

Gestione Domande di lavoro

Il processo B2B GestioneDomande opera in un'agenzia che tratta domande e offerte di lavoro, le prime inviate da candidati e le seconde da aziende. Nel sistema informativo sono registrati i candidati, le aziende, le aree (di interesse) e le competenze relative alle aree. Ad ogni area è associato un gestore (ruolo di staff). Domande e offerte si riferiscono ad un'area.

Ricevuta una domanda il gestore dell'area definisce un quesito per ogni competenza dell'area della domanda (1). Poi il processo manda la domanda con i quesiti al candidato che risponde rimandando la domanda con i quesiti risolti. Il gestore decide se ammettere o respingere la domanda. Nel primo caso la domanda è collocata nel posto (Domanda, pendente). Nel secondo caso il processo informa il candidato che la domanda è stata respinta.

Il processo riceve le offerte e le colloca nel posto (Offerta, pendente). Il gestore fa corrispondere domande e offerte pendenti con un task che associa tre domande ad un'offerta. Le domande e l'offerta devono riferirsi alla stessa area (1) e sono tolte dai posti corrispondenti. Il processo invia l'offerta con le domande all'azienda che risponde indicando la domanda accettata con l'interazione (da, Domanda). Il processo informa i candidati con le interazioni da o dr (domanda accettata o domanda respinta).

(1) Si esprima il vincolo con un invariante.

Candidato

processo

processo

Azienda

Domanda ref Area



dQ, Domanda



Quesito

dQr, Domanda



Quesito

Offerta ref Area

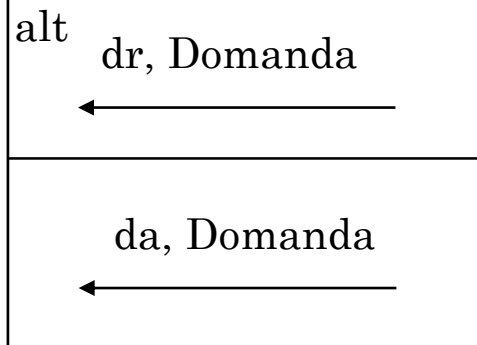


oD, Offerta



Domanda

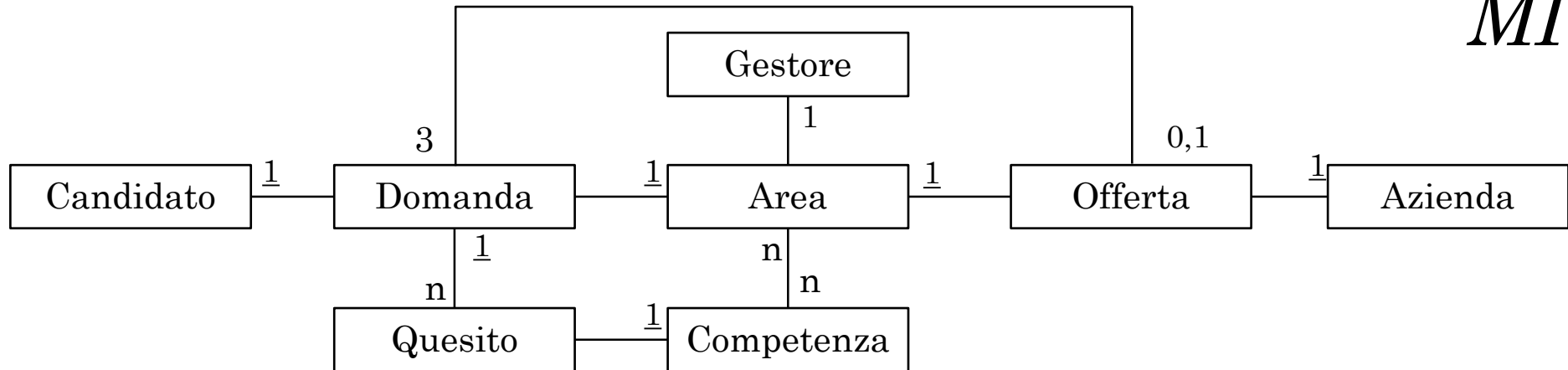
da, Domanda



Albero
Domanda – Area, Quesito

Albero
Offerta – Area, Domanda

da, dr = domanda accettata, respinta
dQ = domanda con quesiti
dQr = domanda con quesiti risolti
oD = offerta con domande



