



# Corso di Automazione industriale

Esercizi in preparazione all'esame



# Programmazione PLC

# Esercizio 1

Sviluppare un software in Ladder, SFC e testo strutturato, che consenta di controllare un forno industriale con il seguente funzionamento.

Quando un operatore preme il pulsante di start ( $START=1$ ), il PLC fa partire il ciclo di riscaldamento del materiale ( $OP=1$ ) che dura complessivamente 30s. Dopo tale tempo il comando  $OP$  deve essere resettato dal PLC ( $OP=0$ ), segnalando l'avvenuto completamento del processo tramite l'accensione (per 10s) della luce verde di segnalazione di operazione terminata ( $VERDE=1$ ). Al termine dei 10s, la luce deve essere spenta ( $VERDE=0$ ).

# Esercizio 1

L'unico ingresso del sistema è:

- START: Pulsante di avvio del forno

Le uscite sono, invece:

- OP: Abilitazione del forno
- VERDE: Luce di indicazione della terminazione del lavoro di riscaldamento

# Esercizio 1

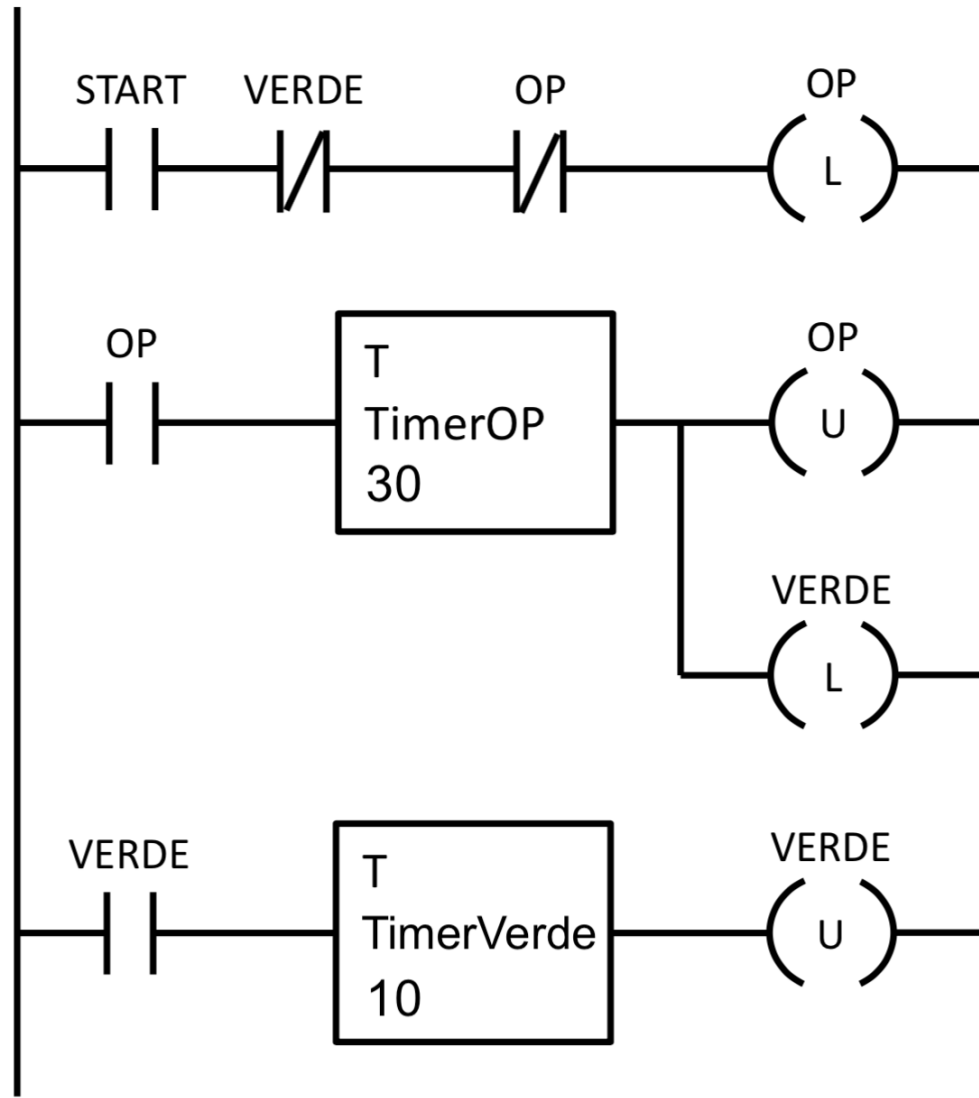
*Nota:* Per lo sviluppo in testo strutturato si supponga di avere a disposizione un function block chiamato **TON** con due ingressi

- IN: ingresso che attiva il timer (se è 1 il timer lavora, se è 0 si resetta)
- PT: valore temporale da raggiungere (formato TIME)

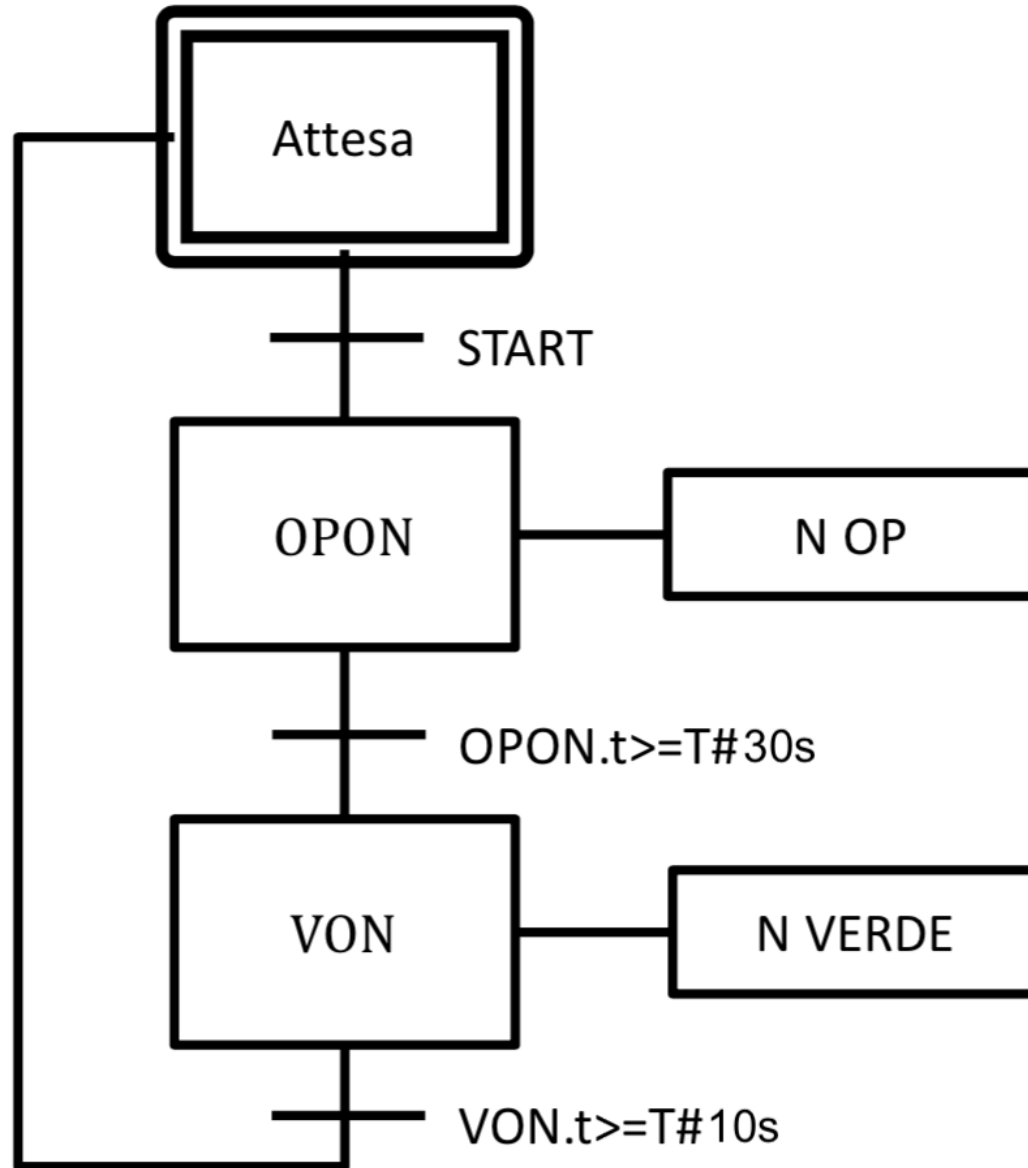
E due uscite:

- ET: tempo trascorso
- Q: flag che va ad 1 se  $ET \geq PT$

# Esercizio 1 – Soluzione LADDER



# Esercizio 1 – Soluzione SFC



# Esercizio 1 – Soluzione Testo Strutturato

```
PROGRAM _INIT
    TOP(IN := 0, PT := T#30s);
    TV(IN := 0, PT := T#10s);
END_PROGRAM
```

```
PROGRAM _CYCLIC
    TOP(IN:=OP);
    TV(IN:=VERDE);
    IF START AND NOT(OP) AND NOT(VERDE) THEN
        OP := 1;
    ELSIF OP THEN
        IF TOP.Q THEN
            OP := 0;
            VERDE := 1;
        END_IF;
    ELSIF VERDE THEN
        IF TV.Q THEN
            VERDE := 0;
        END_IF;
    END_IF;
END_PROGRAM
```

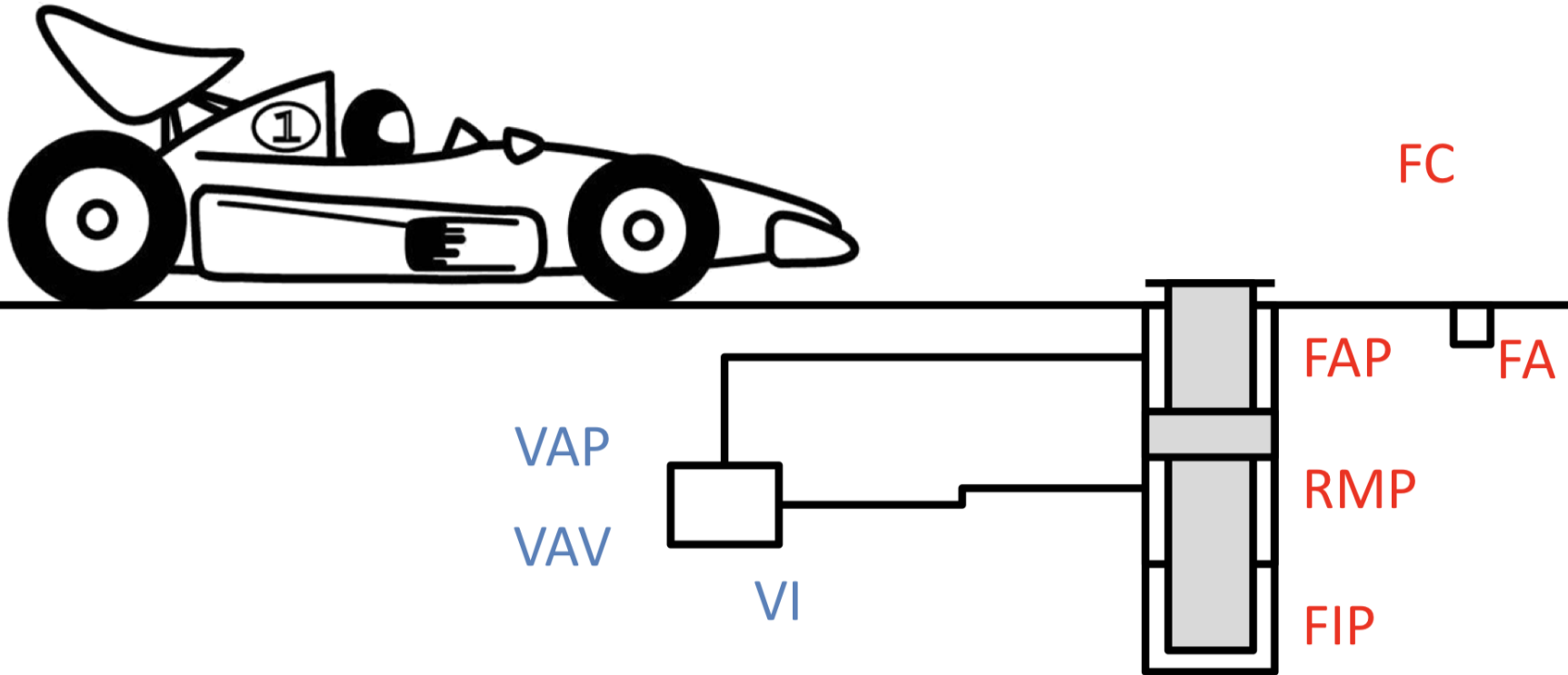
# Esercizio 2

Sviluppare un software in Ladder, SFC e testo strutturato, che consenta di controllare un sollevatore idraulico di auto da corsa per il cambio gomme.

Il funzionamento del sistema è il seguente:

- L'auto arriva nella piazzola, facendosi rilevare dalla fotocellula
- Il pistone inizia il sollevamento in modalità avanti piano (supponendo che esso parta dalla condizione FIP) fino al raggiungimento di RMP
- Il pistone procede fino a FAP in modalità avanti veloce
- Il pistone rimane in posizione finché il non termina il cambio gomme (si suppone, per semplicità, che FC vada a 0 istantaneamente quando il pistone raggiunge FAP)
- Il pistone torna a FIP con il comando VI

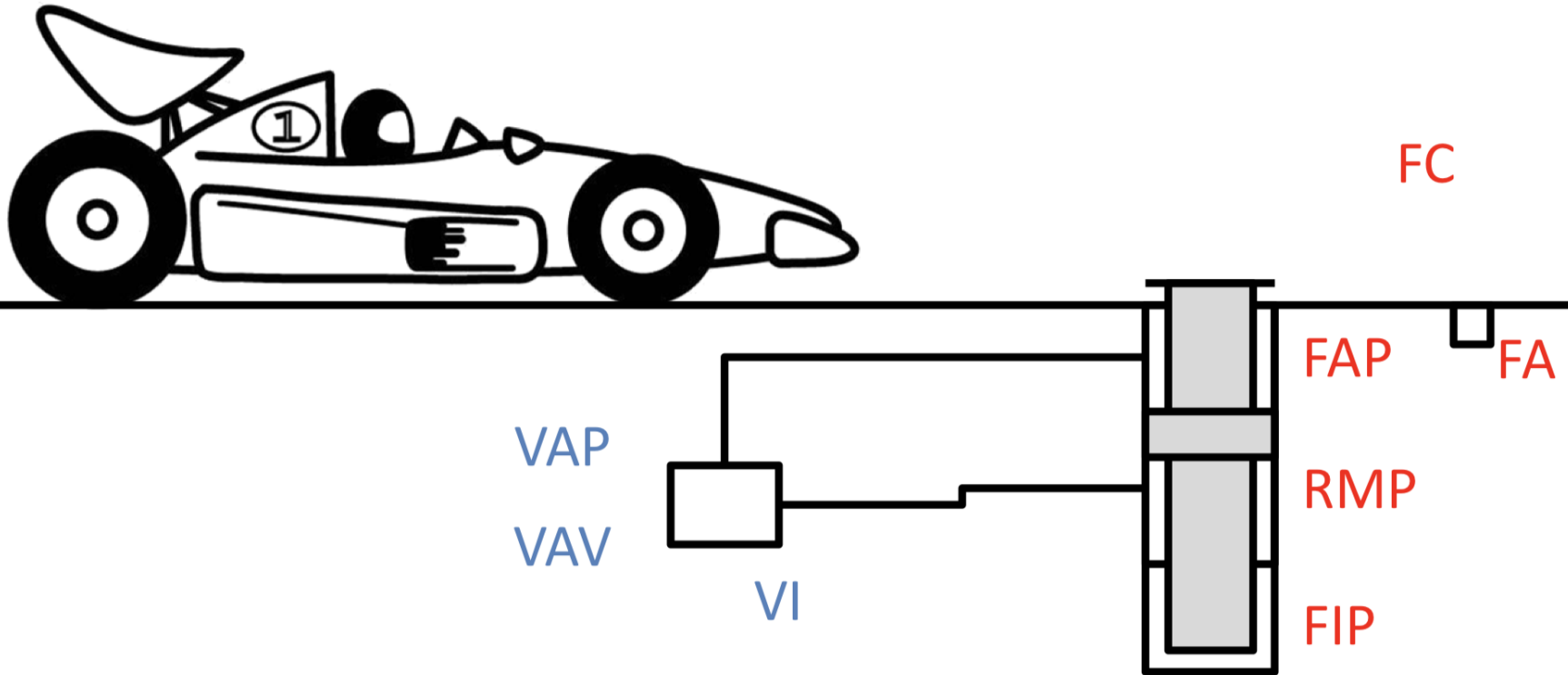
# Esercizio 2



## INPUT:

- VAP: Valvola avanti piano (0 – disattivato – 1 attivato)
- VAV: Valvola avanti veloce (0 – disattivato – 1 attivato)
- VI: Valvola indietro (0 – disattivato – 1 attivato)

