

# *2020*

## *Ingegneria del software:*

### *Soluzione compito 230720*

*Giorgio Bruno*

*Dip. Automatica e Informatica  
Politecnico di Torino  
email: giorgio.bruno@polito.it*

Quest'opera è stata rilasciata sotto la licenza Creative Commons  
Attribuzione-Non commerciale-Non opere derivate 3.0 Unported.  
Per leggere una copia della licenza visita il sito web  
<http://creativecommons.org/licenses/by-nc-nd/3.0/>



## Es. 1 B2B 12 punti

Un'agenzia comunale, avvalendosi di revisori esterni, gestisce le pratiche inviate da studi professionali. Una pratica si riferisce ad un tipo di pratica (TipoP). Nel sistema informativo dell'agenzia sono registrati studi, revisori, tipi di pratiche e tipi di allegati (TipoA). Un tipo di pratica è collegato (n, n) a vari tipi di allegati, i quali sono associati (n, n) a vari revisori competenti. Ogni tipo di pratica è trattato da un funzionario (ruolo di staff).

Ricevuta una pratica, il funzionario idoneo, mediante un *task composto*, genera un allegato preliminare per ogni tipo di allegato relativo al tipo della pratica. Il processo invia allo studio la pratica con gli allegati e lo studio rimanda poi la pratica con gli allegati definiti. Il funzionario assegna ciascun allegato ad un revisore competente (1). I revisori degli allegati sono distinti (1). Il processo invia gli allegati ai revisori.

## *Es. 1    B2B*

I revisori rimandano gli allegati con le interazioni allegato completo (ac) oppure allegato incompleto (ai). Se un allegato è completo, il processo lo colloca nello stadio completo. Se è incompleto, il processo, con l'interazione allegato da modificare (adm), lo manda allo studio che risponde con l'allegato modificato (am). Il processo rimanda l'allegato modificato (con l'interazione am) al revisore che può rispondere con le interazioni ac o ai. Il processo le tratta come indicato prima; si possono quindi avere dei loop.

Un *riduttore*, quando trova nello stadio completo tutti gli allegati di una pratica, li toglie dallo stadio ed emette la pratica; il processo la manda allo studio con l'interazione pratica accettata (pa).

(1) Si indichi il vincolo con un invariante.

Nella risposta si scrivano nello stesso ordine i tre modelli con le tracce seguenti (da copiare e completare).

### **Collaborazioni**

Studio	processo	processo	Revisore
Pratica ->		Allegato ->	

### **Modello informativo + invariante**

Studio	Pratica
--------	---------

### **Processo**

Studio	Funzionario	Revisore
Pratica ->		

*B2B*

# Collaborazioni

Studio    processo

Pratica ref TipoP ->

<- pp, Pratica - Allegato

pd, Pratica - Allegato ->

loop par

alt

    <- adm, Allegato

    am, Allegato ->

---

<- pa, Pratica

break

processo    Revisore

Allegato ->

loop

alt

    <- ac, Allegato

    break

---

    <- ai, Allegato

    am, Allegato ->

ai, ac, am, adm: allegato incompleto, completo, modificato, da modificare

pp, pd, pa: pratica preliminare, definita, accettata

# *Modello informativo*

Funzionario

1|

Studio 1' --- Pratica --- 1' TipoP n --- n TipoA n --- n Revisore

1'|

1'|

1|

-- n Allegato -----

|-----

Invariante

allegato.revisore in allegato.tipoA.revisori.

Altre soluzioni:

revisore.allegati.tipoA in revisore.tipiA.

allegato.tipoA in allegato.revisore.tipiA.

