

Compito Scritto di Ingegneria del Software

11 marzo 2010

Parte teorica, punti 14

Tempo a disposizione: 1 ora

Esercizio 1

Si descriva il principio di Astrazione.

PUNTI 7

Esercizio 2

Si definiscano e descrivano brevemente i diversi criteri di copertura per il testing "white box".

PUNTI 7

Compito Scritto di Ingegneria del Software

11 marzo 2010

Parte pratica, punti 18

Tempo a disposizione: 1 ora e mezza

Esercizio 3

Si modelli con una rete di Petri un sistema formato da due semafori collocati su due strade che si incontrano in un incrocio. I semafori possono emettere luce rossa, arancio oppure verde. I semafori emettono inizialmente entrambi luce rossa. Possono poi passare alla luce verde, successivamente a quella arancio, a quella rossa e da questa nuovamente alla luce verde. Un semaforo può passare dal rosso al verde solo se l'altro semaforo emette luce rossa. Quando un semaforo passa dall'arancio al rosso abilita l'altro semaforo a passare al verde. L'emissione di luce verde da parte dei due semafori è quindi in mutua esclusione.

PUNTI 5

Esercizio 4

Si dia una specifica in Z di un negozio musicale online. Per poter fare acquisti, gli utenti devono registrarsi specificando il loro indirizzo di posta elettronica e una password. Il sistema memorizza inoltre per ciascun utente il suo credito. Al momento della registrazione, agli utenti pagano con carta di credito 10 euro che va a costituire il credito iniziale dell'utente. Si supponga che l'insieme dei brani nel negozio sia un tipo definito dall'utente

Si modellino in Z le seguenti operazioni:

- 1) Registrazione di un nuovo utente: dato l'indirizzo di posta elettronica, la password e un numero di carta di credito, il sistema aggiunge l'utente al sistema e imposta il credito dell'utente a 10 euro. L'operazione fallisce se l'indirizzo di posta elettronica era già associato ad un utente del sistema
- 2) Acquisto di un brano: l'utente si collega fornendo username, password e il brano da acquistare. Il sistema toglie dal credito dell'utente il costo del brano, 1 euro. L'operazione non va a buon fine se nel sistema non esiste una associazione tra l'utente e la password indicata oppure se il credito è 0

PUNTI 7

Esercizio 5

Si disegni il flusso di esecuzione e si esegua l'analisi di flusso, per mezzo di espressioni regolari, per le variabili del seguente programma.

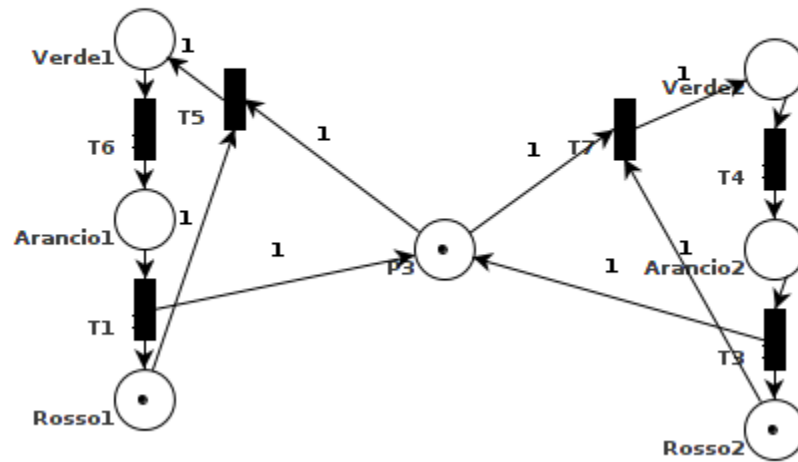
Si individuino inoltre eventuali sequenze non corrette di operazioni e, per ognuna, almeno un caso di test che porti alla sua esecuzione.

```
1. #include<stdio.h>
2. int main (void) {
3.     int a, b = 1, c;
4.     scanf("%d", &a);
5.     if (a > 0) {
6.         a = a - b;
7.         b = b + 1;
8.         c = a / b;
9.     }
10.    else {
11.        b = a / c;
12.    }
13.    b = a - 1;
14.    printf("%d\n", c );
15.    return 0;
16. }
```