Attributi

Domanda: stato (accettata, respinta);

Invarianti

`domanda.quesiti.competenza == domanda.area.competenze.`

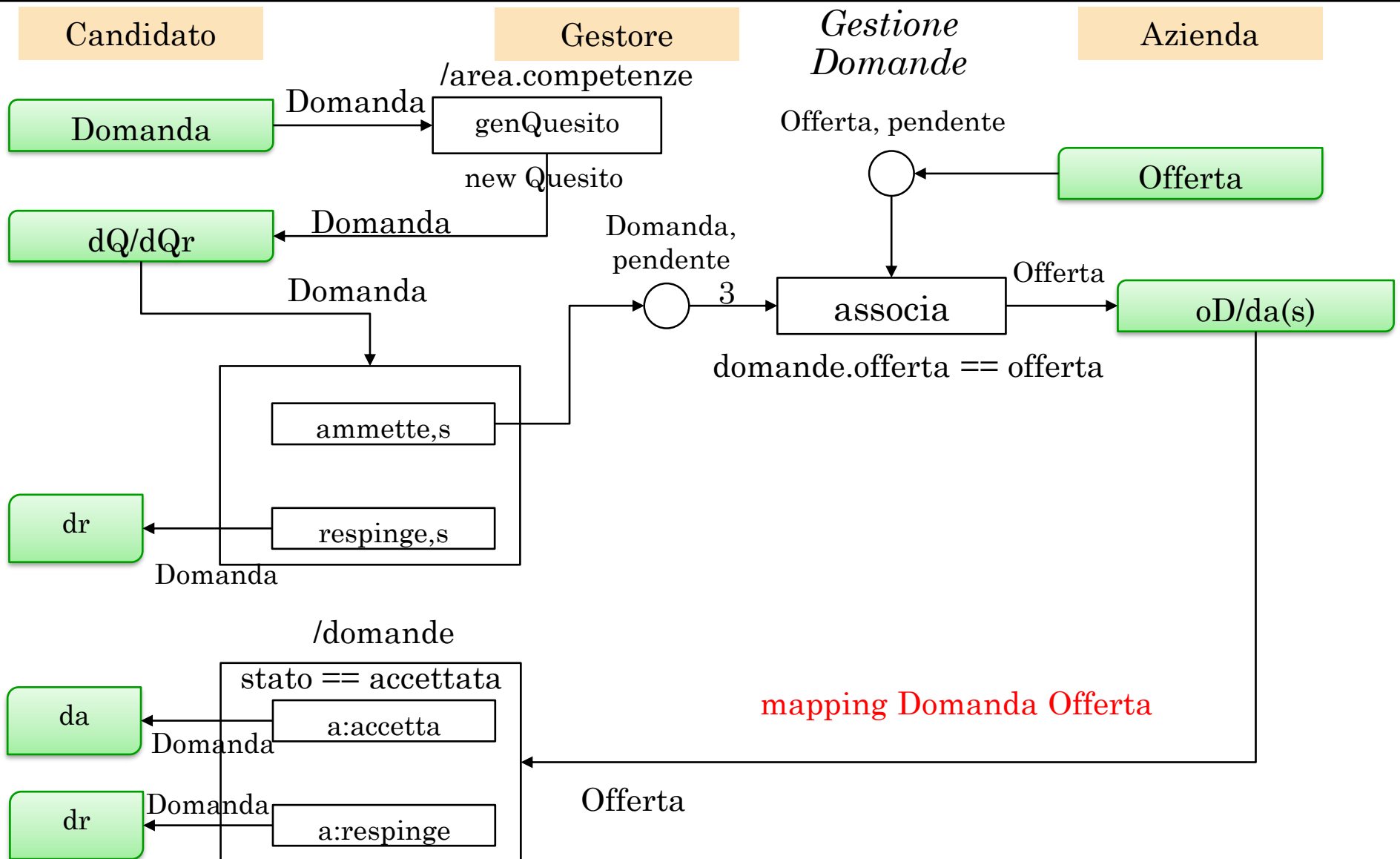
`domanda.area == domanda.offerta.area.`

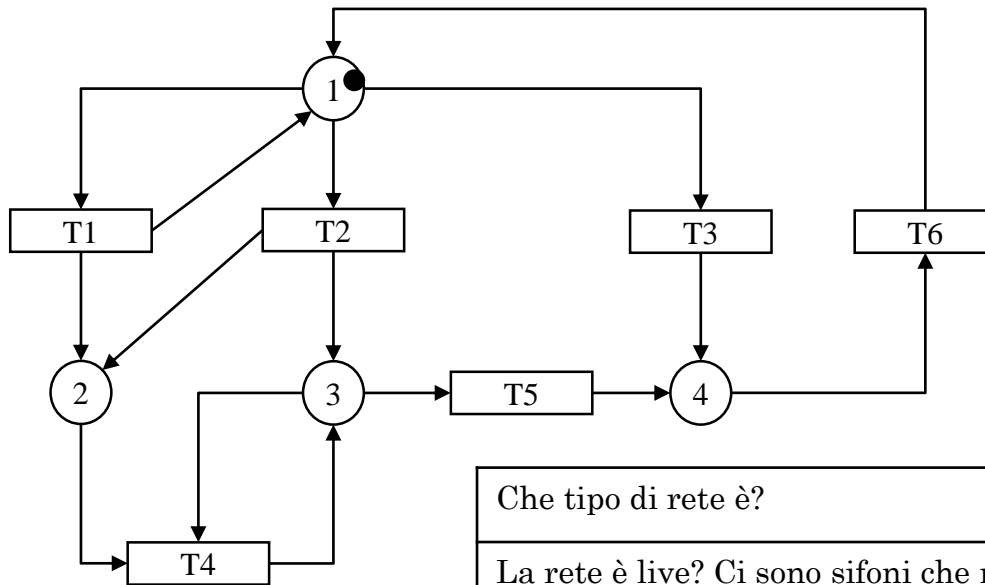
Albero

