

Compito Scritto di Ingegneria del Software

13 luglio 2011

Parte teorica, punti 14

Tempo a disposizione: 1 ora

Si svolga l'esercizio 1 su un foglio e il 2 su un foglio separato

Esercizio 1

Si indichino i vari tipi di manutenzione a cui il software può essere sottoposto.

PUNTI 7

Esercizio 2

Si definiscano i principi di implementazione ed interfaccia di un modulo, descrivendone le relative prassi di buona progettazione.

PUNTI 7

Compito Scritto di Ingegneria del Software

13 luglio 2011

Parte pratica, punti 18

Tempo a disposizione: 1 ora e mezza

Si svolgano gli esercizi 3 e 4 su un foglio e il 5 su un foglio separato

Esercizio 3

Si modelli con una rete di Petri un sistema di controllo di un aeroporto che ha una sola pista di atterraggio. All'arrivo di un aereo, la torre di controllo gli comunica se può atterrare o meno. Se la pista è libera, l'aereo può atterrare, altrimenti deve attendere in volo che la pista si liberi. Quando un aereo atterra, occupa la pista fino al termine dell'atterraggio, che successivamente diventa libera,

PUNTI 5

Esercizio 4

Si dia una specifica in Z dell'iscrizione ad una conferenza composta da un insieme di workshop. Per potersi iscrivere ad una conferenza, gli utenti devono registrarsi specificando il proprio nome, cognome, l'indirizzo di email, una password e l'organizzazione a cui appartengono. Successivamente l'utente indica l'insieme dei workshop a cui intende partecipare.

Si modellino in Z le seguenti operazioni:

- 1) Registrazione di un nuovo utente: dato il nome, il cognome, l'indirizzo di posta elettronica, la password e l'organizzazione, il sistema aggiunge l'utente al sistema. L'operazione fallisce se una persona con lo stesso nome e cognome si era già iscritta oppure se l'indirizzo di posta elettronica era già associato ad una persona
- 2) Indicazione dei workshop: l'utente si collega fornendo nome, cognome e password. Successivamente l'utente indica l'insieme dei workshop a cui intende partecipare. L'operazione fallisce se l'utente non si era ancora registrato, se aveva già indicato i workshop a cui intendeva partecipare o se l'insieme dei workshop indicato dall'utente è vuoto.

