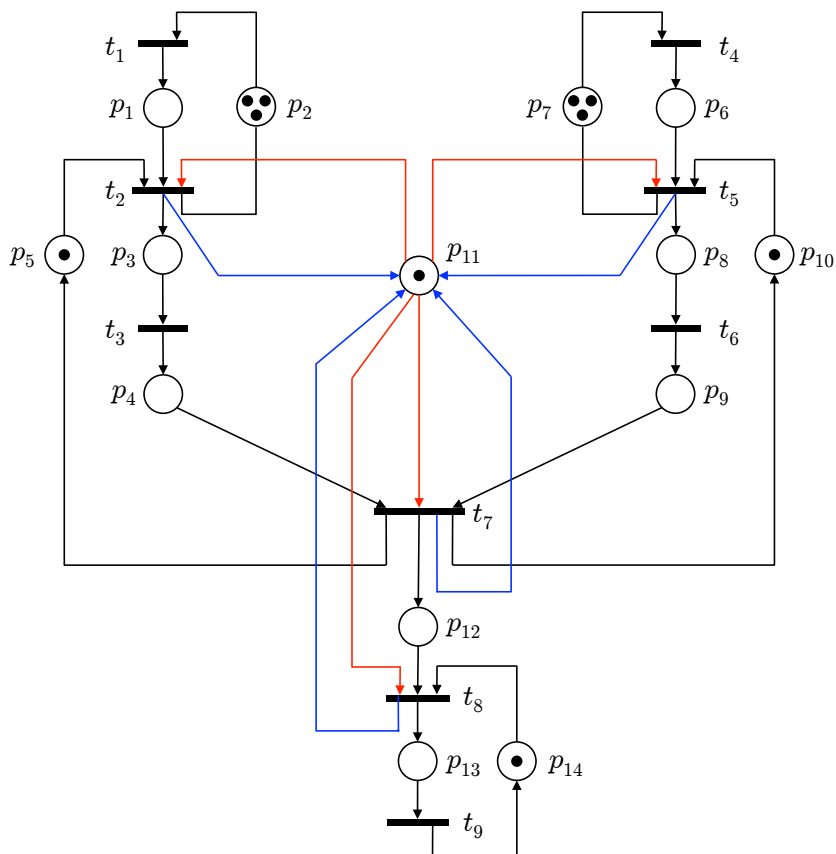


Figura 9: Modellistica con rete di Petri del comportamento della cella di lavorazione di Fig. 2.

### Significato dei posti:

- $p_1 - p_6$ : pezzo  $A - B$  disponibile sul nastro  $NA - NB$
- $p_2 - p_7$ : disponibilità posti sul nastro  $A - B$  (all'inizio, con tre token ciascuno)
- $p_3 - p_8$ : pezzo pronto per la macchina  $M1 - M2$
- $p_4 - p_9$ : pezzi  $A - B$  pronti per assemblaggio
- $p_5 - p_{10}$ : disponibilità della macchina  $M1 - M2$  (all'inizio, con un token ciascuno)
- $p_{11}$ : disponibilità robot  $R$  (all'inizio un token)
- $p_{12}$ : assemblaggio
- $p_{13}$ : pezzo disponibile sul nastro  $NU$  di uscita
- $p_{14}$ : posto disponibile sul nastro di uscita (con un token all'inizio)

### Significato delle transizioni:

- $t_1 - t_4$ : arrivo di un pezzo  $A - B$  sul nastro di ingresso  $NA - NB$
- $t_2 - t_5$ : prelievo dal nastro  $NA - NB$  e deposito in  $M1 - M2$  (i pezzi  $A$  e  $B$  possono essere lavorati indifferentemente da entrambe le macchine!)
- $t_3 - t_6$ : inizio lavorazione su  $M1 - M2$
- $t_7$ : prelievo dei pezzi (pronti)  $A$  e  $B$  dalle due macchine e deposito nella zona di assemblaggio
- $t_8$ : prelievo del pezzo dalla zona assemblaggio e deposito sul nastro di uscita  $NU$
- $t_9$ : prelievo del pezzo pronto dall'esterno

Si noti nella rete la struttura parallela relativa alla gestione dei due nastri  $NA$  e  $NB$  di arrivo dei pezzi alla cella, e il successivo coordinamento per l'assemblaggio e l'uscita del pezzo finito dalla cella tramite il nastro  $NU$ . Il robot  $R$  è coinvolto in tutte le operazioni di trasporto pezzi (nastro ingresso  $\rightarrow$  macchina, macchina  $\rightarrow$  zona assemblaggio, zona assemblaggio  $\rightarrow$  nastro uscita). Il posto  $p_{11}$  di disponibilità del robot ha infatti quattro archi in uscita (in rosso) e quattro di ritorno (in blu) ad operazione terminata. Poiché queste operazioni di trasporto si sono supposte istantanee (durate trascurabili rispetto alle altre operazioni nella cella), non sono presenti posti del tipo 'robot sposta pezzo ...' ma solo transizioni associate a questi spostamenti.

\* \* \* \* \*