Domanda – Area, Quesito

Albero

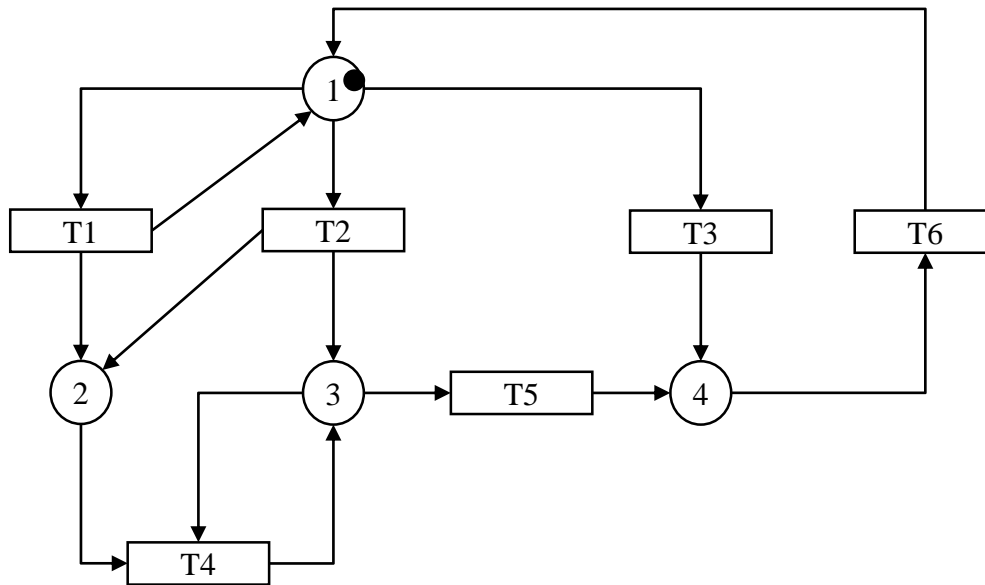
Offerta – Area, Domanda





Si analizzi (senza ridurla) la rete data, che ha un token iniziale in P1, per rispondere alle domande. Si usino sifoni, trappole, marcature e ragionamenti.