# *Processo*

Studio

Funzionario

Revisore

Pratica ->

allegati per tipiA

genAllegato

pp

<-

pd ->

assegnaRevisori ->

Allegato

adm

<- ai

am ->

am

pa

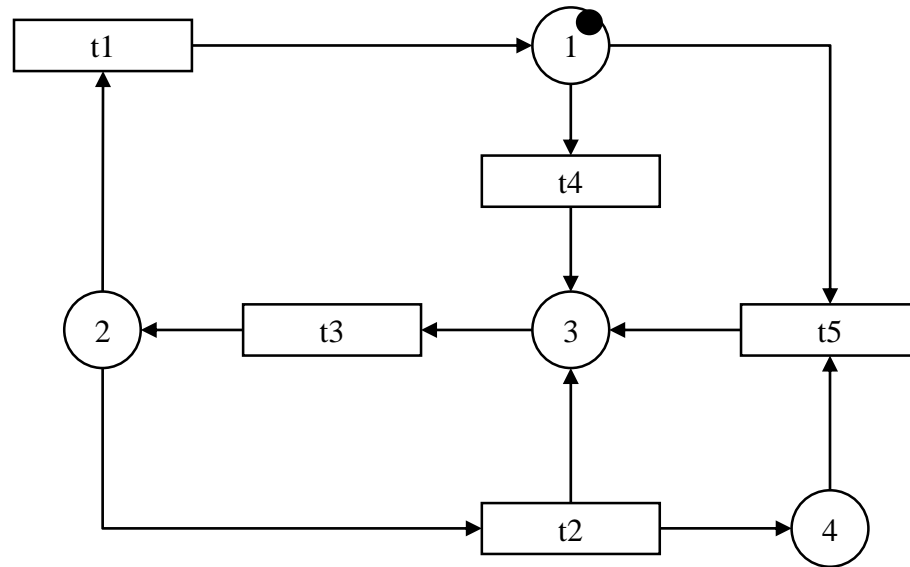
<- (Pratica) R (Allegato)

<- ac

genAllegato: new Allegato.

assegnaRevisori: allegati.revisore def.

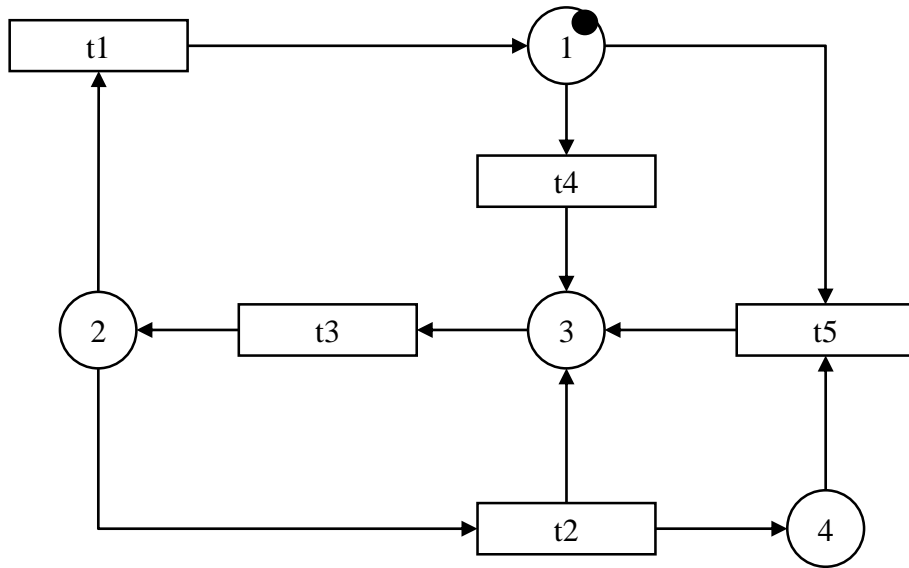
## *Es. 2 PN 8 punti*



Si analizzi (senza modificarla) la rete data, che ha un token iniziale in P1, per rispondere alle domande seguenti.



- 1) Che tipo di rete è?
- 2) La rete è live? Ci sono sifoni che non contengono trappole marcate inizialmente? Se sì quali?
- 3) Ci sono sifoni uguali a trappole? Se sì, quali?
- 4) Ci sono trappole che non sono anche sifoni? Se sì quali?
- 5) La rete è bounded? Se no in quali posti e perché?
- 6) La rete è safe o no e perché?
- 7) La rete è reversibile o no e perché?
- 8) Quali sono le marcature raggiungibili con 3 scatti di transizioni a partire dalla marcatura iniziale?



subset [1, 2, 3]  
[+-, +-, +-, +-, +-, st]  
the net is live

Con lo scatto di t4 il token va in p3. Gli scatti ripetuti di t3 e t2 (fork) accumulano token in p4. Gli scatti ripetuti di t3, t1 e t5 possono eliminare i token in p4 (la rete è reversibile).