# Esercizio 2



## OUTPUT:

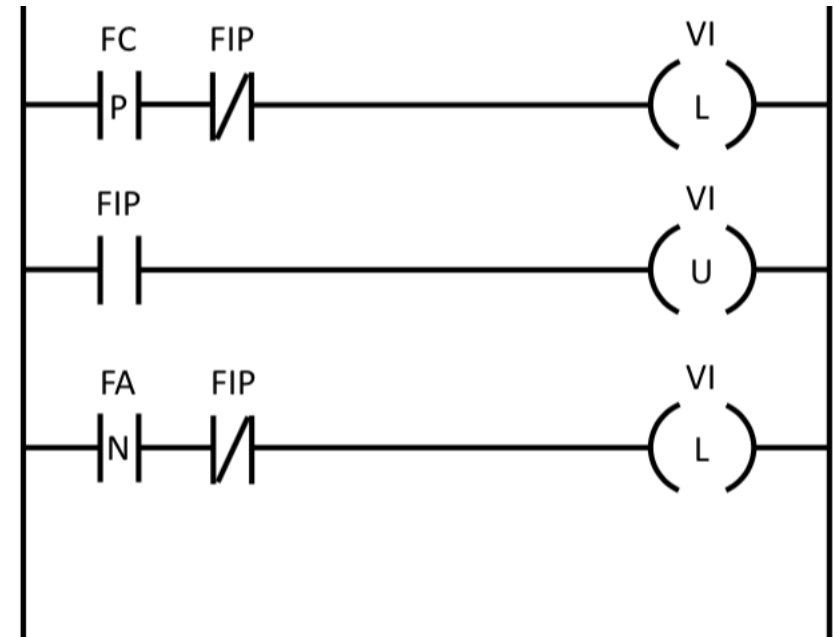
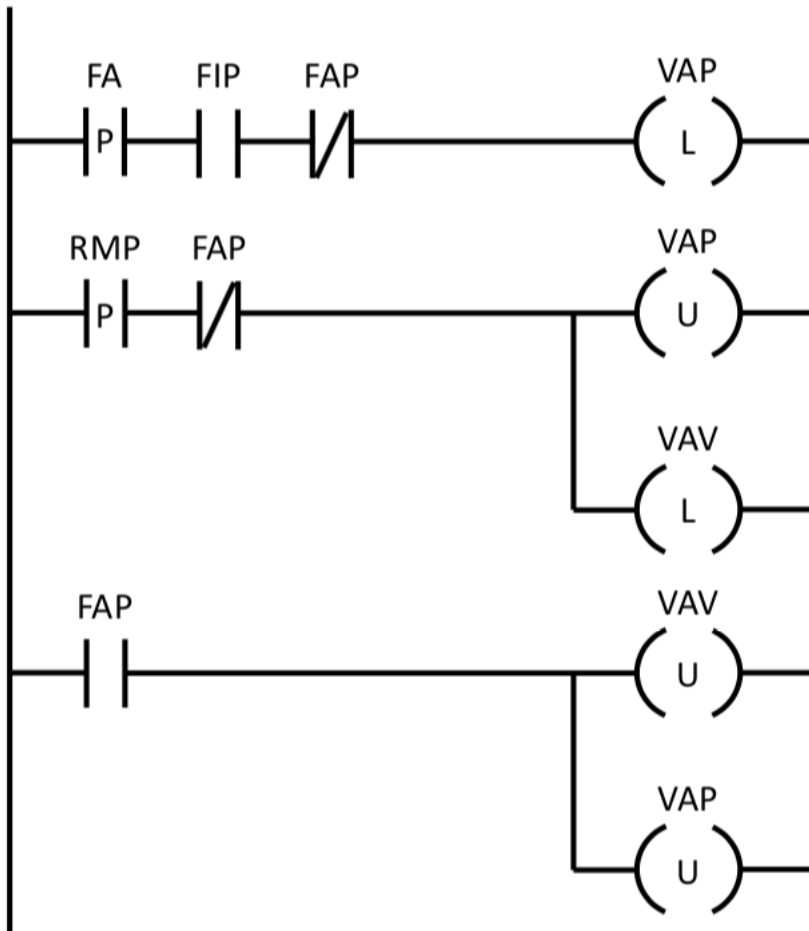
- FC: Fine cambio gomme (0 cambio in corso – 1 cambio non in corso)
- FAP: Finecorsa avanti pistone (0 non attivato – 1 attivato)
- FA: Fotocellula auto (0 auto non presente – 1 auto presente)
- RMP: Raggiungimento metà corsa pistone (0 pistone esteso per meno di metà corsa – 1 pistone esteso per più di metà corsa)
- FIP: Finecorsa indietro pistone (0 non attivato – 1 attivato)

# Esercizio 2

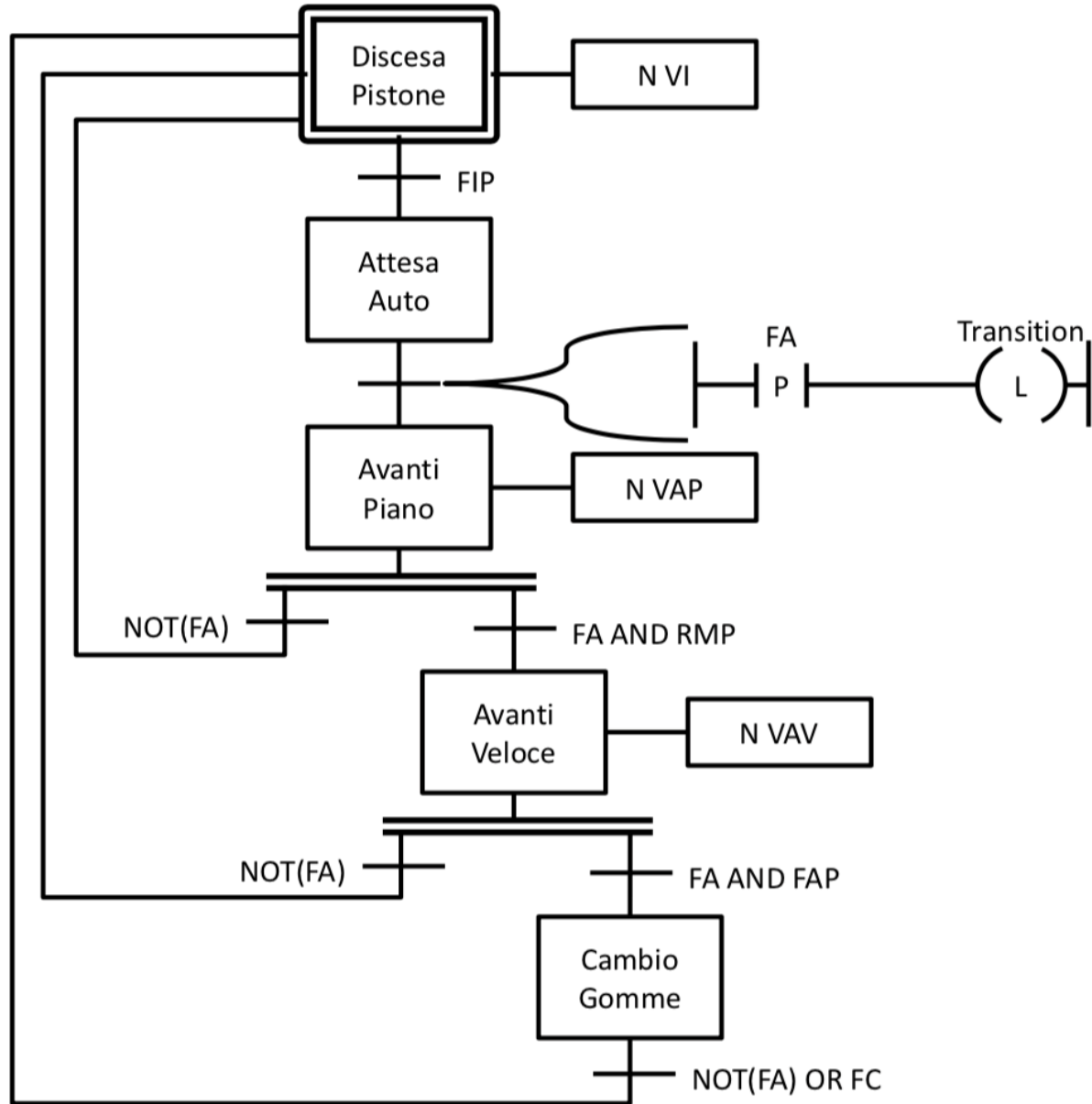
*Note:*

- FA diventerà 0 solo dopo lo spostamento completo dell'auto, che può avvenire diversi cicli dopo l'abbassamento del pistone fino a FIP
- Impostare il programma in maniera tale da imporre al pistone il rientro in caso di  $FA = 0$  (in qualunque condizione esso si ritrovi).

# Esercizio 2 – Soluzione LADDER



# Esercizio 2 – Soluzione SFC



# Esercizio 2 – Soluzione Testo Strutturato

```
PROGRAM _INIT
```

```
    FIP := 1;  
    State := 0;
```

```
END_PROGRAM
```

```
PROGRAM _CYCLIC
```

```
    CASE State OF
```

```
        0:
```

```
            IF FIP THEN
```

```
                VI := 0;  
                State := 1;
```

```
            ELSE
```

```
                VI := 1;
```

```
            END_IF;
```

```
        1:
```

```
            IF EDGEPOS(FA) THEN
```

```
                State := 2;
```

```
            END_IF;
```

```
        2:
```

```
            IF NOT FA THEN
```

```
                State := 0;  
                VAP := 0;
```

```
            ELSE
```

```
                IF RMP THEN
```

```
                    State := 3;  
                    VAP := 0;
```

```
                ELSE
```

```
                    IF RMP THEN
```

```
                        State := 3;  
                        VAP := 0;
```

```
                    ELSE
```

```
                        VAP := 1;
```

```
                    END_IF;
```

```
                END_IF;
```

```
        3:
```

```
            IF NOT FA THEN
```

```
                State := 0;  
                VAV := 0;
```

```
            ELSE
```

```
                IF FAP THEN
```

```
                    State := 4;  
                    VAV := 0;
```

```
                ELSE
```

```
                    VAV := 1;
```

```
                END_IF;
```

```
            END_IF;
```

```
        4:
```

```
            IF (NOT FA) OR FC THEN
```

```
                State := 0;
```

```
            END_IF;
```

```
        END_CASE;
```

```
    END_PROGRAM
```

# Esercizio 3

Sviluppare un software in Ladder, SFC e testo strutturato, che consenta di controllare un trapano industriale, il cui funzionamento è descritto di seguito.

Quando un operatore preme il tasto di START (START=1) allora il PLC deve avviare il motore del trapano (MOTORE=1) solamente se è stata abbassata anche la protezione (PROTEZIONE=1).

In caso contrario, il PLC deve allertare l'operatore con un allarme (ALLARME=1).

Quando l'operatore abbassa la protezione, allora il PLC deve spegnere l'allarme (ALLARME=0) e l'operatore può ripremere START (START=1) per far partire il motore (MOTORE=1).

Per fermare la mola l'operatore può premere un tasto di STOP (STOP=1). In caso di mancata pressione del pulsante il trapano si arresta automaticamente dopo 10 secondi.

# Esercizio 3

Gli ingressi del sistema sono:

- START: Pulsante di avvio del trapano (0 pulsante non premuto – 1 pulsante premuto)
- STOP: Pulsante di spegnimento del trapano (0 pulsante non premuto – 1 pulsante premuto)
- PROTEZIONE: Sensore sulla barriera protettiva (0 non attivata – 1 attivata)

Le uscite sono

- MOTORE: Abilitazione del motore del trapano (0 – disattivato – 1 attivato)
- ALLARME: Allarme che segnala il mancato utilizzo della protezione (0 – disattivato – 1 attivato)

# Esercizio 3

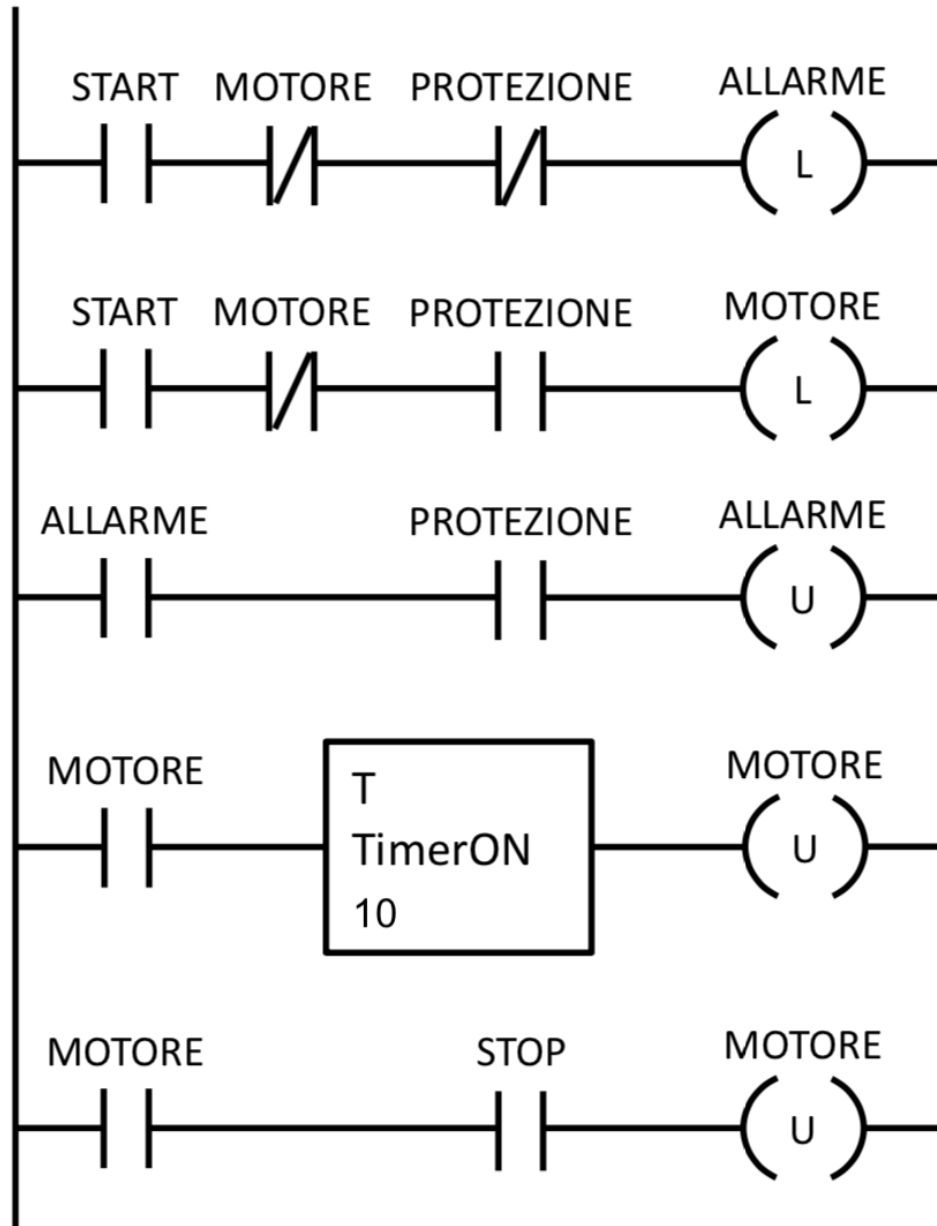
*Nota:* Per lo sviluppo in testo strutturato si supponga di avere a disposizione un function block chiamato **TON** con due ingressi

- IN: ingresso che attiva il timer (se è 1 il timer lavora, se è 0 si resetta)
- PT: valore temporale da raggiungere (formato TIME)

