

NOME \_\_\_\_\_ COGNOME \_\_\_\_\_ MATRICOLA \_\_\_\_\_

## **Ingegneria del Software II**

**22 Dicembre 2011**

**Parte teoria, punti 14 - Tempo a disposizione: 1h**

### **Esercizio 1 (punti 7)**

Si descrivano le specifiche algebriche e se ne fornisca un esempio.

### **Esercizio 2 (punti 7)**

Si definiscano le basi della teoria del testing: test case, test set, criteri e relative caratteristiche.

## Ingegneria del Software II

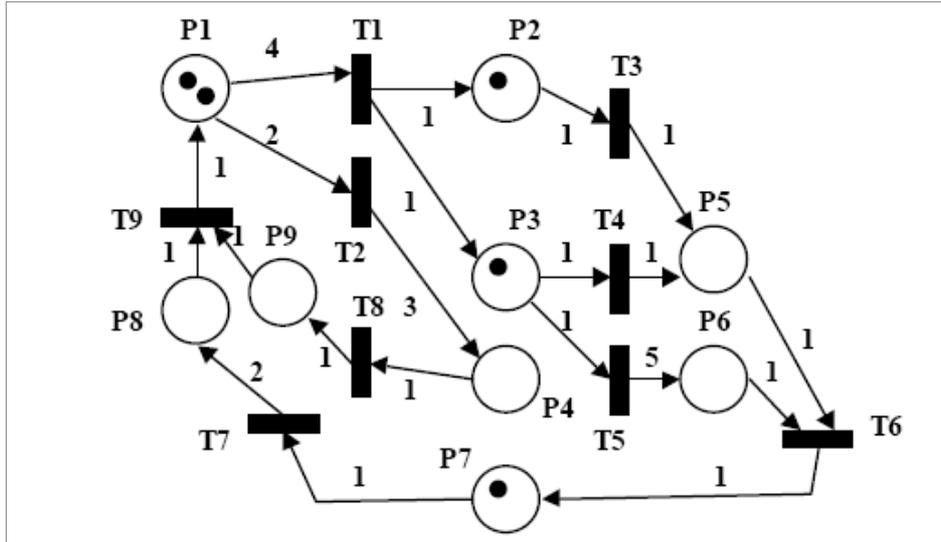
22 Dicembre 2011

### Parte pratica, punti 18 - Tempo a disposizione: 2h

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

#### Esercizio 3 (punti 5)

Sia data la seguente rete di Petri (si notino i pesi non unitari di alcuni degli archi).



- a. Indicare quali sono le transizioni abilitate con la marcatura indicata e le transizioni eventualmente in conflitto.
- b. Dire quali sono le marcature che la rete può raggiungere se scattano tutte le transizioni abilitate. (scrivendo le matrici I e O e il vettore marcatura lo svolgimento è molto semplificato – AA2011-12)

#### Esercizio 4 (punti 7)

Si dia una specifica in Z del sistema di gestione delle iscrizioni ad una associazione, I nuovi soci possono iscriversi specificando il proprio nome, cognome, l'indirizzo di e-mail, una password e specificando il metodo di pagamento, carta di credito o bonifico. I soci che devono rinnovare l'iscrizione devono solo specificare l'indirizzo di e-mail e la password.

Si modellino in Z le seguenti operazioni:

1. Iscrizione di un nuovo socio: dato il nome, il cognome, l'indirizzo di posta elettronica, la password e il metodo di pagamento (carta di credito o bonifico), il sistema memorizza l'associazione tra l'e-mail e la coppia (nome, cognome) e tra l'e-mail e la password. Inoltre Il sistema iscrive il socio per l'anno corrente. L'operazione fallisce se una persona con la stessa e-mail era già registrata nel sistema.
2. Rinnovo dell'iscrizione: l'utente fornisce indirizzo di e-mail, password e metodo di pagamento.. Il sistema controlla se l'utente era socio per l'anno precedente e, se questo e' vero, rinnova l'iscrizione. L'operazione fallisce se l'utente non è registrato nel sistema, se la password non è giusta oppure se non era socio nell'anno precedente.

Lo stato del sistema deve includere il valore dell'anno corrente, inizializzato a 2011. Inoltre, si usino le e-mail per identificare i soci nel sistema.