- 1) AC in p1,p4.
- 2) Sì, no. (Rete live?)
- 3) Sì, [1, 2, 3]. (Sifoni uguali a trappole?)
- 4) No. (Trappole che non sono sifoni?)
- 5) No, p4. (Rete bounded?)
- 6) No, perché unbounded. (Rete safe?)
- 7) Sì, con gli scatti ripetuti di t5, t3 e t1 si possono eliminare i token in p4. (Rete reversibile?)
- 8) [1, 0, 0, 0] e [0, 0, 1, 1]. (Marcature raggiungibili con 3 scatti di transizioni?)

t1, t3 e t4 passanti; t5 join;  
t2 fork.

Con gli scatti ripetuti di  
t4, t3, t2, t3 e t1 si  
accumulano token in p4.

Si analizzi il metodo seguente e si risponda alle domande.

### *Es. 3 WBT 8 punti*

```
public static int wbt1 (boolean a,  
boolean b, boolean c,  
int x, int y, int z, boolean f,  
boolean g) {  
    int k = 0;  
    if (a && b && c) {                //1  
        if (x > 100) {                //2  
            if (y > 100) k = 10; //3  
        } else k = 30;  
    } else {  
        if (f && g) k = 1000; //4  
        else k = 2000;  
        if (z > k) return 0; //5  
    }  
    if (a && b) return k * 2; //6  
    else return k * 3;  
}
```

1) Qual è il numero minimo di test per la copertura dei nodi?

2) Qual è il numero minimo di test per la copertura dei link?

3) Qual è il numero minimo di test per la copertura dei percorsi?

4) Qual è il numero minimo di test per la copertura delle condizioni multiple?

5) Qual è il numero minimo di test per la copertura sia dei percorsi sia delle condizioni multiple?

Si indichino i test relativi ai nodi, ai link e ai percorsi (domande 1, 2 e 3) con sequenze di condizioni vere o false (ad es. 1T 2F 6T).

Per le altre domande si spieghi il valore.