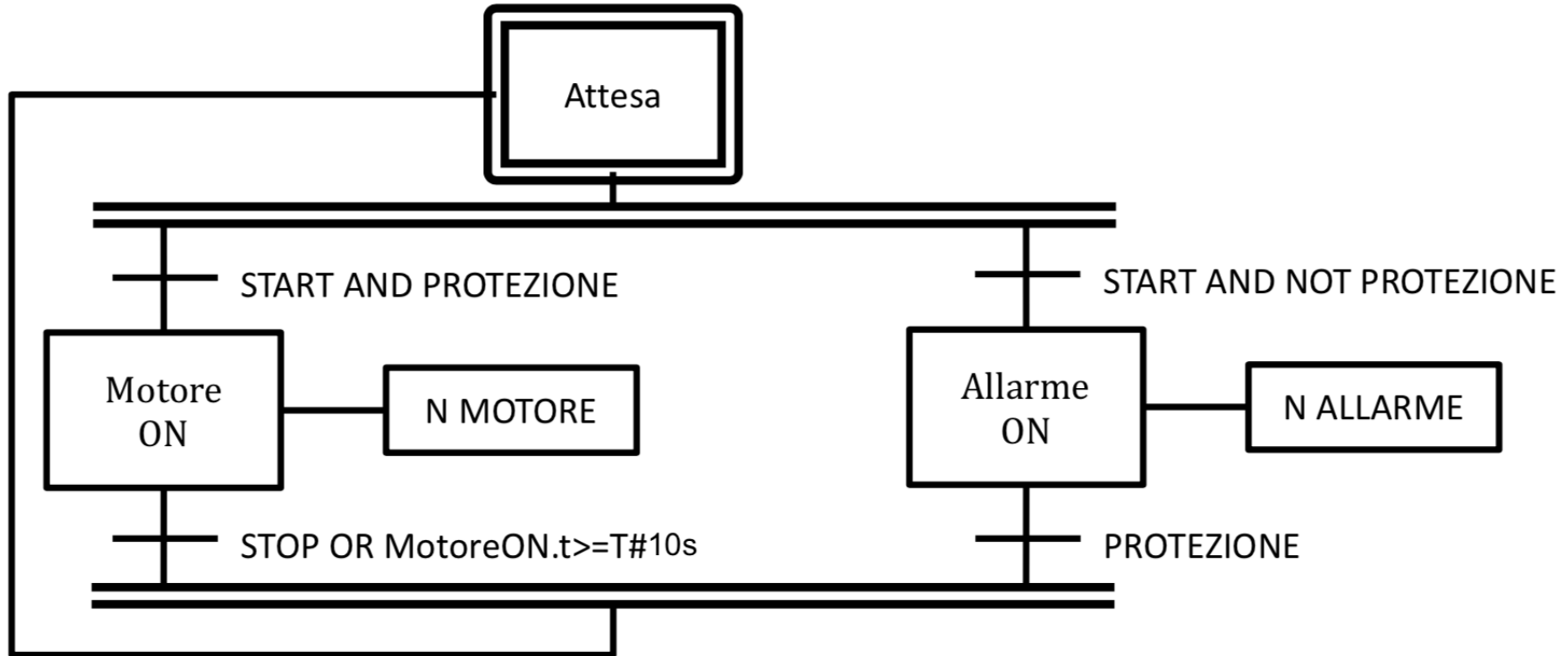
E due uscite:

- ET: tempo trascorso
- Q: flag che va ad 1 se  $ET \geq PT$

# Esercizio 3 – Soluzione LADDER



# Esercizio 3 – Soluzione SFC



# Esercizio 3 – Soluzione Testo Strutturato

```
PROGRAM _INIT
```

```
    MOTORE := 0;
```

```
    ALLARME := 0;
```

```
    Timer(IN := 0, PT := T#10s);
```

```
END_PROGRAM
```

```
PROGRAM _CYCLIC
```

```
    IF NOT MOTORE AND START THEN
```

```
        IF PROTEZIONE THEN
```

```
            MOTORE := 1;
```

```
        ELSE
```

```
            ALLARME := 1;
```

```
        END_IF;
```

```
    END_IF;
```

```
    IF ALLARME AND PROTEZIONE THEN
```

```
        ALLARME := 0;
```

```
    END_IF;
```

```
    Timer(IN := MOTORE);
```

```
    IF Timer.Q OR STOP THEN
```

```
        MOTORE := 0;
```

```
    END_IF;
```

```
END_PROGRAM
```



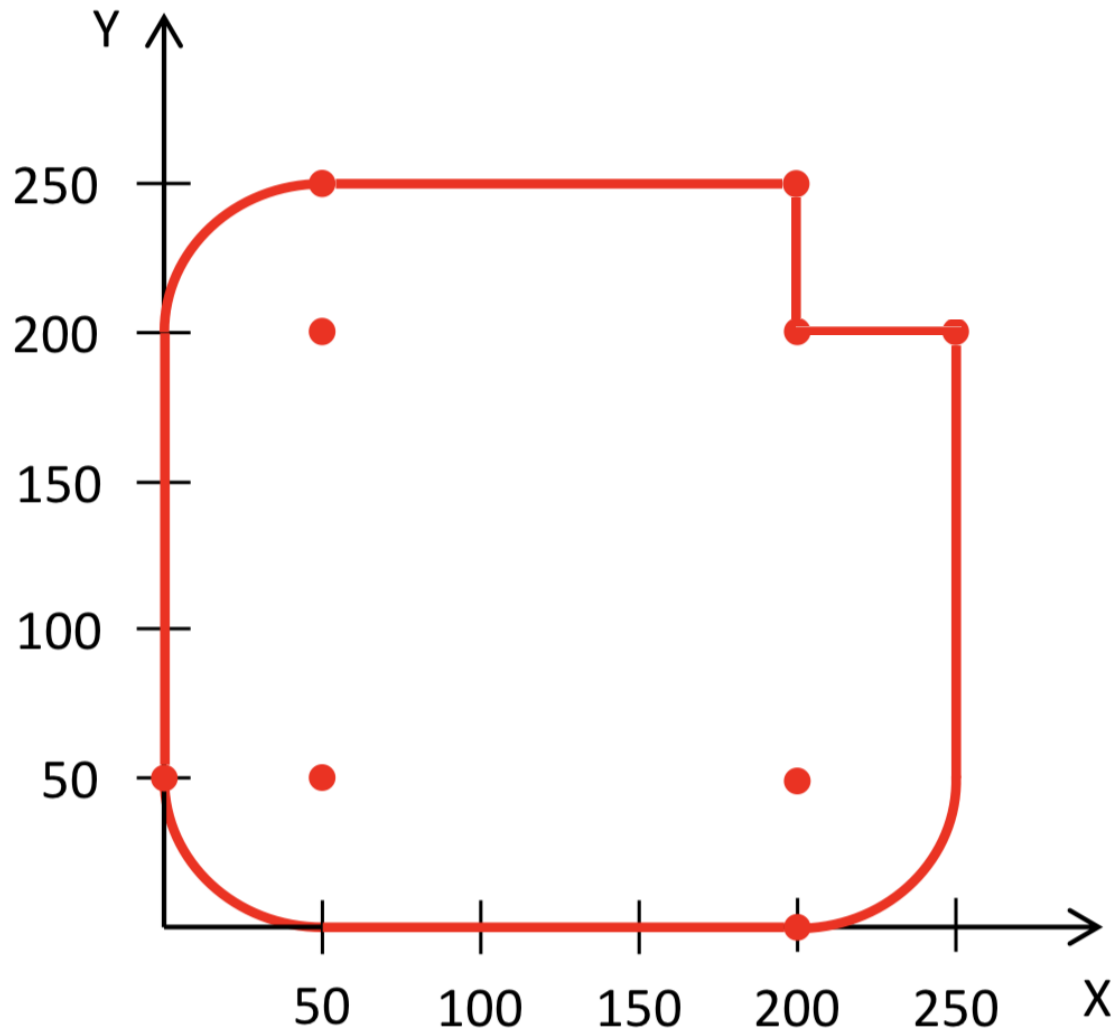
## G-Code

# Esercizio 1

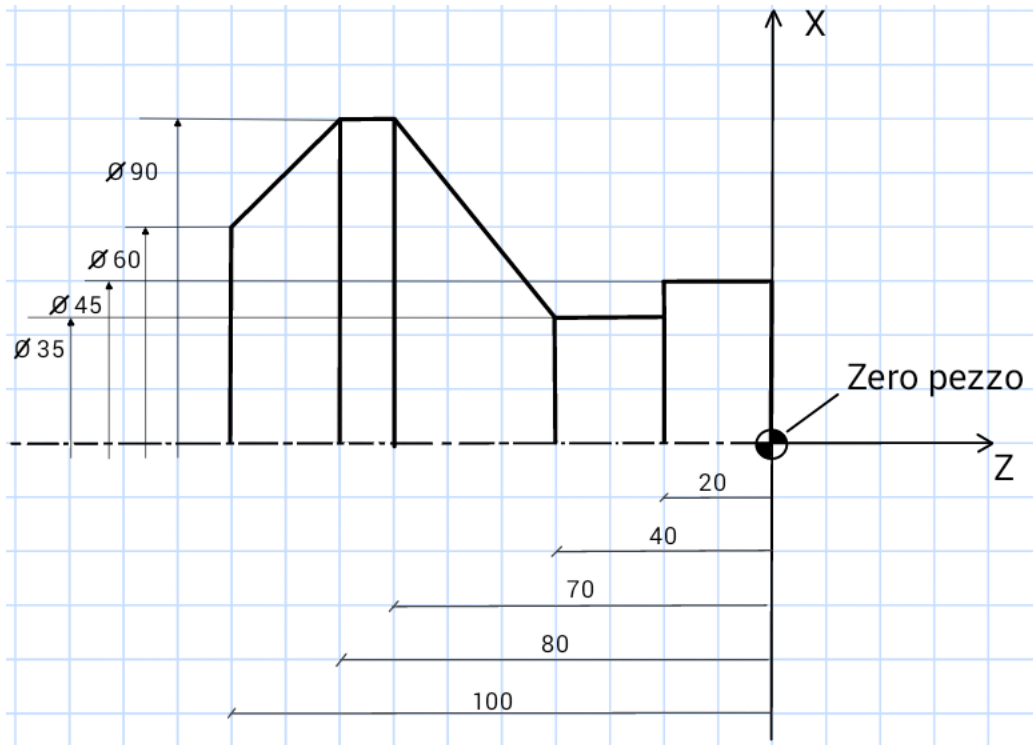
Descrivere la forma della lavorazione effettuata con il seguente codice:

N1	G00	X0	Y50		
N2	G01	X0	Y200		
N3	G02	X50	Y250	I50	J200
N4	G01	X200	Y250		
N5	G01	X200	Y200		
N6	G01	X250	Y250		
N7	G01	X250	Y50		
N8	G02	X200	Y0	I200	J50
N9	G01	X50	Y0		
N10	G02	X0	Y50	I50	J50

# Esercizio 1 - Soluzione



# Esercizio 2



Si realizzi un software per CN che consenta di eseguire la tornitura rappresentata, con:

- Rotazione antioraria del mandrino, con lubrificante
- Velocità di taglio di 260 mm/min
- Utilizzo lubrificante
- Tool 01

# Esercizio 2 - Soluzione

N10	T0101	M06								
	G90	G96	S260	M04	M08					
N20	G00	X45	Z0							
N30	G01	Z-20								
N40	G01	X45								
N50	G01	Z-40								
N60	G01	X90	Z-70							
N70	G01	Z-80								
N80	G01	X60	Z-100							
N90	G00	X100	Z-200							
N100	M05	M09	M30							

Selezione tool 01

Impostazione quote assolute, setup mandrino e avvio lubrificante

Movimento lineare veloce

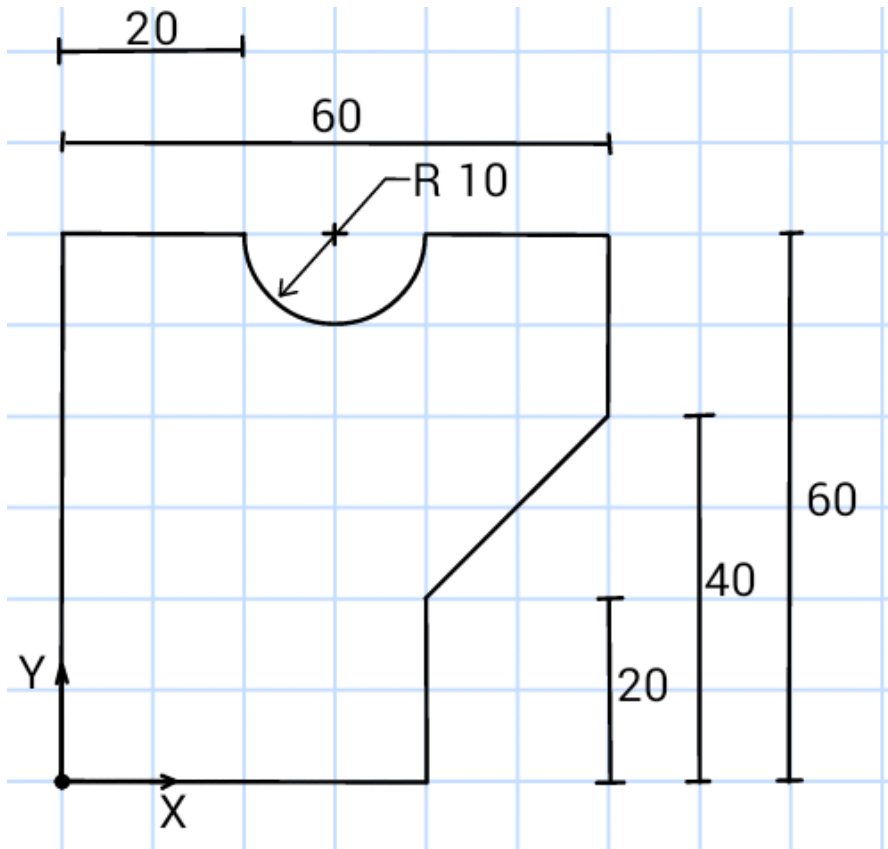
Interpolazione lineare

Spegnimento mandrino, lubrificante e fine programma

# Esercizio 3

Si realizzi un software per CN che consenta di realizzare il pezzo rappresentato a fianco, con:

- Mandrino in senso orario, con lubrificante (utensile 03)
- Utilizzo compensazione raggio utensile
- $F=400$  mm/min.
- $S=500$  giri/min.



# Esercizio 3 - Soluzione

N10	G90	T03	M06						
	G42	F400	S500	M03	M08				
N20	G01	X0	Y60						
N30	G01	X20	Y60						
N40	G03	X40	Y60	I30	J60				
N50	G01	X60	Y60						
N60	G01	X60	Y40						
N70	G01	X40	Y20						
N80	G01	X40	Y0						
N90	G01	X0	Y0						
N100	G00	X-10	Y-10	M05	M09				
	M30								

Abilitazione quote assolute e selezione tool 03

Settaggi mandrino ed avvio lubrificante

Interpolazione lineare

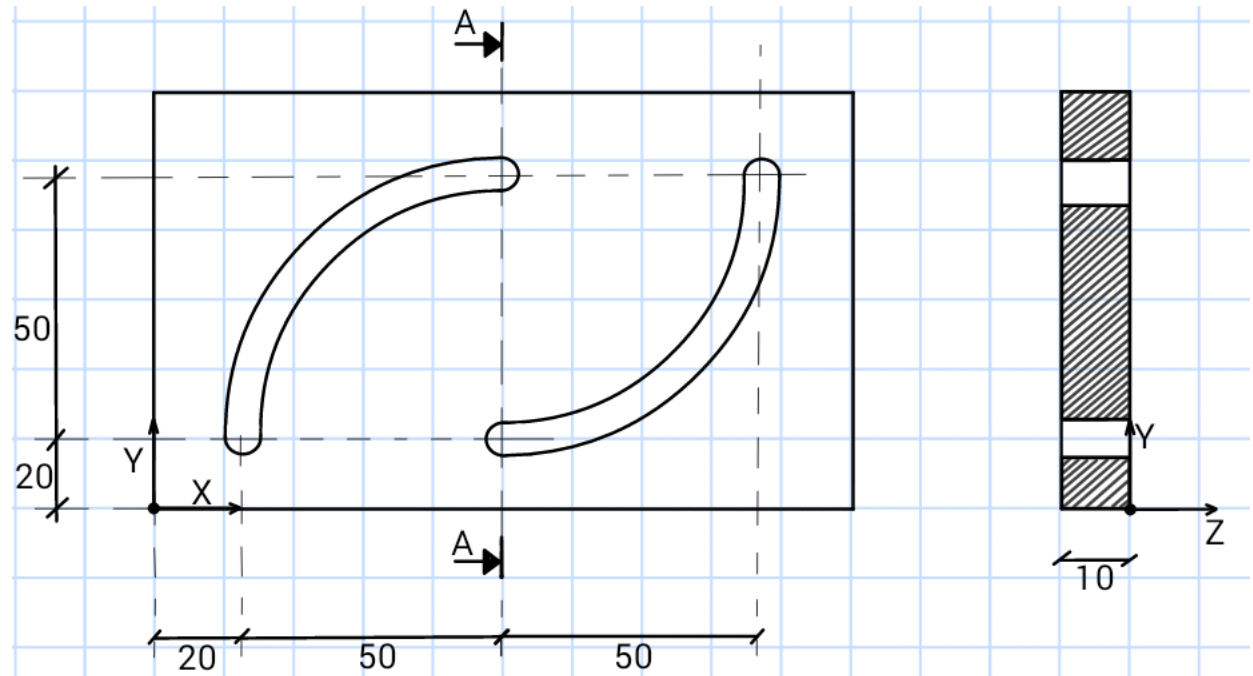
Interpolazione circolare anti-oraria

Movimento veloce fino a (-10,-10), spegnimento mandrino, lubrificante e fine programma