NOME \_\_\_\_\_ COGNOME \_\_\_\_\_ MATRICOLA \_\_\_\_\_

**Esercizio 5 (punti 6)**

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", b );
17.    return 0;
18. }
```

NOME \_\_\_\_\_ COGNOME \_\_\_\_\_ MATRICOLA \_\_\_\_\_

NOME \_\_\_\_\_ COGNOME \_\_\_\_\_ MATRICOLA \_\_\_\_\_

### Esercizio 3

### Esercizio 4

Tipi definiti dall'utente:

[Nomi, Cognomi, Email, Password, Anni, MetodiDiPagamento]

Nomi = insieme dei nomi delle persone

Cognomi = insieme dei cognomi delle persone

Email = insieme delle email

Password = insieme delle password

Anni = insieme degli anni

MetodoDiPagamento = carta\_di\_credito | bonifico

Variabili che descrivono lo stato del sistema:

19. registrati: è una funzione parziale dalle email alle coppie (nome,cognome)
20. password: è una funzione parziale dalle email alle password
21. iscritti: è una funzione parziale dagli anni ad un insieme di email
22. anno: l'anno corrente

Ass

registrati:  $\text{Email} \rightarrow \text{Nomi} \times \text{Cognomi}$

password:  $\text{Email} \rightarrow \text{Password}$

iscritti:  $\text{Anni} \rightarrow \mathbb{P} \text{Email}$

anno:  $\text{Anni}$

$\forall a: \text{Anni} \cdot \text{iscritti}(a) \supseteq \text{dom password}$

NOME \_\_\_\_\_ COGNOME \_\_\_\_\_ MATRICOLA \_\_\_\_\_

InitAss \_\_\_\_\_

$\Delta$ Ass

anno'=2011

Success \_\_\_\_\_

rep!: Report

rep! = 'Okay'

1) Iscrizione:

IscrizioneOK \_\_\_\_\_

$\Delta$ Ass

nome?: Nomi

cognome?: Cognomi

email?: Email

password?: Password

metodo?: MetodoDiPagamento

email?  $\notin$  dom(registrati)

registrati"=registrati  $\cup$  {email?  $\mapsto$  (nome?,cognome?)}

password"=password  $\cup$  {email?  $\mapsto$  password?}

iscritti"=iscritti  $\oplus$  {anno  $\mapsto$  iscritti(anno)  $\cup$  email?}

anno"=anno

EmailPresente \_\_\_\_\_

$\exists$ Ass

email?: Email

rep!: Report

email?  $\in$  dom(registrati)

rep! = 'Email già presente'

Iscrizione  $\cong$  IscrizioneOK  $\wedge$  Success

$\vee$

EmailPresente

NOME \_\_\_\_\_ COGNOME \_\_\_\_\_ MATRICOLA \_\_\_\_\_

## 2)Rinnovo

### RinnovoOK

$\Delta$ Ass  
email?: Email  
password?: Password  
metodo?: MetodoDiPagamento

email?  $\in$  dom(email)  
registrati'=registrati  
password(email?)=password?  
email?  $\in$  iscritti(anno-1)  
password"=password  
iscritti"=iscritti  $\oplus$  { anno  $\mapsto$  iscritti(anno)  $\cup$  email?}  
anno"=anno

### NonRegistrato

$\exists$ Ass  
email?: Email  
rep!: Report

email?  $\in$  dom(email)  
rep! = 'Utente non registrato'

### PasswordErrata

$\exists$ Ass  
email?: Email  
password?: Password  
rep!: Report

password(email?) $\neq$ password?  
rep! = 'Password errata'

### NonIscritto

$\exists$ Ass  
email?: Email  
rep!: Report

email?  $\notin$  iscritti(anno-1)  
rep! = 'Non iscritto nell'anno precedente'

NOME \_\_\_\_\_ COGNOME \_\_\_\_\_ MATRICOLA \_\_\_\_\_

Rinnovo  $\equiv$  RinnovoOK  $\wedge$  Success

∨

NonRegistrato

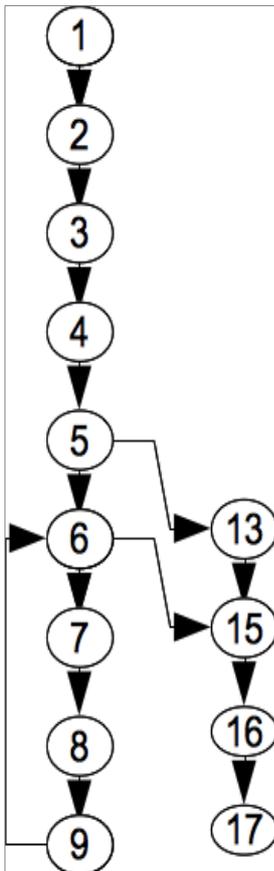
∨

PasswordErrata

∨

NonIscritto

**Esercizio 5**



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

**Variabile a:**  $adu (u(udu)^* + u) u$   
**Variabile b:**  $ad ( u (udu)^* + d) du$   
**Variabile c:**  $a ( ( d )^* + u )$

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  $a=1$ )

In ogni caso, nel ramo “if” la variabile c viene definita senza mai essere usata. Questo accade quando  $a > 0$ .

Infine, la variabile b, se si segue il ramo “else”, ha due definizioni consecutive, senza alcun utilizzo intermedio. Questo accade per  $a \leq 0$ .