Nodi: **4**: 2 con 1T e 2 con 1F.  
 1T (2T 3T, 2F) 6T; 1F 4T 5F 6F,  
 1F 4F 5T

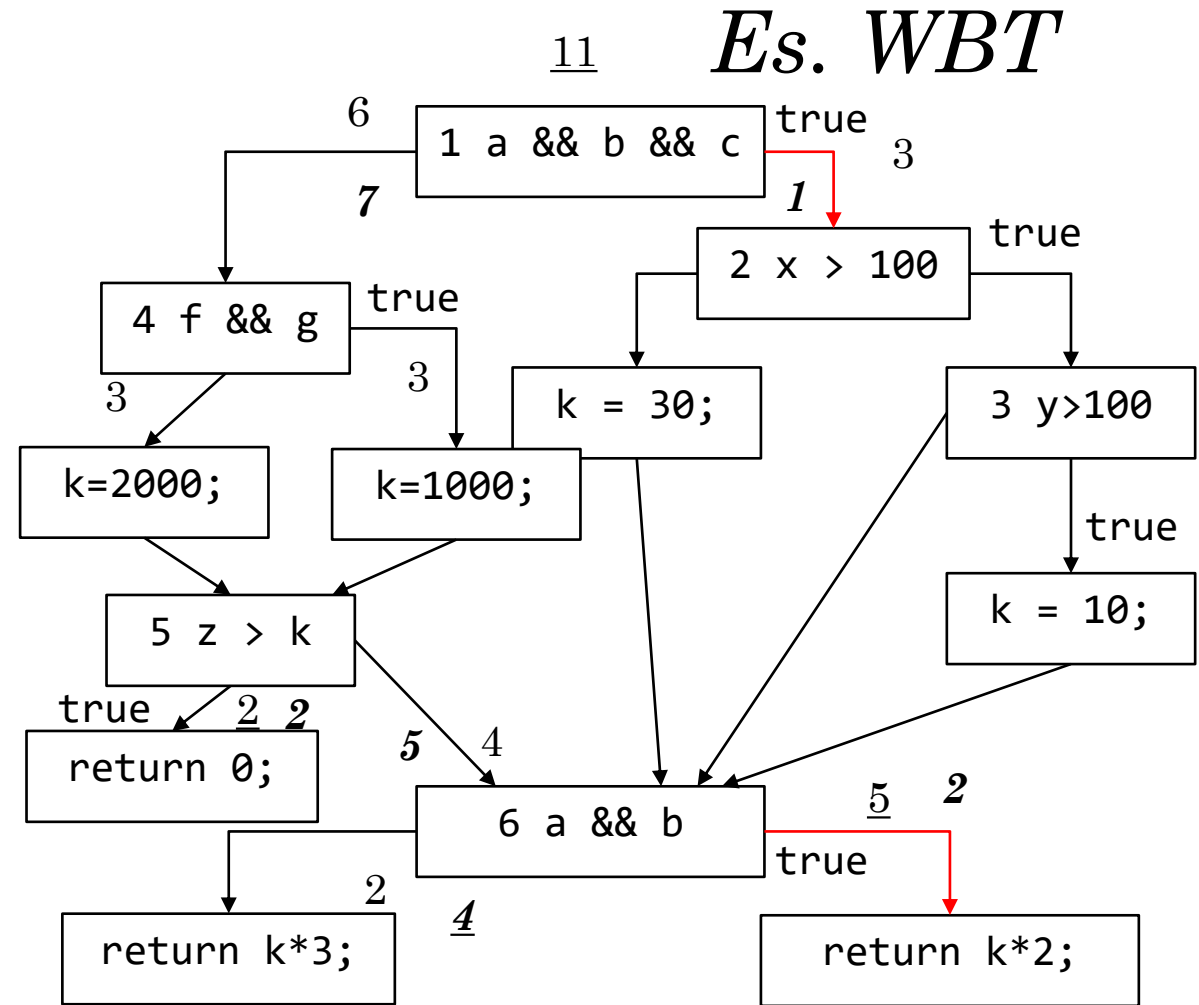
Link: **5**: + 1T 2T 3F 6T

Percorsi: **9**: 3 con 1T e 6 con 1F.  
 1T (2F, 2T 3) 6T. 1F 4 (5T, 5F  
 6).

CM: **8**. I 7 casi con 1F possono  
 soddisfare la doppia 4 e la  
 doppia 6.

Tutti i criteri: **11**.  
 I casi di test delle CM con 6T  
 sono 2 ma ne servono 5 per i  
 percorsi; in totale 11.  
 I percorsi con 6F sono 2 ma ne  
 servono 4 per le CM; in totale  
 11.

Nota: con le CM ci sono due casi  
 con a e b true; essi escono con  
 6T. In totale 5 casi con 6T e 4  
 con 6F.



correlazione: 1T → 6T

Caratteri normali per il n. dei percorsi,  
 corsivo per le CM.

## *Es. 4 Domande 4 punti*

Si risponda alle domande seguenti con Vero o Falso.

1 punto per risposta corretta, -1 per risposta errata, 0 se manca la risposta.

- 1) Le confluenze si trovano nelle reti di tipo State Machine e Grafi Marcati.
- 2) La regola di riduzione che riguarda le transizioni in parallelo si applica quando due o più transizioni hanno un posto di input in comune e un posto di output in comune.
- 3) Il punto di partenza della pipeline di un sistema di continuous integration può essere un ambiente di version control come Git.

## Domande

4) Con il modello informativo seguente, che comprende Dipartimenti, Progetti e Impiegati, relazioni e attributi (slide teoria 1),

Dipartimento 1' ---- n Impiegato

----- Progetto -----

Attributi: Dipartimento: String nome.

```
Impiegato: String nome; String qualifica.
```

l'espressione navigazionale seguente dà la lista dei nomi dei dipartimenti che hanno almeno un progettista tra i loro impiegati.

```
dipartimenti ([impiegati (qualifica == progettista)] > 0).nome;
```

# *Risposte*

1) Le confluenze si trovano nelle reti di tipo State Machine e Grafi Marcati.

Falso. Non ci sono confluenze nei grafi marcati.

2) La regola di riduzione che riguarda le transizioni in parallelo si applica quando due o più transizioni hanno un posto di input in comune e un posto di output in comune.

Falso. Occorre aggiungere che le transizioni non devono avere altri posti di input (output) oltre a quello in comune.

3) Il punto di partenza della pipeline di un sistema di continuous integration può essere un ambiente di version control come Git.

Vero. Slides teoria parte 4.

4) Vero.