# Esercizio 4

Si realizzi un software per CN che consenta di eseguire la lavorazione rappresentata, con:

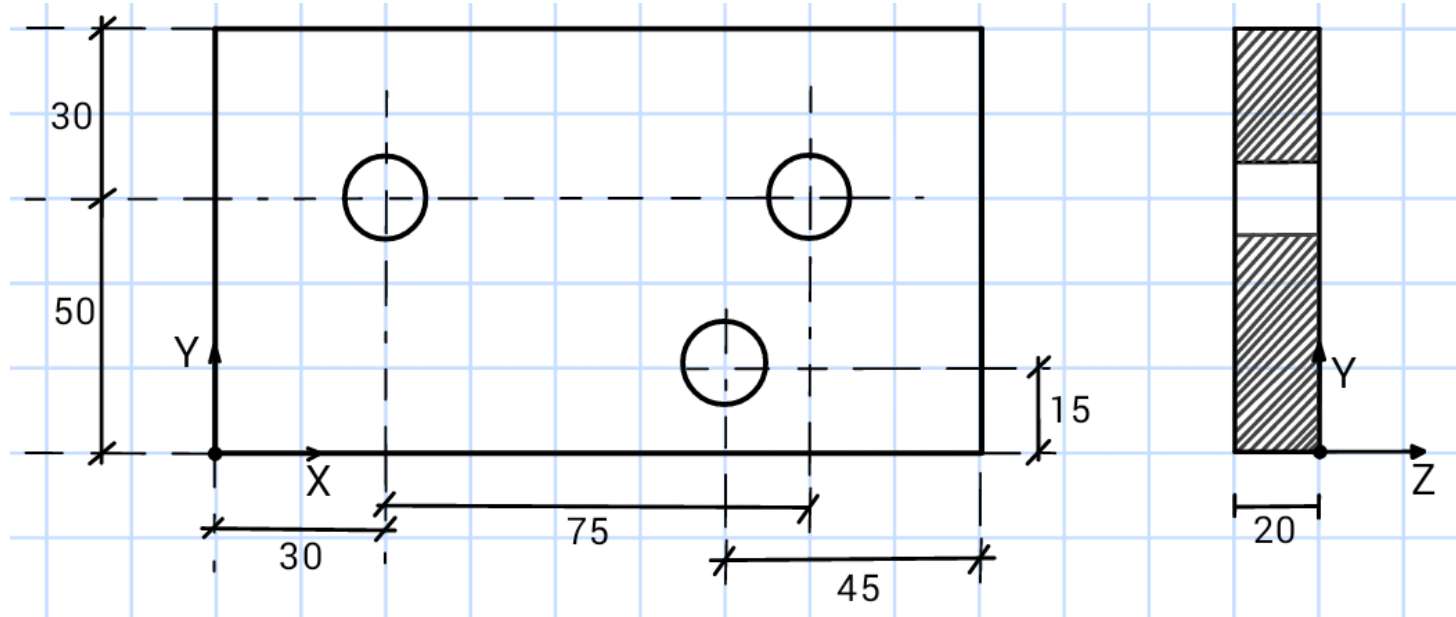
- Utilizzo utensile 04 (configurazione 02) con rotazione oraria del mandrino ed utilizzo lubrificante
- Velocità di taglio 100 mm/min
- Velocità di avanzamento di 6000 giri/min



# Esercizio 4 - Soluzione

N10	G90	T0402	M06	Abilitazione quote assolute e selezione tool 04 e conf. 02			
	G94	G97	S100	F6000	M03	M08	Settaggi mandrino ed avvio lubrificante
N20	G00	X20	Y20	Z10	Movimento rapido sopra al punto di foratura		
N30	G01	Z-11	foratura				
N40	G02	X70	Y70	I70	J20	Arco in senso orario	
N50	G00	Z10					
N60	G00	X70	Y20				
N70	G01	Z-11					
N80	G03	X120	Y70	I70	J70	Arco in senso anti-orario	
N90	G00	Z10					
N100	G00	X-10	Y-10	M05	M09	M30	Movimento veloce lontano dal pezzo, spegnimento del mandrino, del lubrificante e fine programma

# Esercizio 5



Si realizzi un software per CN che consenta di eseguire i fori riportati nella lavorazione rappresentata, con:

- Utilizzo utensile 02 (configurazione 03) con rotazione oraria del mandrino ed utilizzo lubrificante
- Velocità di avanzamento 0.2 mm/giro
- Velocità di taglio di 1200 giri/min

# Esercizio 5 - Soluzione

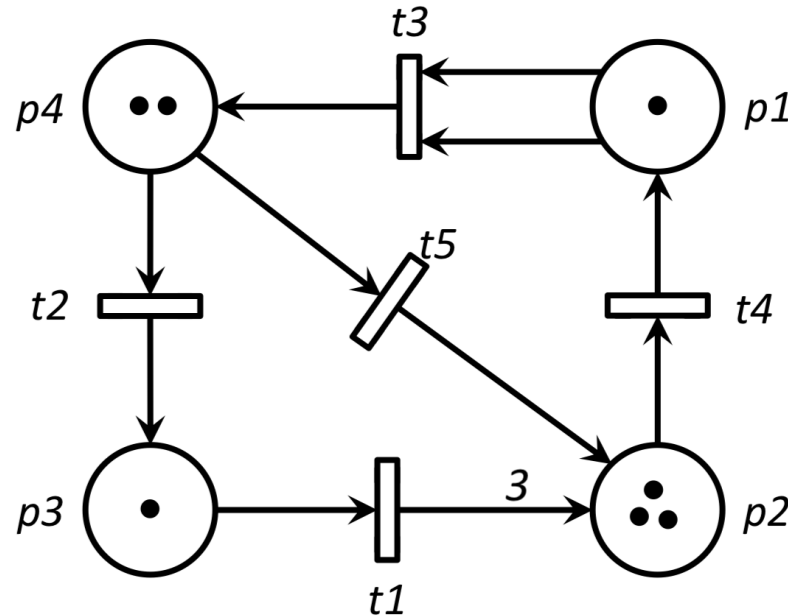
N10	G90	T0203	M06	→	Abilitazione quote assolute e selezione tool 02 e conf. 03			
	G95	G97	F0.2	S1200	M03	M08	→	Settaggi mandrino ed avvio lubrificante
N20	G00	X30	Y50	Z10	→	Movimento rapido fino a sopra il punto di foratura		
N30	G01	Z-22	→	Movimento di foratura				
N40	G00	Z10						
N50	G00	X90	Y15					
N60	G01	Z-22						
N70	G00	Z10						
N80	G00	X105	Y50					
N90	G01	Z-22						
N100	G00	Z10						
N110	G00	X-10	Y-10					
	M05	M09	M30	→	Spegnimento mandrino, lubrificante e fine programma			



# Reti di Petri

# Esercizio 1

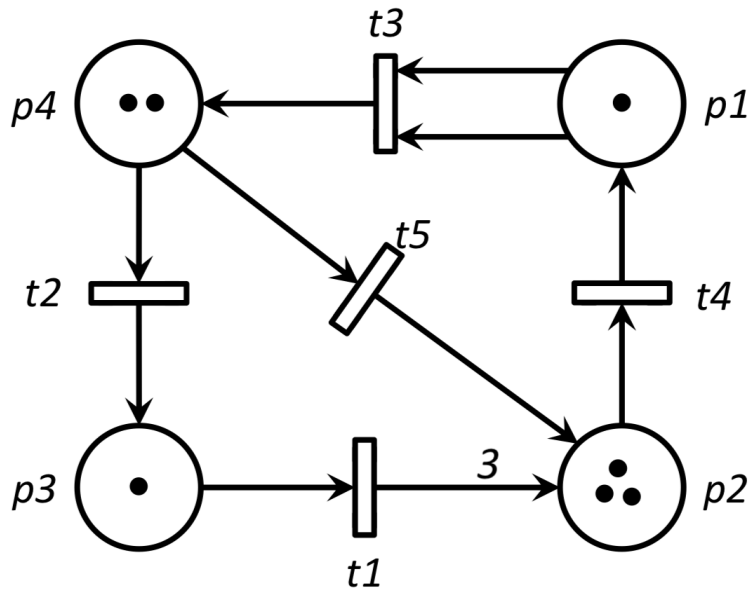
Per tutto questo esercizio consideriamo la seguente rete di Petri



Determinare:

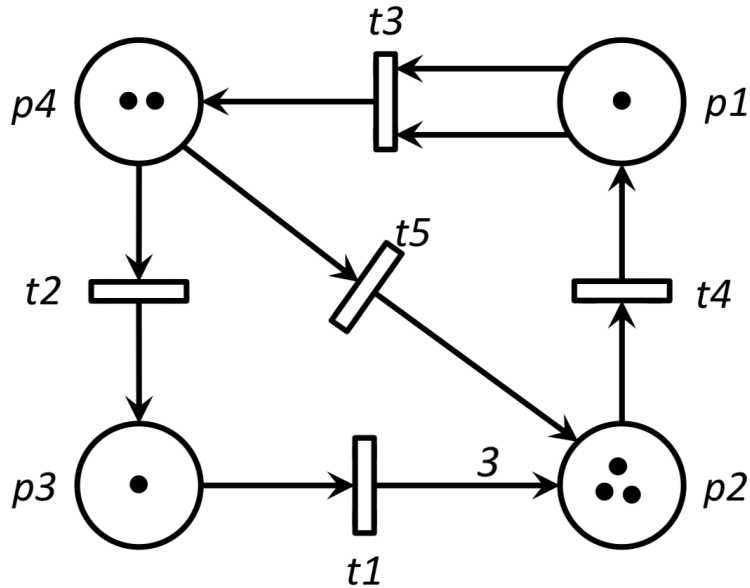
- Pre-set e Post-set di tutti i posti e tutte le transizioni
- Quali sono le transizioni abilitate
- Le marcature determinate dallo scatto delle transizioni abilitate
- Una sequenza ammissibile di transizioni che porti la rete alla marcatura  $M_f = [0 \ 3 \ 2 \ 3]^T$

# Esercizio 1– Soluzione Pre-Set e Post-Set



- $p1 = \{t4\}$
  - $p2 = \{t1, t5\}$
  - $p3 = \{t2\}$
  - $p4 = \{t3\}$
  - $t1 = \{p3\}$
  - $t2 = \{p4\}$
  - $t3 = \{p1\}$
  - $t4 = \{p2\}$
  - $t5 = \{p4\}$
- $p1 \bullet = \{t3\}$
  - $p2 \bullet = \{t4\}$
  - $p3 \bullet = \{t1\}$
  - $p4 \bullet = \{t2, t5\}$
  - $t1 \bullet = \{p2\}$
  - $t2 \bullet = \{p3\}$
  - $t3 \bullet = \{p4\}$
  - $t4 \bullet = \{p1\}$
  - $t5 \bullet = \{p2\}$

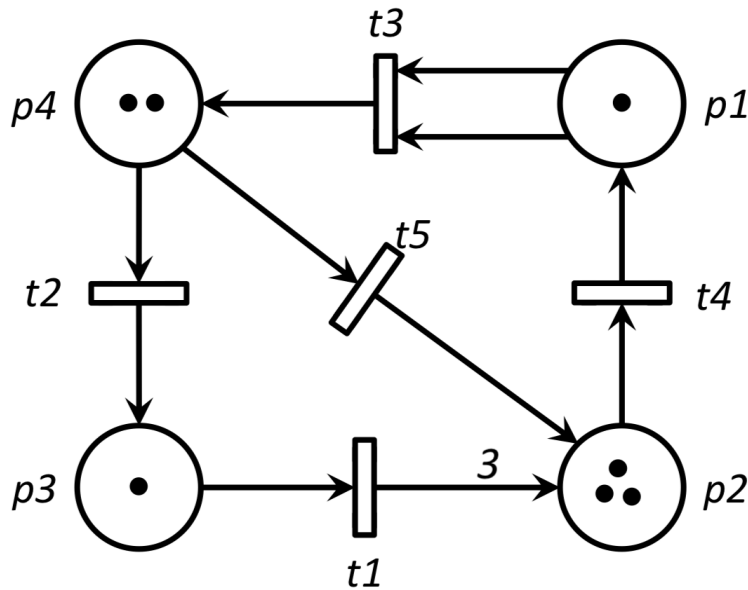
# Esercizio 1– Soluzione Transizioni abilitate



- $t1$  abilitata
- $t2$  abilitata
- $t3$  NON abilitata
- $t4$  abilitata
- $t5$  abilitata

# Esercizio 1– Soluzione evoluzione marcature

Partiamo dalla marcatura iniziale:



$$M_0 = [1 \quad 3 \quad 1 \quad 2]^T$$

Otteniamo:

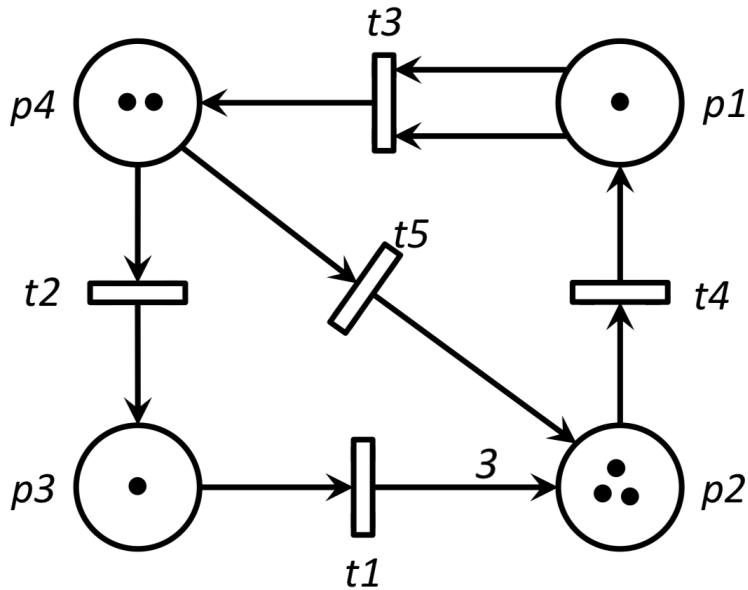
$$M_0[t_1 > M_1 = [0 \quad 6 \quad 1 \quad 2]^T$$

$$M_0[t_2 > M_2 = [1 \quad 3 \quad 2 \quad 1]^T$$

$$M_0[t_4 > M_3 = [2 \quad 2 \quad 1 \quad 2]^T$$

$$M_0[t_5 > M_4 = [1 \quad 4 \quad 1 \quad 1]^T$$

# Esercizio 1– Soluzione sequenza ammissibile di transizioni



Partiamo dalla marcatura iniziale:

$$M_0 = [1 \quad 3 \quad 1 \quad 2]^T$$

Vogliamo arrivare a:

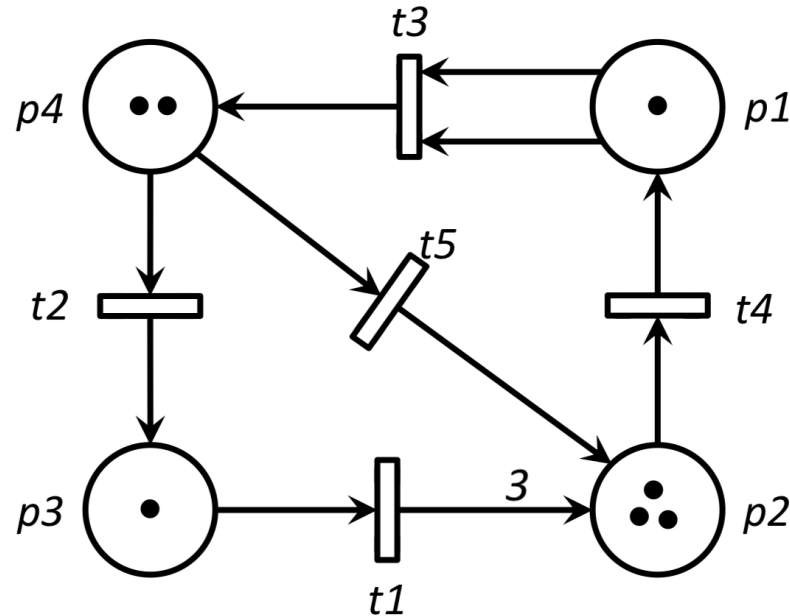
$$M_f = [0 \quad 3 \quad 2 \quad 3]^T$$

La sequenza corretta di transizioni è:

$$t_4, t_3, t_1, t_4, t_4, t_3, t_2, t_2$$

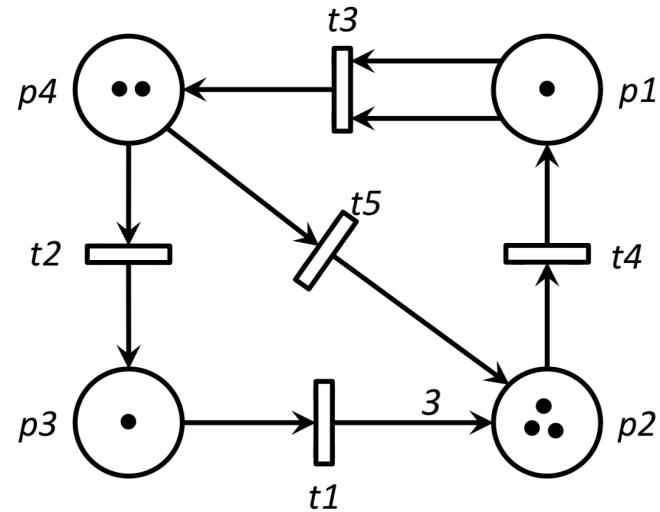
# Esercizio 2

Si consideri la seguente rete di Petri



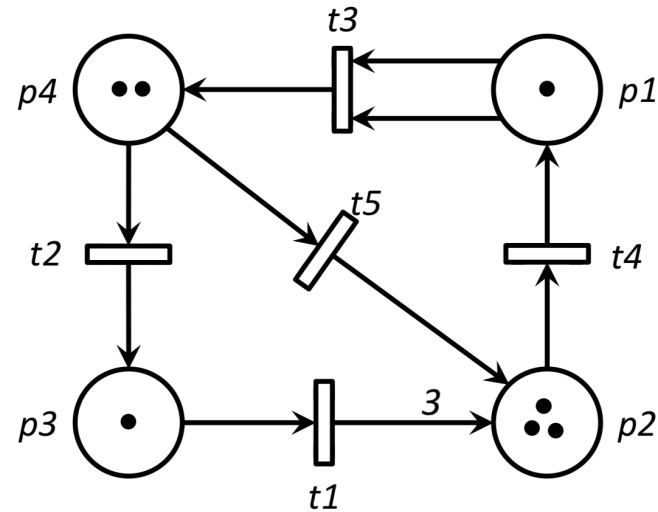
Si verifichi che le sequenze  $(t_1, t_4, t_3, t_2, t_5)$ ,  $(t_4, t_3, t_5, t_2, t_1)$ ,  $(t_2, t_4, t_1, t_5, t_3)$  sono ammissibili e portano alla medesima marcatura finale

# Esercizio 2 - Soluzione



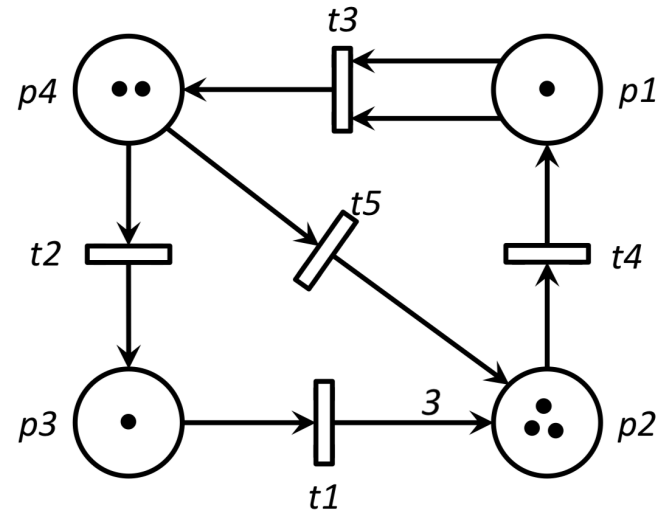
Trans.	$t_1$	$t_4$	$t_3$	$t_2$	$t_5$
$M_0$	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$
1	1	2	0	0	0
3	6	5	5	5	6
1	0	0	0	1	1
2	2	2	3	2	1

# Esercizio 2 - Soluzione



Trans.	$t_4$	$t_3$	$t_2$	$t_5$	$t_1$
$M_0$	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$
1	2	0	0	0	0
3	2	2	2	3	6
1	1	1	2	2	1
2	2	3	2	1	1

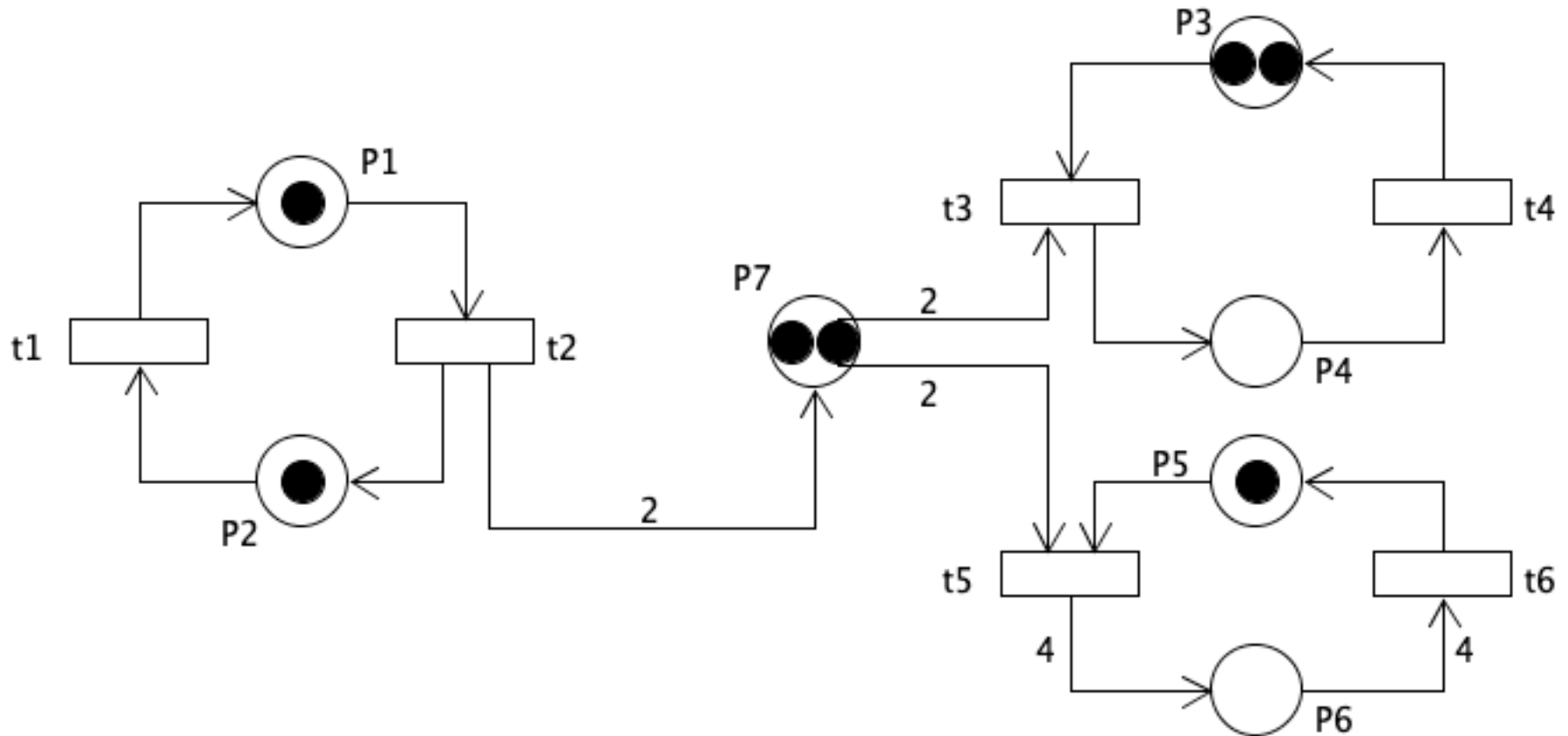
# Esercizio 2 - Soluzione



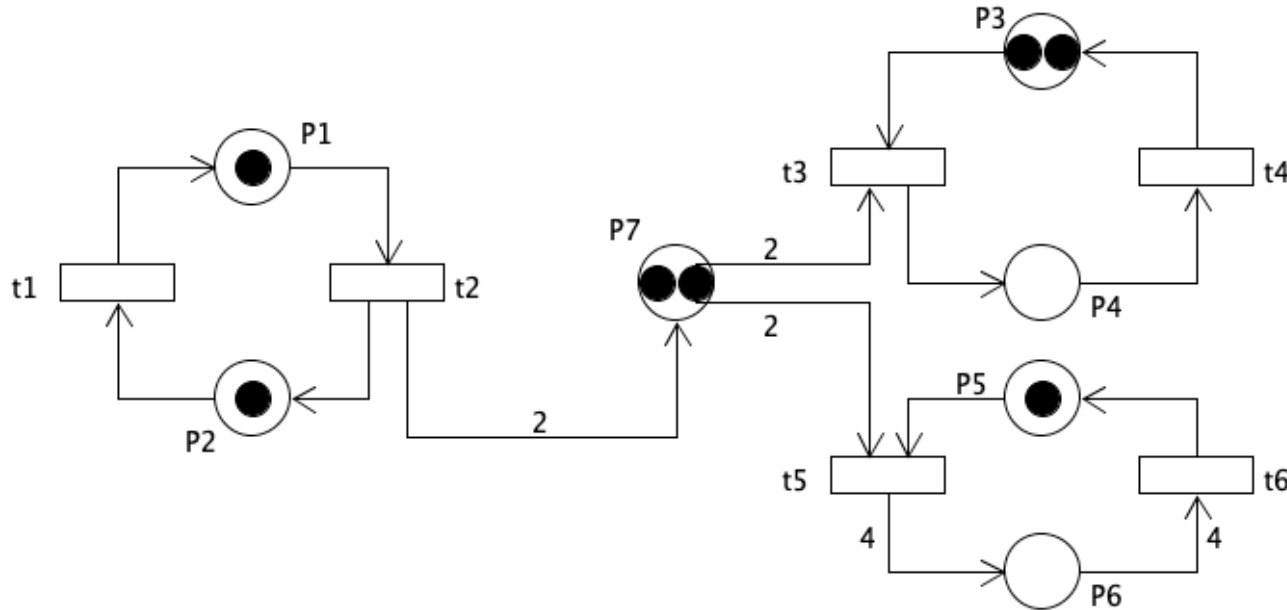
Trans.	$t_2$	$t_4$	$t_1$	$t_5$	$t_3$
$M_0$	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$
1	1	2	2	2	0
3	3	2	5	6	6
1	2	2	1	1	1
2	1	1	1	0	1

# Esercizio 3

Definire le proprietà e la tipologia della seguente rete di Petri:



# Esercizio 3 - Soluzione



## Proprietà:

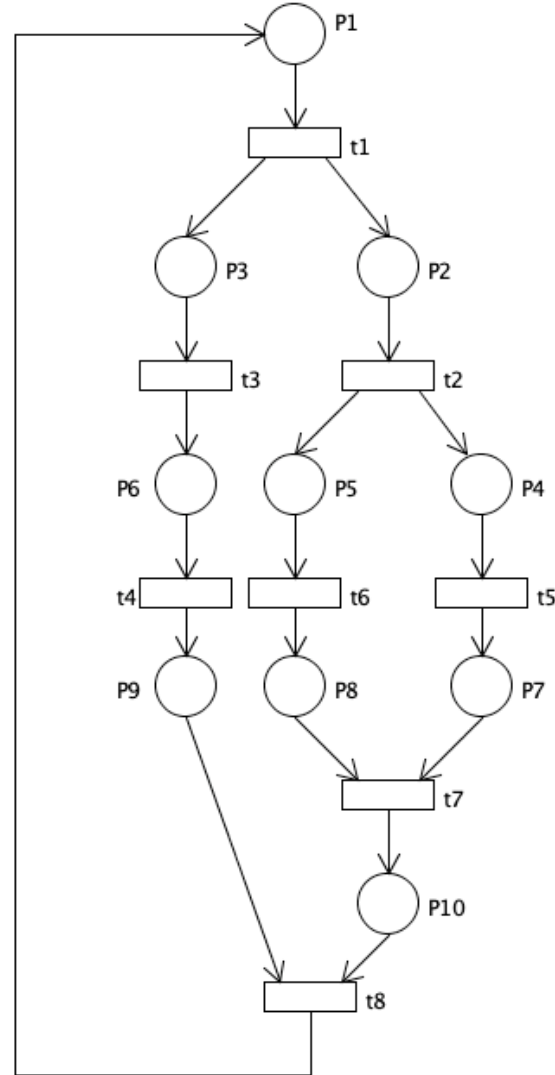
- Limitatezza -> NO ( $p7$  è illimitato)
- Vivezza -> SI
- Reversibilità -> SI (*essendo un sistema di produttori-consumatori*)

## Tipologia

- SM -> NO
- MG -> NO
- FC -> NO
- EFC -> NO
- AFC -> SI

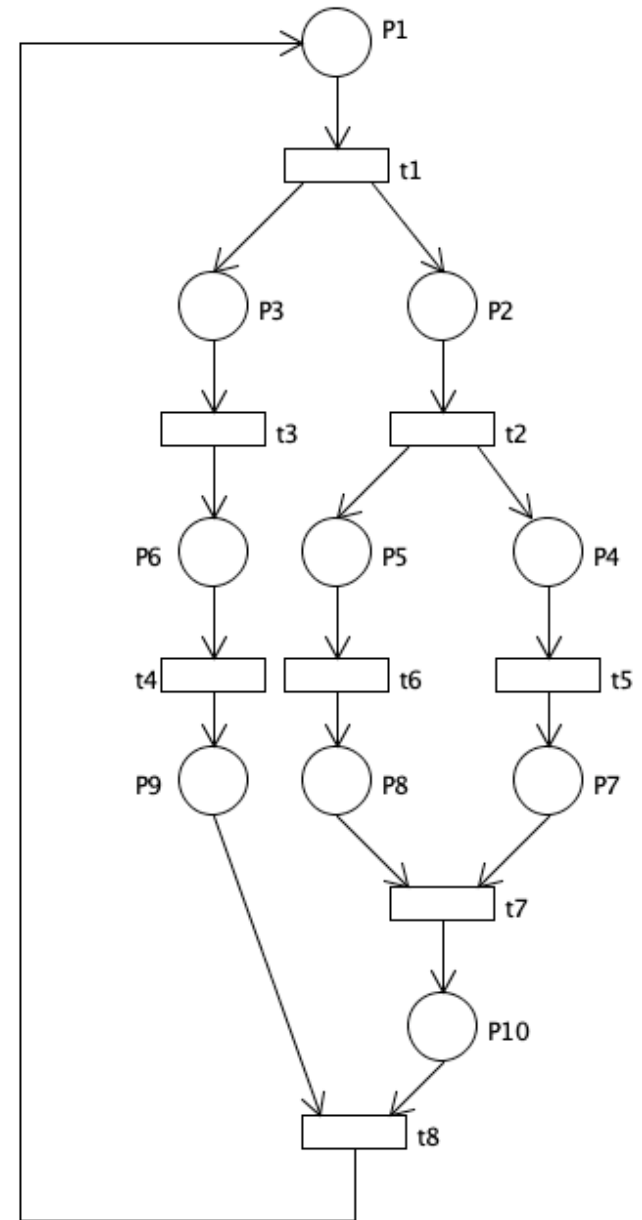
# Esercizio 4

Identificare, all'interno della rete di Petri seguente, le strutture base:



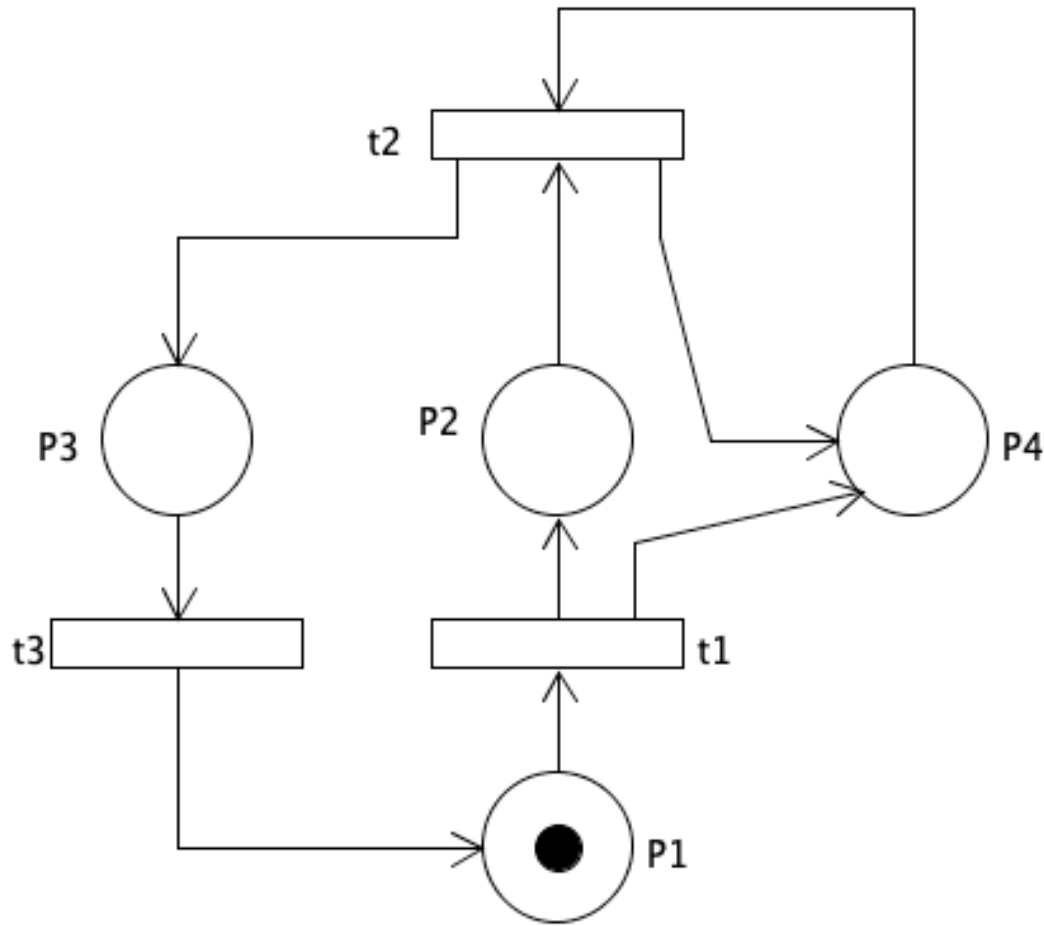
# Esercizio 4 - Soluzione

- Sequenze
  - $p1, p3, p6, p9$
  - $p1, p2, p5, p8, p10$
  - $p1, p2, p4, p7, p10$
- Transizioni di concorrenza
  - $t1, t2$
- Transizioni di sincronizzazione
  - $t7, t8$

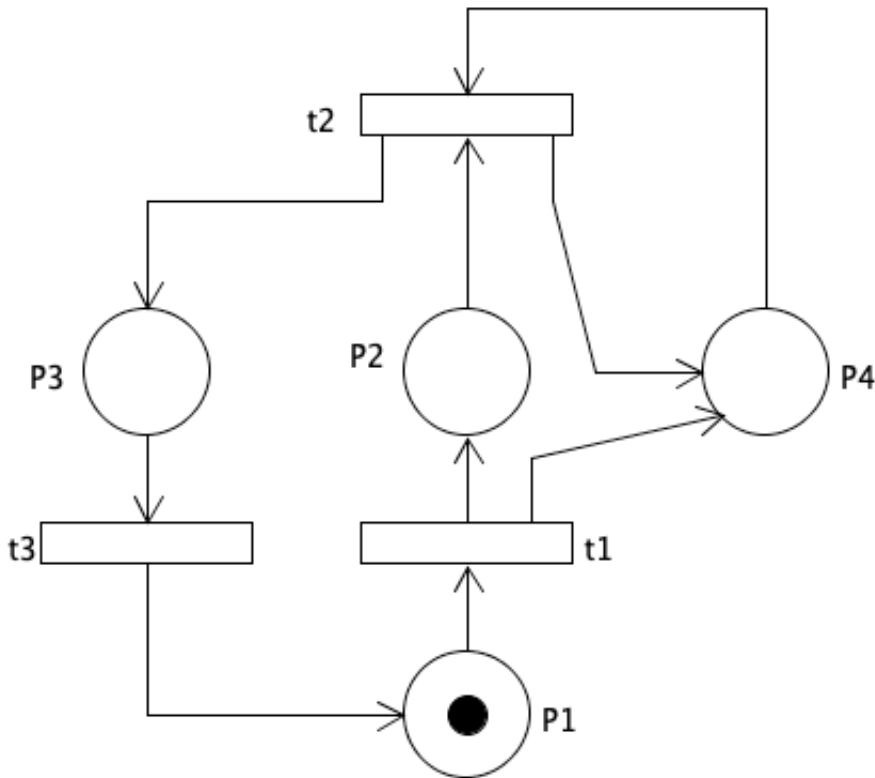


# Esercizio 5

Definire le proprietà e la tipologia della seguente rete di Petri:



# Esercizio 5 - Soluzione



## Proprietà:

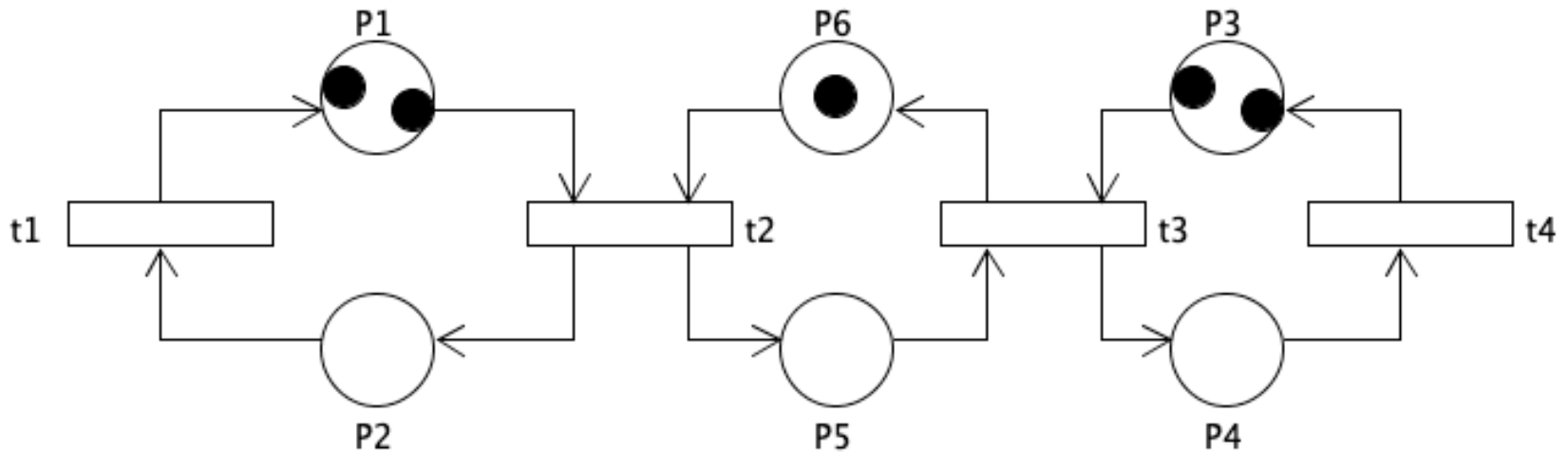
- Limitatezza -> NO (si veda la sequenza  $t1, t2, t3$  che porta l'accumulazione di token in  $P4$ )
- Vivezza -> SI
- Reversibilità -> NO (in  $P4$  si accumulano token)

## Tipologia

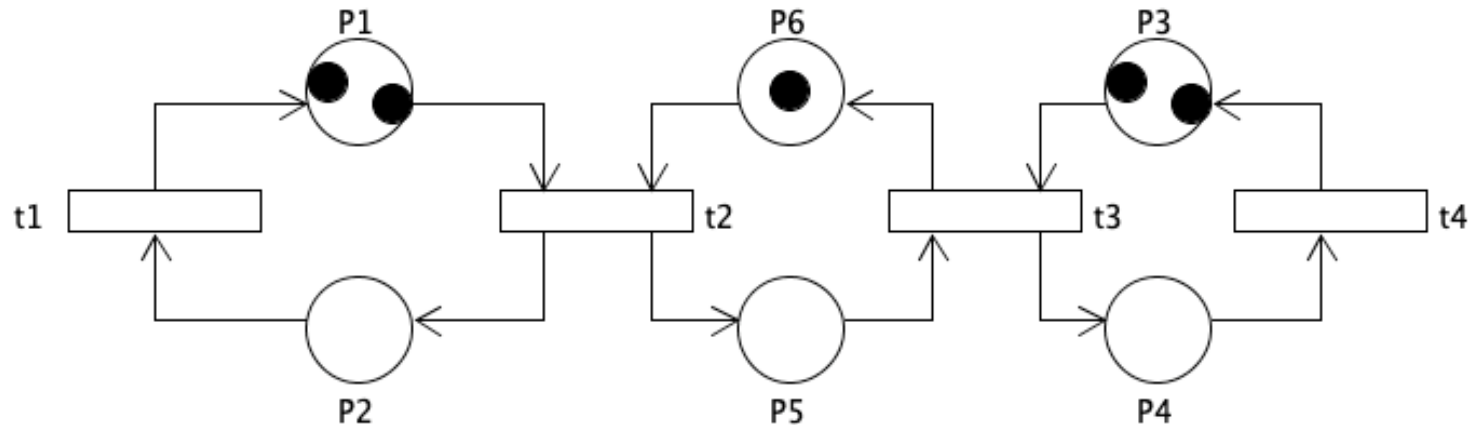
- SM -> NO
- MG -> NO
- FC -> SI
- EFC -> SI
- AFC -> SI

# Esercizio 6

Definire le proprietà e la tipologia della seguente rete di Petri:



# Esercizio 6 - Soluzione



## Proprietà:

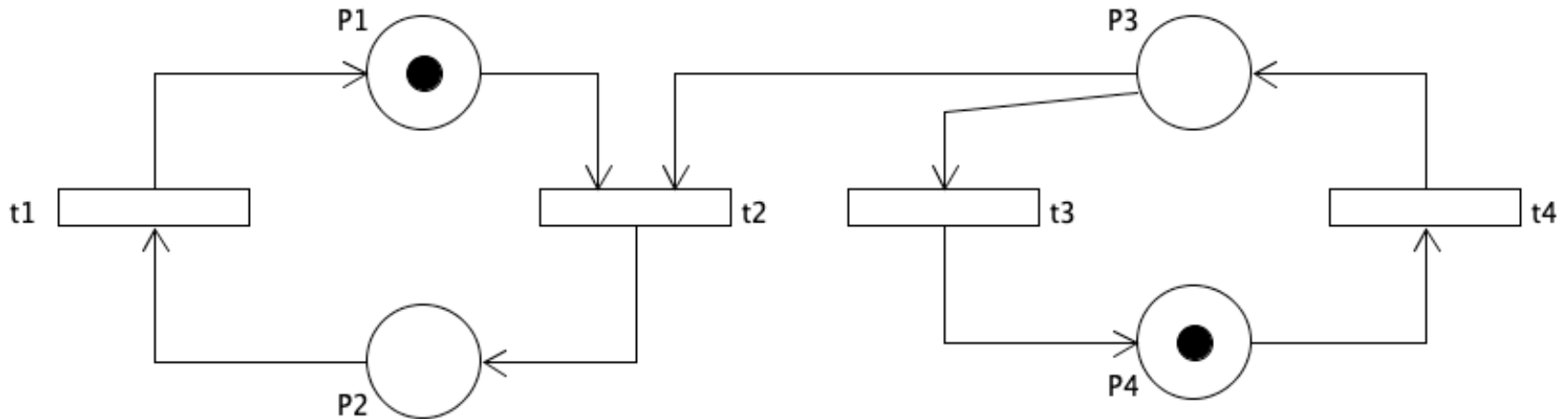
- Limitatezza -> SI
- Vivezza -> SI
- Reversibilità -> SI (*sistema di produttori e consumatori*)

## Tipologia

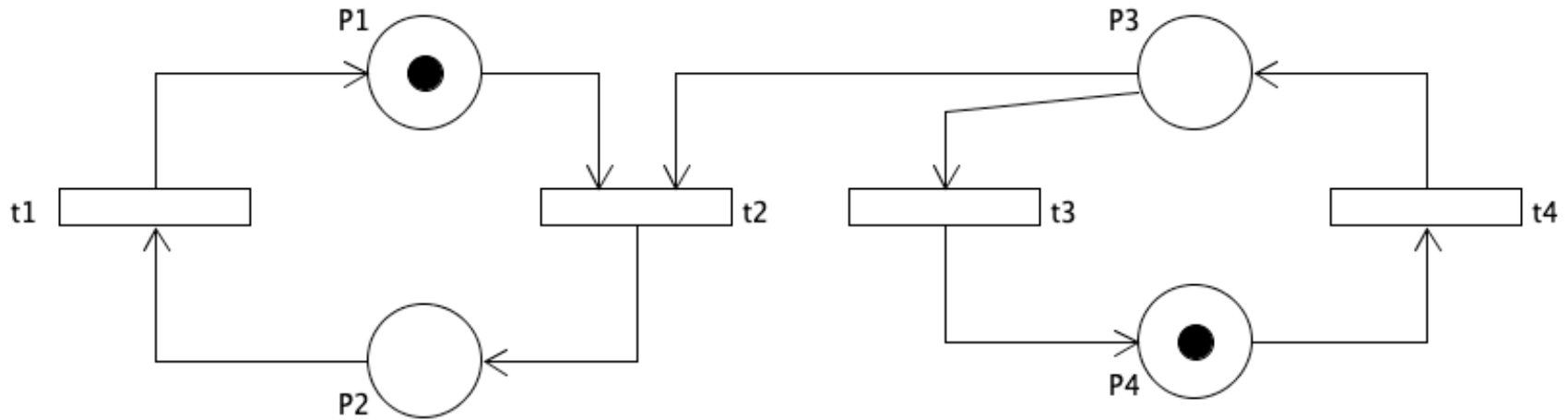
- SM -> NO
- MG -> SI
- FC -> SI
- EFC -> SI
- AFC -> SI

# Esercizio 7

Definire le proprietà e la tipologia della seguente rete di Petri:



# Esercizio 7 - Soluzione



## Proprietà:

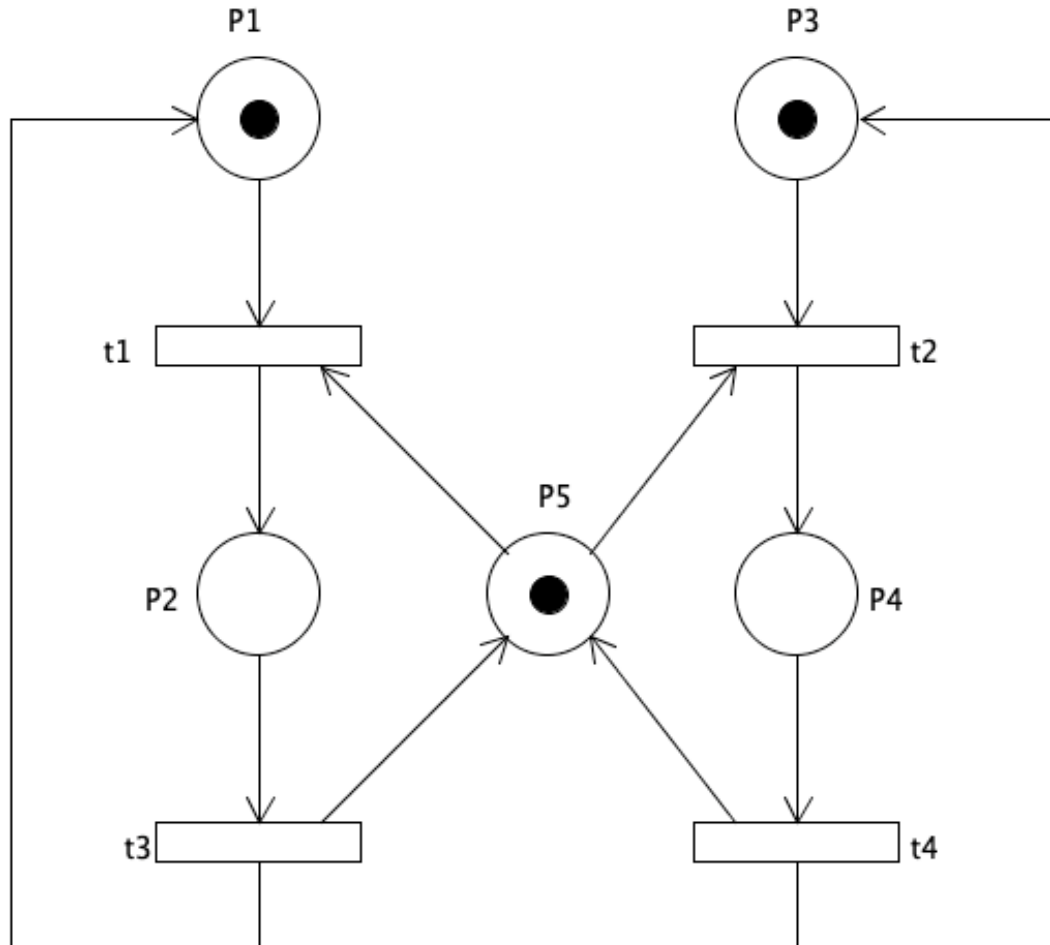
- Limitatezza -> SI
- Vivezza -> SI
- Reversibilità -> SI