Che tipo di rete è?	
La rete è live? Ci sono sifoni che non contengono trappole marcate inizialmente; se sì quali?	
Ci sono sifoni uguali a trappole; se sì, quali?	
La rete ha dei deadlock o no? Se sì con quale marcatura?	
La rete è bounded? Se no in quali posti e perché?	
La rete è safe o no e perché?	
La rete è reversibile o no e perché?	
Nel grafo delle marcature come sono scritte le marcature che si ottengono con uno scatto di transizione da M0?	



Si analizzi (senza ridurla) la rete data, che ha un token iniziale in P1, per rispondere alle domande. Si usino sifoni, trappole, marcature e ragionamenti.

T1 fork, T2 fork, T3 passante, T4 join, T5 passante, T6 passante.

P1 free choice, P3 e P2 formano una AC.

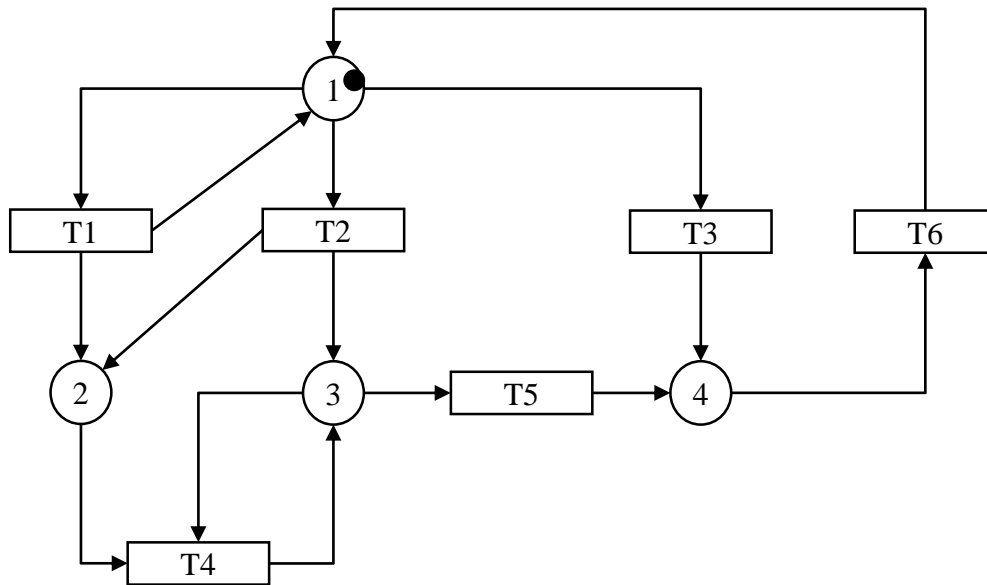
P3 ha T4 come input e output ma non è né un sifone né una trappola; idem per P1 con T1.

Lo scatto ripetuto di T1 accumula token in P2.

I posti P1, P3 e P4 formano una trappola (sifone) contenente 1 solo token; tutte le transizioni sono neutre rispetto al subset.

La transizione join T4 può sempre scattare.

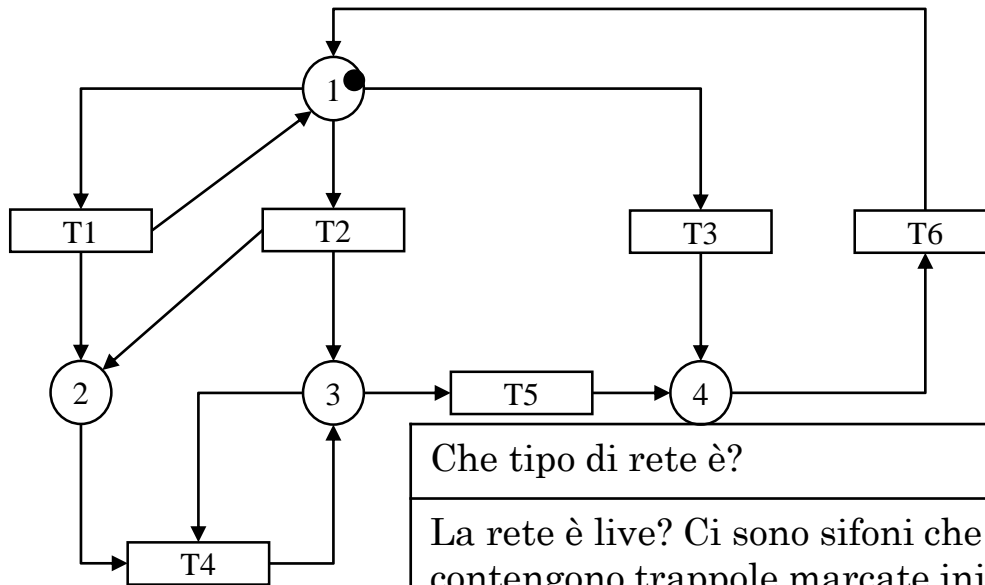
La rete è live e unbounded in P2. P2 può essere svuotato quindi la rete è reversibile.