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.         while ( b < a ){
7.             a = a - b;
8.             b = b + 1;
9.             c = a / b;
10.        }
11.    }
12.    else {
13.        b = a / c;
14.    }
15.    b = a - 1;
16.    printf("%d\n",  c );
17.    return 0;
18. }
```

PUNTI 6

Soluzione

Esercizio 3



Esercizio 4

Tipi definiti dall'utente:

[Nomi, Cognomi, Email, Password, Organizzazioni, Workshop]

Nomi={insieme dei nomi degli utenti}

Cognomi={insieme dei cognomi degli utenti}

Email={insieme delle email}

Password={insieme delle password}

Organizzazioni={insieme delle organizzazioni}

Workshop={insieme dei workshop della conferenza}

Variabili che descrivono lo stato del sistema:

- 1) email: è una funzione parziale che associa un nome e cognome ad una email
- 2) password: è una funzione parziale che associa un nome e cognome ad una password
- 3) organizzazione: è una funzione parziale che associa un nome e cognome ad una organizzazione;
- 4) workshop: è una funzione parziale che associa un nome e un cognome ad un insieme di workshop

Conferenza

email: $\text{Nomi} \times \text{Cognomi} \rightarrow \text{Email}$
password: $\text{Nomi} \times \text{Cognomi} \rightarrow \text{Password}$
organizzazione: $\text{Nomi} \times \text{Cognomi} \rightarrow \text{Email}$
workshop: $\text{Nomi} \times \text{Cognomi} \rightarrow \mathcal{P} \text{ Workshop}$

$\emptyset \notin \text{ran workshop}$

InitConferenza

$\Delta \text{Conferenza}$

email'={}
password'={}
organizzazione'={}
workshop'={}

Success

rep!: Report

rep! = 'Okay'

1) registrazione:

RegistrazioneOK

Δ Conferenza
nome?: Nomi
cognome?: Cognomi
email?: Email
password?: Password
organizzazione?: Organizzazioni

$(\text{nome?}, \text{cognome?}) \notin \text{dom email}$
 $\text{email}^{-1}(\text{email?}) = \{\}$
 $\text{email}' = \text{email} \cup \{(\text{nome?}, \text{cognome?}) \mapsto \text{email?}\}$
 $\text{password}' = \text{password} \cup \{(\text{nome?}, \text{cognome?}) \mapsto \text{password?}\}$
 $\text{organizzazione}' = \text{organizzazione} \cup \{(\text{nome?}, \text{cognome?}) \mapsto \text{organizzazione?}\}$
 $\text{workshop}' = \text{workshop}$

NomeCognomeGia'Presente

\exists Conferenza
nome?: Nomi
cognome?: Cognomi
rep!: Report

$(\text{nome?}, \text{cognome?}) \in \text{dom email}$
rep! = 'Nome e Cognome gia' presenti'

EmailGia'Presente

\exists Conferenza
email?: Email
rep!: Report

email⁻¹ (email?) ≠ {}
rep! = 'Email gia' presente'

Registrazione \cong RegistrazioneOK \wedge Success

∨

NomeCognomeGia'Presente

∨

EmailGia'Presente

2) indicazione dei workshop

IndicazioneOK

Δ Conferenza

nome?: Nomi

cognome?: Cognomi

workshop?: \mathbb{P} Workshop

(nome?,cognome?) \in dom email

(nome?,cognome?) \notin dom workshop

workshop? ≠ {}

email'=email

password'=password

organizzazione'=organizzazione

workshop'=workshop \cup {(nome?,cognome?) \mapsto workshop?}

NomeCognomeAssente

\exists Conferenza

nome?: Nomi

cognome?: Cognomi

(nome?,cognome?) \notin dom email

rep! = 'Persona assente'

WorkshopGia'Indicati

\exists Conferenza

nome?: Nomi

cognome?: Cognomi

(nome?,cognome?) \in dom workshop

rep! = 'Workshop gia' indicati'

InsiemeWorkshopVuoto

\exists Conferenza

workshop?: \mathbb{P} Workshop

workshop? = {}

rep! = 'Insieme workshop vuoto'

Indicazione \cong IndicazioneOK \wedge Success

\vee

NomeCognomeAssente

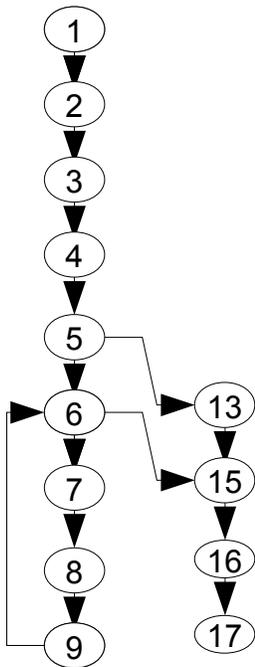
\vee

WorkshopGiàIndicati

\vee

InsiemeWorkshopVuoto

Esercizio 5



Riga	a	b	c
3	a	ad	a
4	d		
5	u	u	
6	ud	u	
7		ud	
8	u	u	d
9			
13	u	d	u
15	u	d	
16			u
17			

Variabile a: $adu (u(uduu)^* + u) u$

Variabile b: $ad (u (uuduu)^* + d) d$

Variabile c: $a ((d)^* + u) u$

La variabile c potrebbe essere usata senza essere definita. Questo accade :

- se si va nel ramo "else", ovvero per $a \leq 0$
- se non si esegue il ciclo while, ovvero per $b \geq a$ (quindi per $a=1$)