PUNTI 6

Soluzione

Esercizio 3



Esercizio 2

Tipi definiti dall'utente:

[Brani, IndirizziPostaElettronica, Password, NumeriCartaDiCredito]

Brani={insieme dei brani del sito}

IndirizziPostaElettronica={insieme degli indirizzi di posta elettronica}

Password={insieme delle password}

NumeriCartaDiCredito={insieme dei numeri di carta di credito}

Variabili che descrivono lo stato del sistema:

- 1) utenti: è una funzione parziale che associa ad un indirizzo di email di un utente la sua password
- 2) credito: è una funzione parziale che associa ad un indirizzo di email il credito dell'utente

Sito

utenti: IndirizziPostaElettronica \Downarrow Password

credito: IndirizziPostaElettronica \Downarrow \mathbb{N}

dom credito = dom utenti

InitSito

Δ Sito

utenti'={}
credito'={}

Success

rep!: Report

rep! = 'Okay'

1) Registrazione

RegistrazioneOK

Δ Sito

indirizzo?: IndirizziPostaElettronica
password?: Password
cartadicredito?: NumeriCartaDiCredito

indirizzo? %o dom utenti

utenti'=utenti \cup {indirizzo? Σ password?}
credito'=credito \cup {indirizzo? Σ 10}

UtenteGiàPresente

\exists Sito

indirizzo?: IndirizziPostaElettronica

indirizzo? \in dom utenti

rep! = 'Utente gia' presente'

Registrazione \vee RegistrazioneOK \leftrightarrow Success

\vee

UtenteGiàPresente

2) acquisto di un brano

AcquistoOK

Δ Sito

indirizzo?: IndirizziPostaElettronica
password?: Password
brano?: Brani

credito(indirizzo?)>0

$(\text{indirizzo?} \sum \text{passwd?}) \in \text{utenti}$
 $\text{credito}' = \text{credito} \oplus \{\text{indirizzo?} \sum (\text{credito}(\text{indirizzo?}) - 1)\}$
 $\text{utenti}' = \text{utenti}$

IndirizzoPasswordSbagliati

$\exists \text{Sito}$
 $\text{indirizzo?} : \text{IndirizziPostaElettronica}$
 $\text{password ?} : \text{Password}$

$(\text{indirizzo?} \sum \text{Passwod?}) \in \text{utenti}$
 $\text{rep!} = \text{'Indirizzo o password sbagliati'}$

CreditoInsufficiente

$\exists \text{Sito}$
 $\text{indirizzo?} : \text{IndirizziPostaElettronica}$

$\text{credito}(\text{indirizzo?}) = 0$
 $\text{rep!} = \text{'Credito insufficiente'}$

Acquisto $\bar{\wedge}$ AcquistoOK \Rightarrow Success

\vee

IndirizzoPasswordSbagliati

\vee

CreditoInsufficiente

Esercizio 5

riga	a	b	c
3	a	ad	a
4	d		
5	u		
6	ud	u	
7		ud	
8	u	u	d
11	u	d	u
13	u	d	
14			u

Espressioni regolari:

per a: **adu (udu+u) u**
per b: **ad (uudu+d) d**
per c: **a (d+u) u**

Osservazioni:

- Nel caso “else” la variabile c viene usata senza essere definita. Tale anomalia può essere evidenziata da qualsiasi test case con $a < 0$.
- Inoltre, sempre nel caso “else” abbiamo una sequenza anomala per la variabile b, che viene definita tre volte di seguito, senza mai essere utilizzata.