Sifoni e trappole

[1, 3, 4] [+-, +-, +-, +-, +-, +-, st]

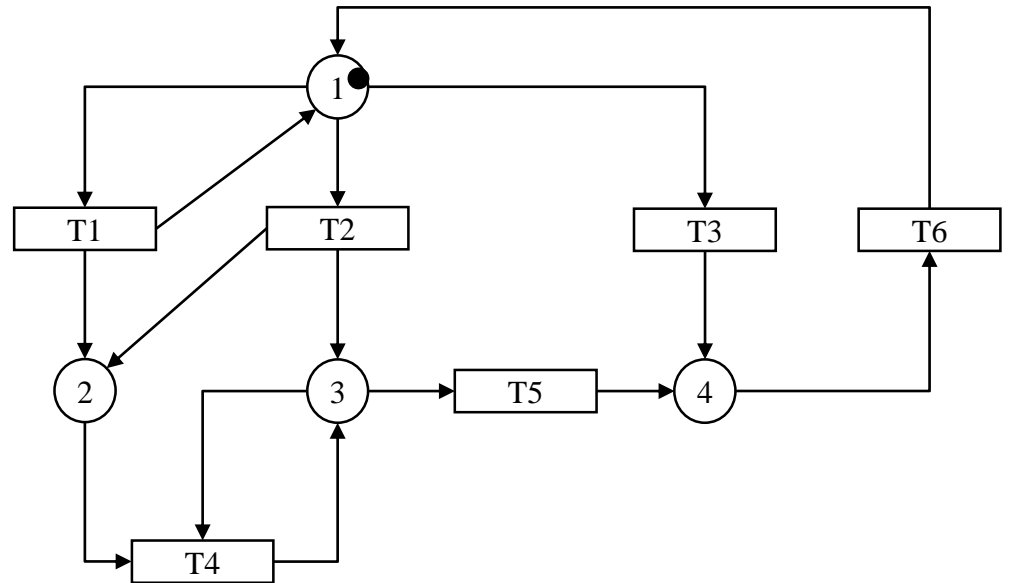
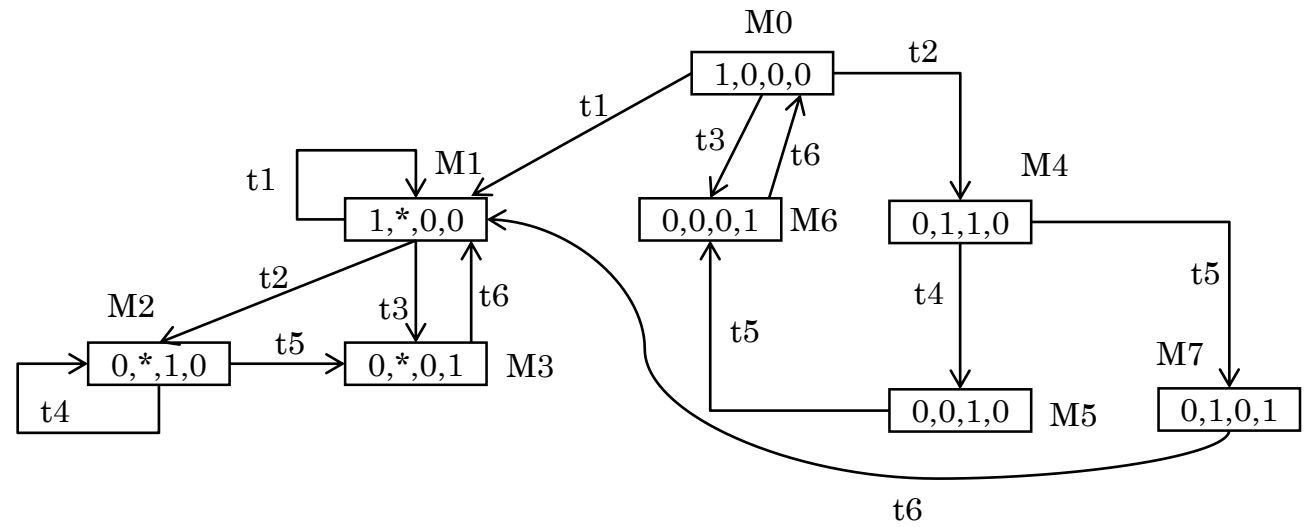
the net is live