## Tipologia

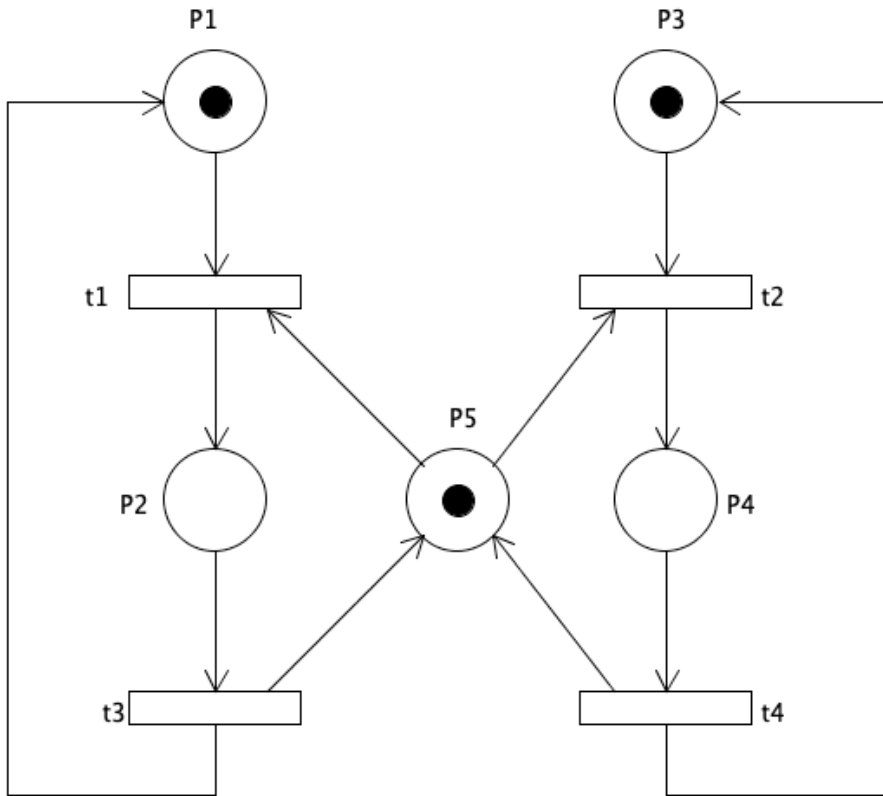
- SM -> NO
- MG -> NO
- FC -> NO
- EFC -> NO
- AFC -> SI (i post-set di  $p1$  e  $p3$  sono uno il sottoinsieme dell'altro)

# Esercizio 8

Definire le proprietà e la tipologia della seguente rete di Petri:



# Esercizio 8 - Soluzione



## Proprietà:

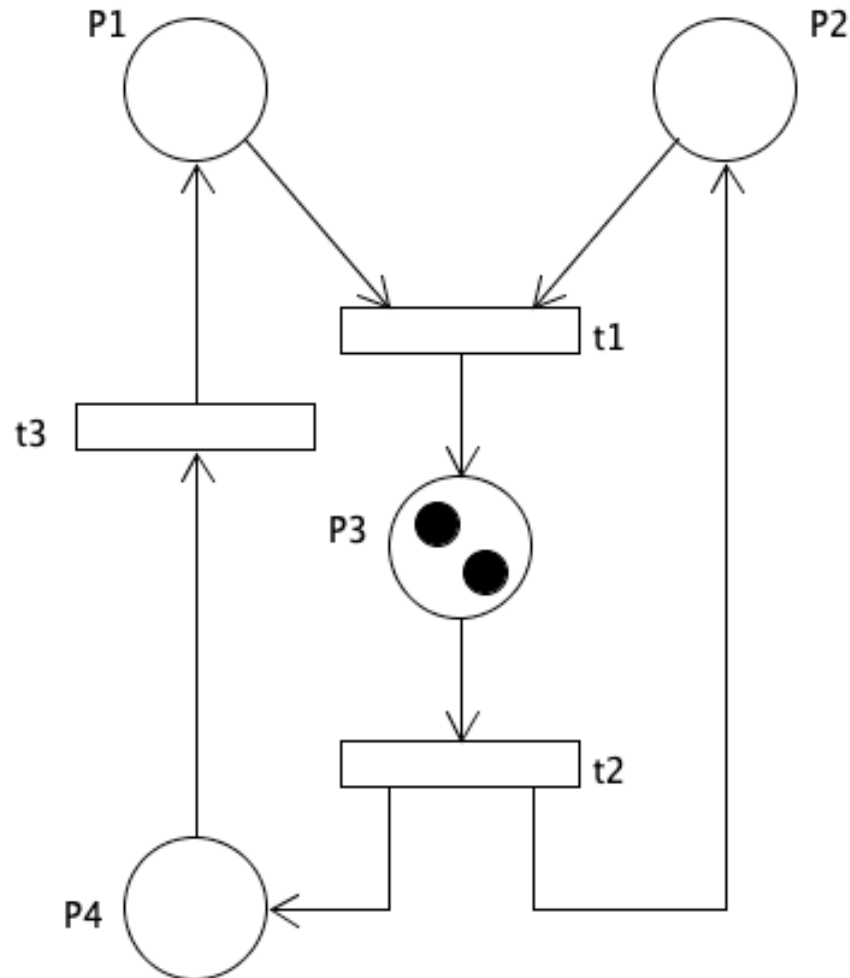
- Limitatezza -> SI
- Vivezza -> SI
- Reversibilità -> SI

## Tipologia

- SM -> NO
- MG -> NO
- FC -> NO
- EFC -> NO
- AFC -> SI (post-set di  $p5$  e  $p1$  sono uno il sottoinsieme dell'altro, come per  $p5$  e  $p3$ )

# Esercizio 9

Definire le proprietà e la tipologia della seguente rete di Petri:



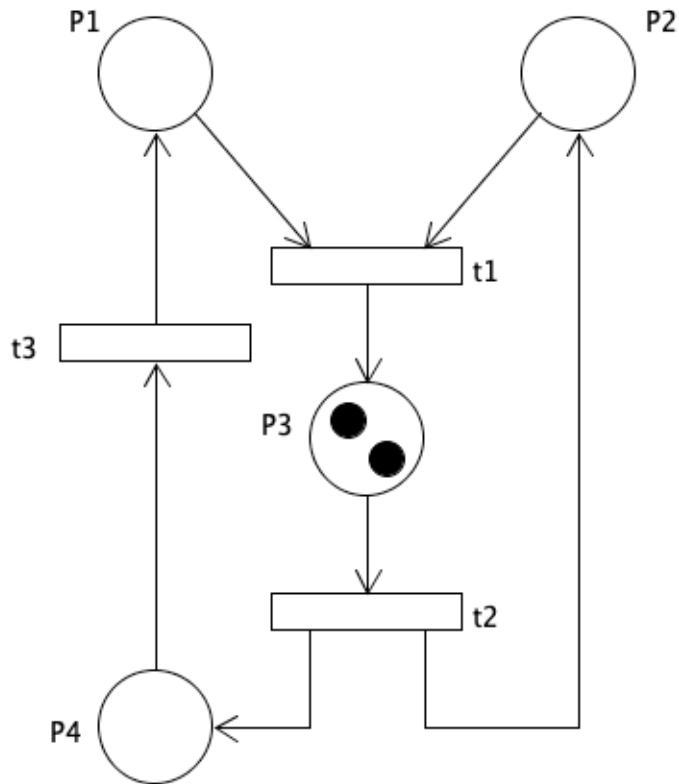
# Esercizio 9 - Soluzione

## Proprietà:

- Limitatezza -> SI
- Vivezza -> SI
- Reversibilità -> SI

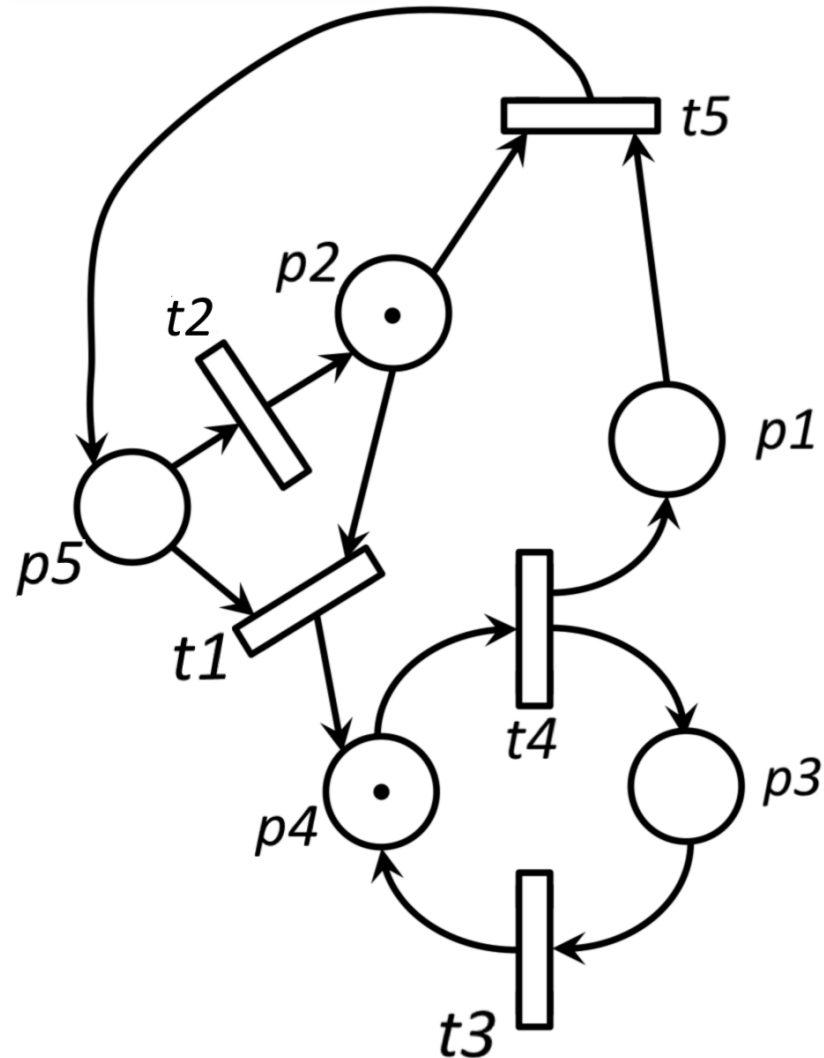
## Tipologia

- SM -> NO
- MG -> SI
- FC -> SI
- EFC -> SI
- AFC -> SI

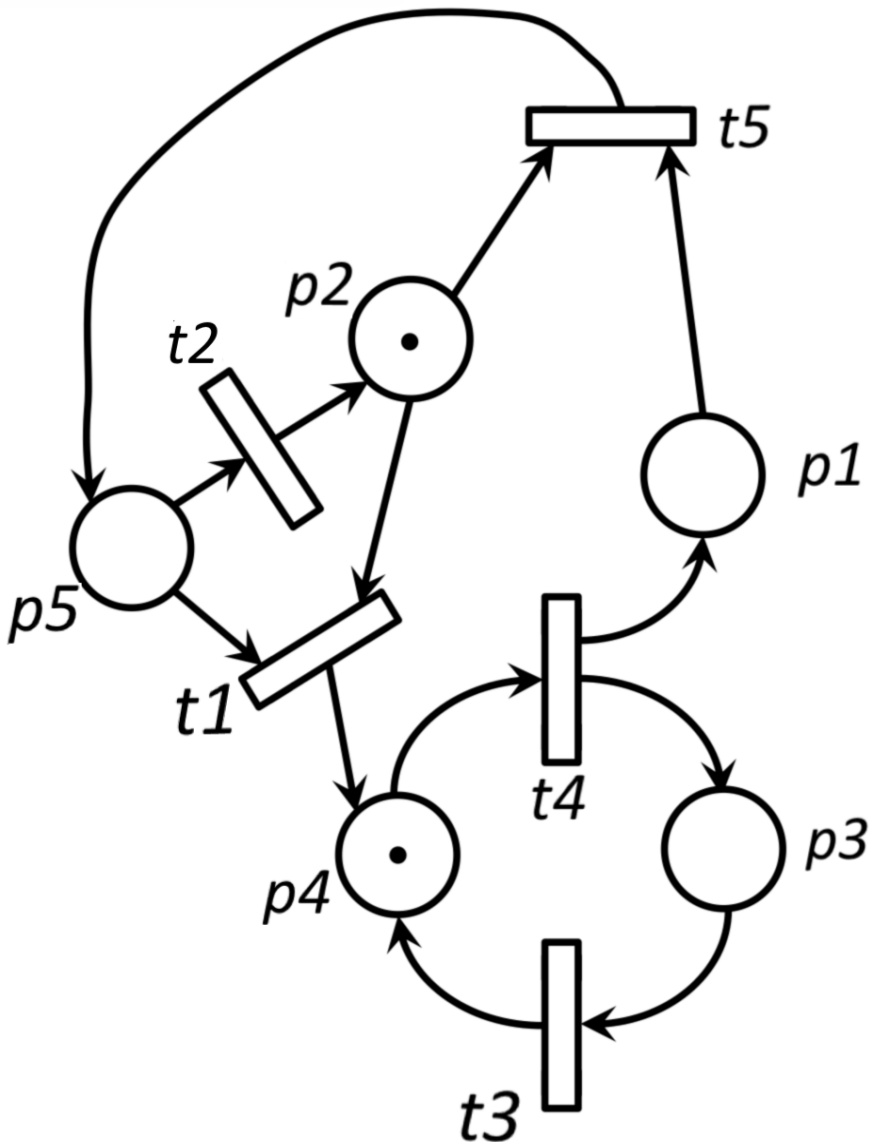


# Esercizio 10

Definire le proprietà e la tipologia della seguente rete di Petri:



# Esercizio 10 - Soluzione



## Proprietà:

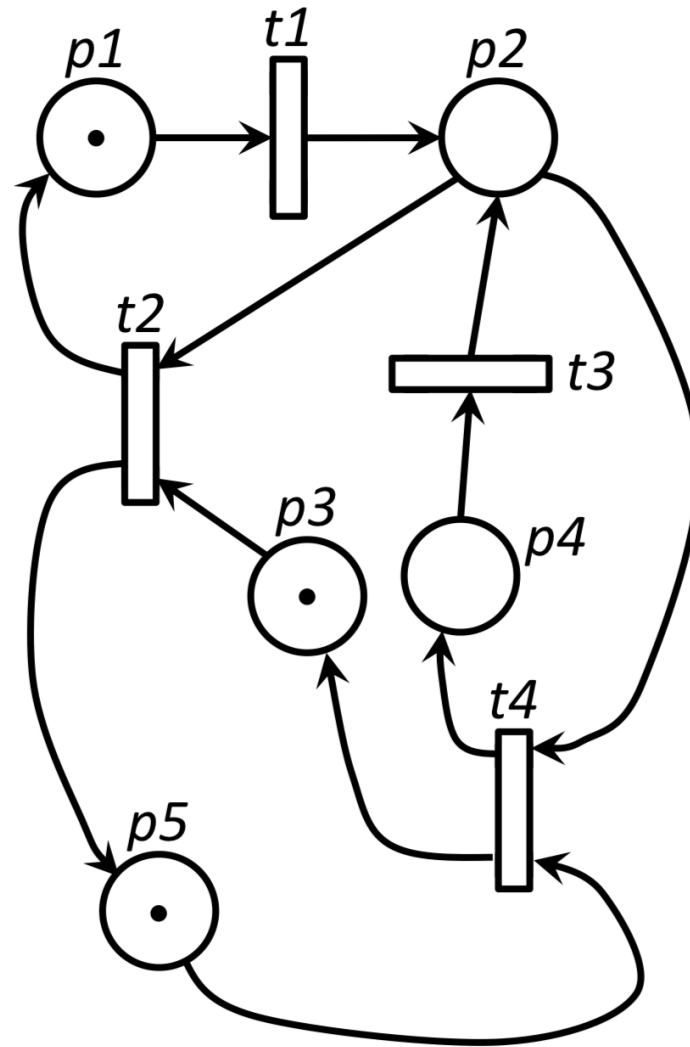
- Limitatezza  $\rightarrow$  NO ( $p_1$  è illimitato)
- Vivezza  $\rightarrow$  NO ( $t_1$  non è mai abilitata)
- Reversibilità  $\rightarrow$  SI (i token in  $p_1$  possono essere consumati con  $t_2$  e  $t_5$ )