subset [1, 3, 4] [+-, +-, +-, +-, +-, +-, st]
the net is live

Che tipo di rete è?	AC (P2,P3)
La rete è live? Ci sono sifoni che non contengono trappole marcate inizialmente; se sì quali?	Sì; no.
Ci sono sifoni uguali a trappole; se sì, quali?	1,3,4.
La rete ha dei deadlock o no? Se sì con quale marcatura?	No, in quanto live.
La rete è bounded? Se no in quali posti e perché?	No, è unbounded in p2.
La rete è safe o no e perché?	No, perché è unbounded.
La rete è reversibile o no e perché?	Sì: t4 può consumare i token posti in p2 da t1 e t2.
Nel grafo delle marcature come sono scritte le marcature che si ottengono con uno scatto di transizione da M0?	T1 (1,ω,0,0) T3 (0,0,0,1) T2 (0,1,1,0)

nota: * rappresenta ω



CG

WBT

```
static int w3 (boolean f, boolean g,  
boolean h, int a, int b, int c) {  
    int x = 0; int y = 0;  
    if (f && g) {                                1  
        if (a > b) y = a;                        2  
        x = a + b;  
    } else {  
        if (c > 100) y = b;                      3  
        x = a + b;  
    }  
    if ((f && g) && h) return 0;                  4  
    else {  
        if (a > 100) y += b;                     5  
        else x += y;  
        return x;  
    }  
}
```

N. min di test per la copertura dei
criteri seguenti; si spieghi il valore.

Nodi:

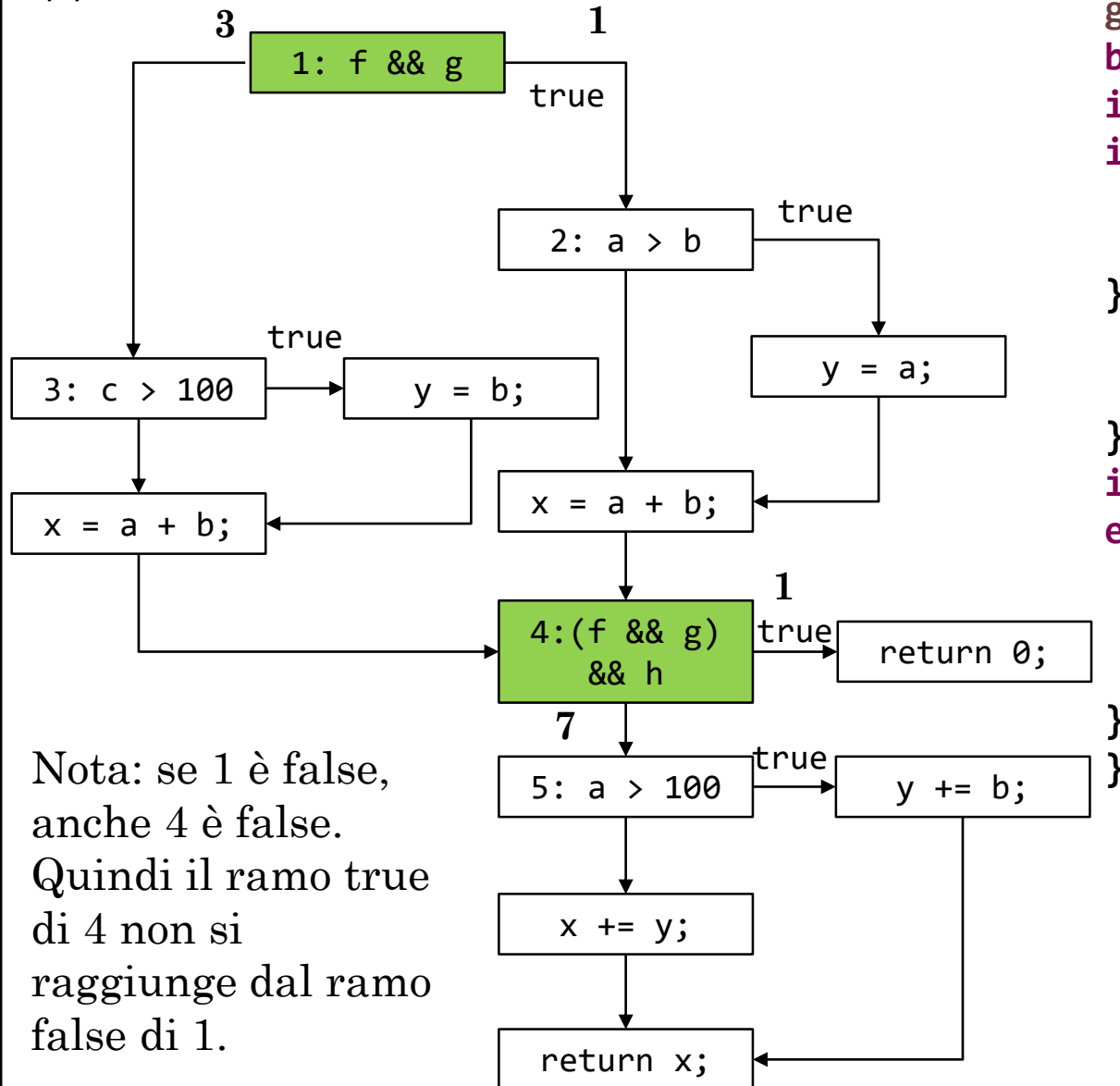
Link (edge):

Percorsi:

Condizioni multiple:

N. min test per tutti i criteri:

WBT



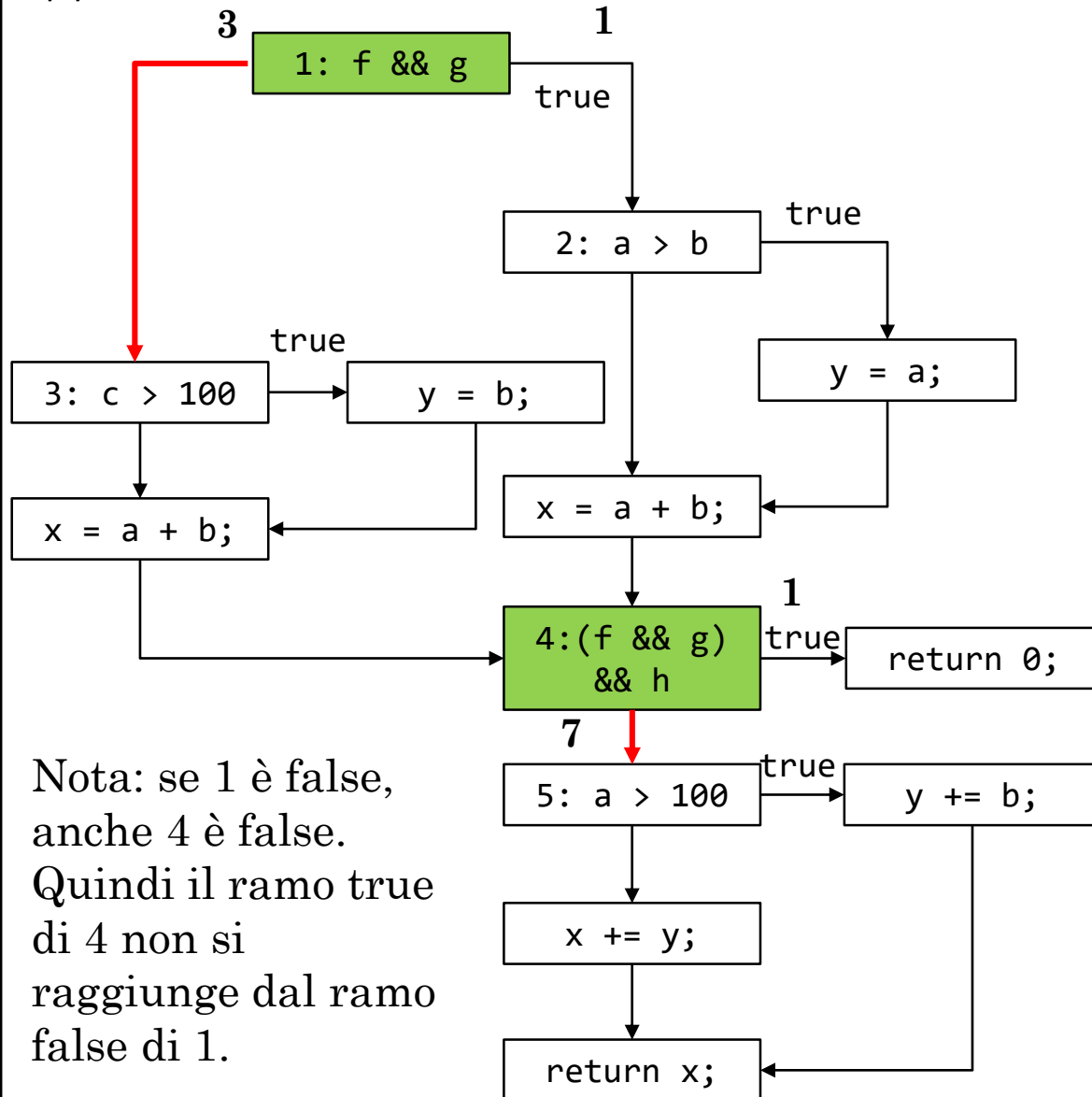
```

static int w3 (boolean f, boolean
g,
boolean h, int a, int b, int c) {
int x = 0; int y = 0;
if (f && g) {                                1
    if (a > b) y = a;                          2
    x = a + b;
} else {
    if (c > 100) y = b;                        3
    x = a + b;
}
if ((f && g) && h) return 0;                  4
else {
    if (a > 100) y += b;                      5
    else x += y;
    return x;
}
}
    
```

Nota: omesso il nodo
iniziale per motivi di
spazio.

```
int x = 0; int y = 0;
```

WBT



N. min di test per la copertura dei criteri seguenti; si spieghi il valore.

Nodi: **3**; 1(2) se la prima condiz. è true, 2 (1) se false; es. 1T 2T 4T, 1F 3T 4F 5T, 1F 3F 4F 5F

Link (edge): **4**; 2 se la prima condiz. è true, 2 se false; oltre ai precedenti 1T 2F 4T

Percorsi: **10**; 6 se 1T, 4 se 1F; 1T 2 4T, 1T 2 4F 5, 1F 3 4F 5

Condizioni multiple: **8**

Gli 8 casi entrano in 1 e raggiungono 4.

N. min test per tutti i criteri: **12**

Degli 8 casi di test delle c. multiple, 2 escono da 1T e 6 da 1F. Per averne 6 in uscita da 1T occorre aggiungerne 4. Quindi 12 in tutto.

Domande

Domanda	Vero	Falso
Nel testing white box il numero minimo di casi di test necessari per la copertura delle condizioni multiple può essere uguale a quello per la copertura dei percorsi.		
Nel BPMN un'attività multi-instance è sempre eseguita sequenzialmente per ogni elemento della collezione di input.		
Dato il class model Department-Employee-Project, l'espressione navigazionale seguente dà i dipartimenti che gestiscono tanti progetti quanti sono i loro impiegati: departments ([projects] == [employees]).		

Risposte

Domanda	Vero	Falso
Nel testing white box il numero minimo di casi di test necessari per la copertura delle condizioni multiple può essere uguale a quello per la copertura dei percorsi.	X	
Nel BPMN un'attività multi-instance è sempre eseguita sequenzialmente per ogni elemento della collezione di input.		X
Dato il class model Department-Employee-Project, l'espressione navigazionale seguente dà i dipartimenti che gestiscono tanti progetti quanti sono i loro impiegati: departments ([projects] == [employees]).	X	