## Tipologia

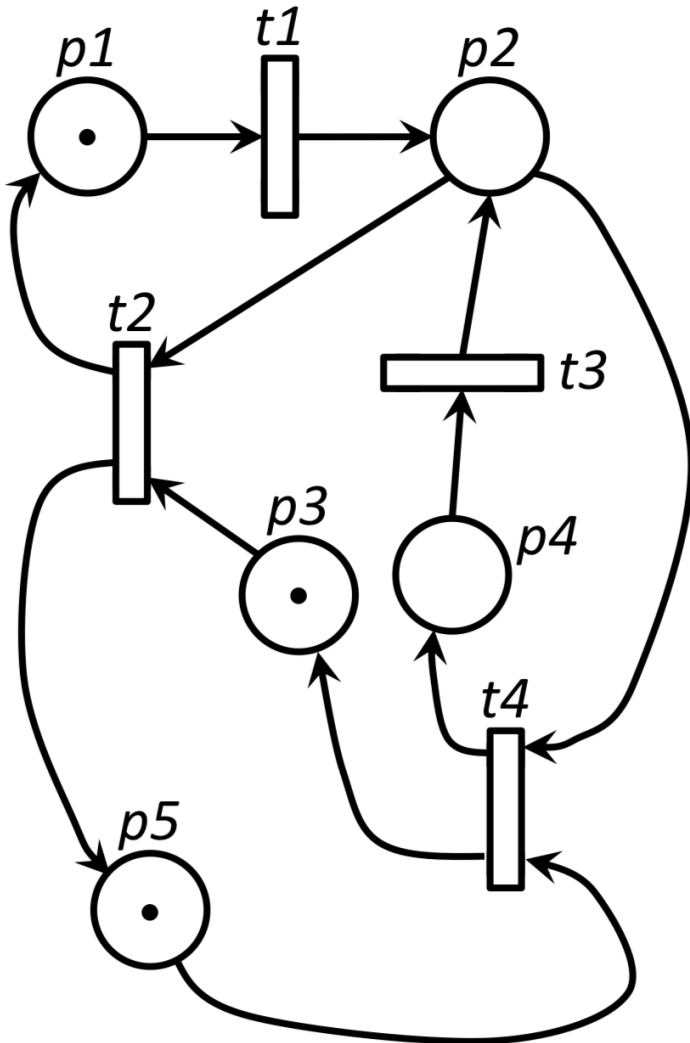
- SM  $\rightarrow$  NO
- MG  $\rightarrow$  NO
- FC  $\rightarrow$  NO
- EFC  $\rightarrow$  NO
- AFC  $\rightarrow$  NO (tra i post-set di  $p_2$  e  $p_5$  c'è confusione)

# Esercizio 11

Definire le proprietà e la tipologia della seguente rete di Petri:



# Esercizio 11 - Soluzione



## Proprietà:

- Limitatezza  $\rightarrow$  NO ( $p_5$  è illimitato)
- Vivezza  $\rightarrow$  SI
- Reversibilità  $\rightarrow$  NO

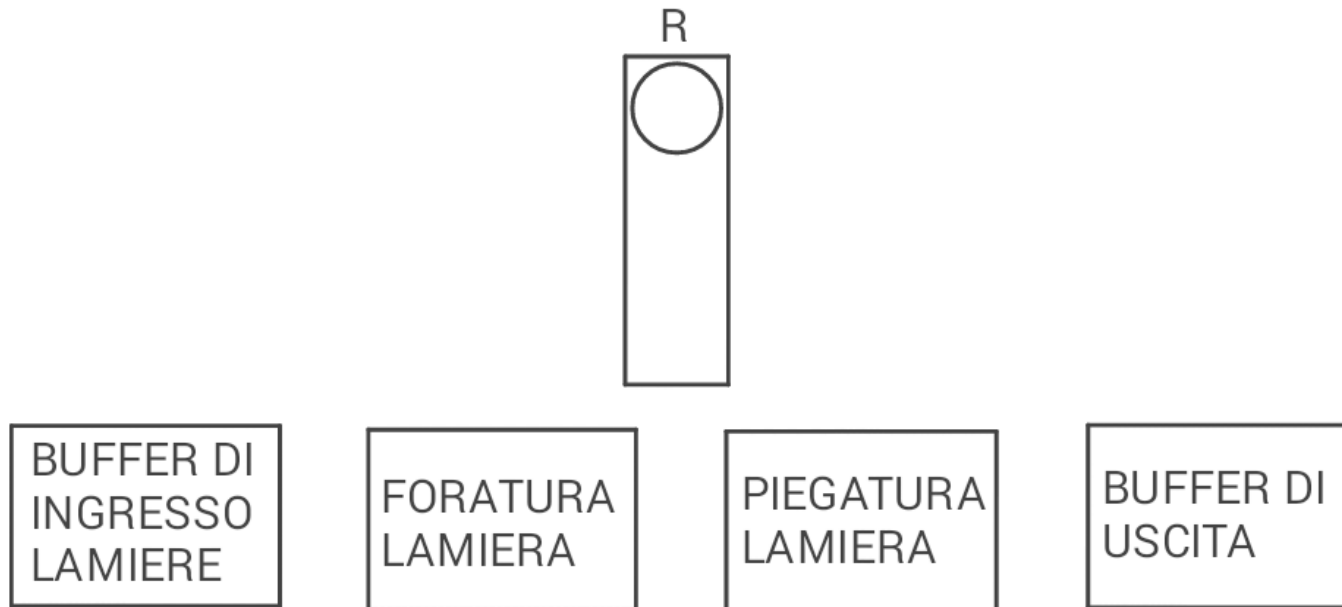
## Tipologia

- SM  $\rightarrow$  NO
- MG  $\rightarrow$  NO
- FC  $\rightarrow$  NO
- EFC  $\rightarrow$  NO
- AFC  $\rightarrow$  SI (i post-set di  $p_3$  e  $p_5$  sono sottoinsiemi del post-set di  $p_2$ )

# Esercizio 12

Modellare il seguente processo produttivo utilizzando le tre diverse tecniche:

1. Modello a due eventi
2. Modello a un evento
3. Modello FMS



# Esercizio 12

## Ipotesi:

- I pezzi arrivano al buffer di ingresso automaticamente
- I pezzi vengono prelevati dal buffer di uscita automaticamente
- I buffer hanno capacità 2 pezzi

## Le **attività** del sistema sono:

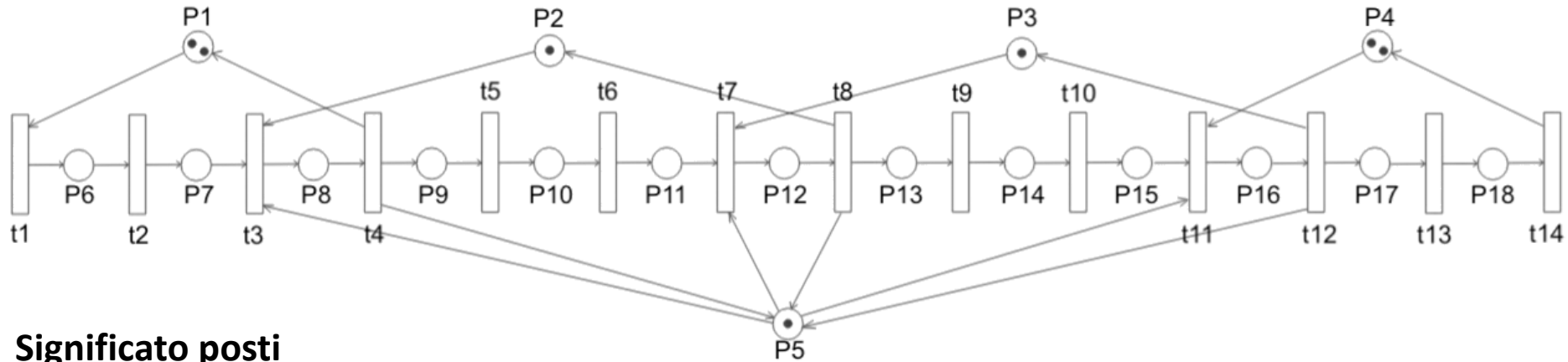
- Il buffer di ingresso riceve il pezzo
- *R* carica il pezzo prelevato sulla macchina di foratura
- La macchina di foratura lavora il pezzo
- *R* carica il pezzo prelevato dalla macchina di foratura sulla macchina di piegatura
- La macchina di piegatura piega il pezzo
- *R* carica il pezzo prelevato dalla macchina di foratura nel buffer di uscita
- Il buffer di uscita si svuota del pezzo

# Esercizio 12

Analizzando le attività e le sequenze di operazioni descritte, è possibile costruire la seguente tabella:

Attesa attività	Attività in corso	Conclusione attività
-	Arrivo pezzo nel buffer di ingresso	Pezzo da caricare nella foratrice
Pezzo da caricare nella foratrice	Caricamento pezzo nella foratrice	Pezzo caricato nella foratrice
Pezzo caricato nella foratrice	Foratura	Pezzo forato
Pezzo forato	Spostamento pezzo forato nella piegatrice	Pezzo spostato nella piegatrice
Pezzo spostato nella piegatrice	Piegatura	Pezzo piegato
Pezzo piegato	Spostamento pezzo nel buffer di uscita	Pezzo scaricato nel buffer di uscita
Pezzo scaricato nel buffer di uscita	Svuotamento buffer di uscita	

# Esercizio 12 – Soluzione – Modello a due eventi



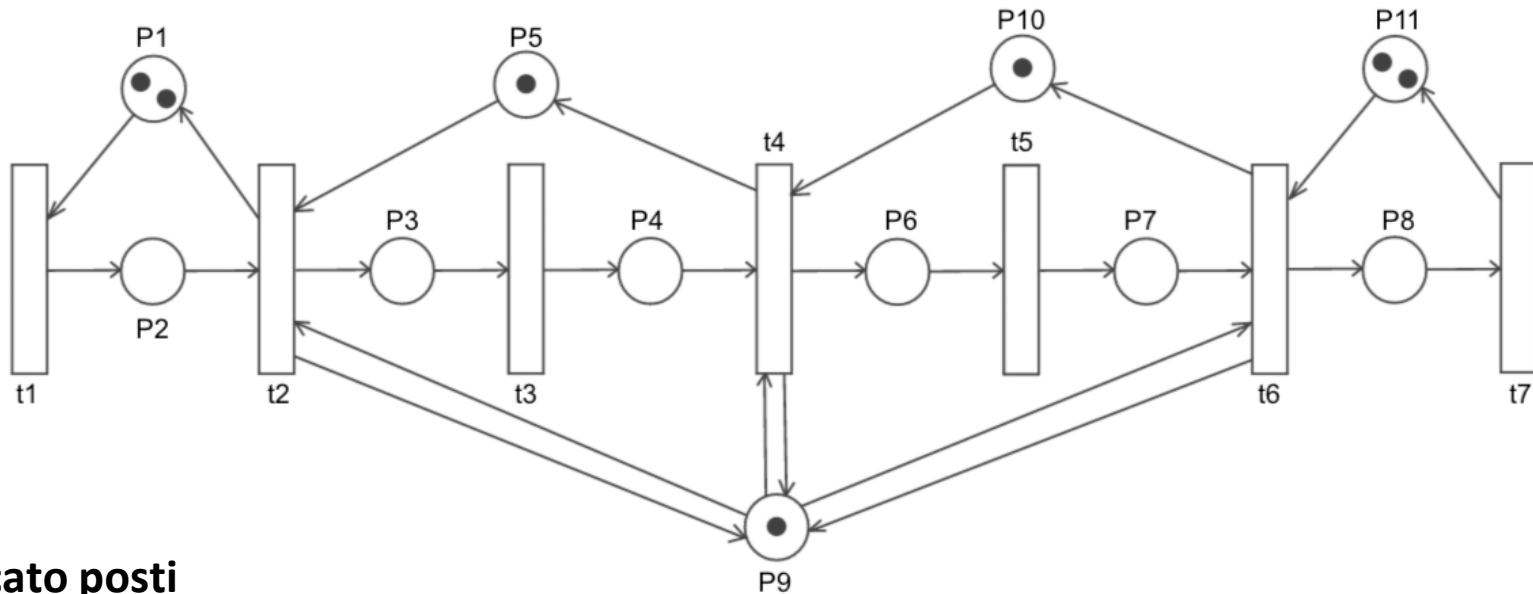
## Significato posti

- P1: Buffer IN disponibile
- P2: Foratrice disponibile
- P3: Piegatrice disponibile
- P4: Buffer OUT disponibile
- P5: R disponibile
- P6: Arrivo in Buffer IN
- P7: Lamina da caricare
- P8: Caricamento in foratrice
- P9: Lamina caricata
- P10: Foratura
- P11: Pezzo forato
- P12: Pezzo da spostare
- P13: Pezzo in piegatrice
- P14: Piegatura
- P15: Pezzo piegato
- P16: Pezzo da scaricare
- P17: Pezzo scaricato
- P18: Uscita da buffer OUT

## Significato transizioni

- T1: Inizio ingresso
- T2: Fine ingresso
- T3: Inizio caricamento
- T4: Fine caricamento
- T5: Inizio foratura
- T6: Fine foratura
- T7: Inizio spostamento in pieg.
- T8: Fine spostamento in pieg.
- T9: Inizio piegatura
- T10: Fine piegatura
- T11: Inizio scaricamento
- T12: Fine scaricamento
- T13: Inizio uscita
- T14: Fine uscita

# Esercizio 12 – Soluzione – Modello a un evento



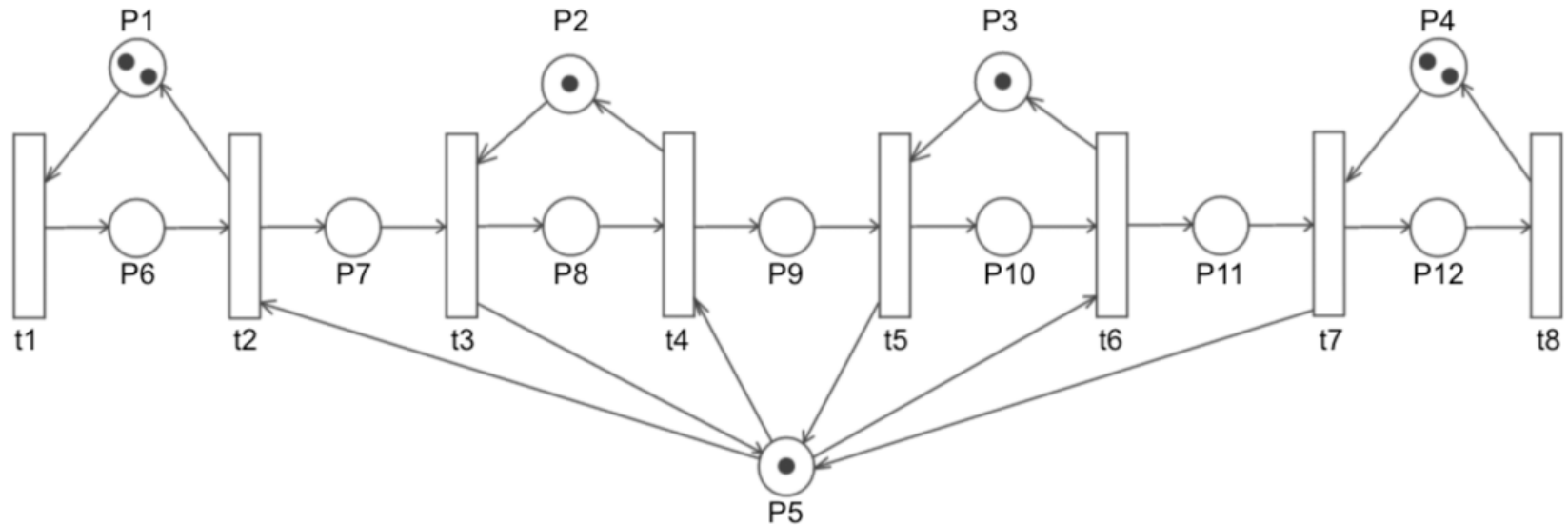
## Significato posti

- P1: Buffer IN disponibile
- P2: Lamiera da caricare
- P3: Lamiera in foratrice
- P4: Lamiera forata
- P5: Foratrice disponibile
- P6: Lamiera in piegatrice
- P7: Lamiera piegata
- P8: Lamiera scaricata
- P9: R disponibile
- P10: Piegatrice disponibile
- P11: Buffer OUT disponibile

## Significato transizioni

- T1: Ingresso
- T2: Caricamento in foratrice
- T3: Foratura
- T4: Spost. in piegatrice
- T5: Piegatura
- T6: Spostamento in OUT
- T7: Uscita

# Esercizio 12 – Soluzione – Modello FMS



## Significato posti

- P1: Buffer IN disponibile
- P2: Foratrice disponibile
- P3: Piegatrice disponibile
- P4: Buffer OUT disponibile
- P5: R disponibile
- P6: Occupazione buffer IN
- P7: Spostamento IN-Forat.
- P8: Foratura
- P9: Spostamento Forat.-Pieg.
- P10: Piegatura
- P11: Spostamento in OUT
- P12: Occupazione buffer OUT

## Significato transizioni

- T1: Inizio ingresso
- T2: Inizio caricamento in Forat.
- T3: Inizio foratura
- T4: Inizio spost. in piegatrice
- T5: Inizio piegatura
- T6: Inizio spostamento in OUT
- T7: Inizio uscita
- T8: Fine uscita