

1) Si consideri il seguente requisito:

RQ: Dopo 60 unità di tempo dall'inizio dell'esecuzione vale la seguente proprietà:

se 10 unità di tempo nel passato x era maggiore di 0 allora ora y è negativa.

Tenendo presente che, al tempo time, delay(z, w) ritorna 0 se time <= w e ritorna il valore che z aveva al tempo (time - w), se time = w.

Quale dei seguenti monitor meglio descrive il requisito RQ ?

```
1.class MonitorInputReal x, y; OutputBoolean wy;Boolean wz;initial equationwy = false;equationwz = (time > 60) and (delay(x, 10) > 0) and (y >= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;
```

```
2.class MonitorInputReal x, y; OutputBoolean wy;Boolean wz;initial equationwy = false;equationwz = (time > 60) or (delay(x, 10) > 0) or (y >= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;
```

```
3.class MonitorInputReal x, y; OutputBoolean wy;Boolean wz;initial equationwy = false;equationwz = (time > 60) and (delay(x, 10) <= 0) and (y >= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;
```

Risposta : 1

2) Un processo software può essere rappresentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del processo software e gli archi le transizioni da una fase all'altra. Gli archi sono etichettati con le probabilità della transizione e gli stati sono etichettati con il costo per lasciare lo stato.

Ad esempio lo state diagram in figura

Rappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità dello 0.4 di dover essere ripetuta (a causa di errori) ed F2 ha una probabilità 0.2 di dover essere ripetuta (a causa di errori).

Uno scenario è una sequenza di stati.

Qual'è la probabilità dello scenario: 1, 3, 4? In altri termini, qual'è la probabilità che non sia necessario ripetere la seconda fase (ma non la prima) ?

Immagine

1.0.12

2.0.08

3.0.32

Risposta : 1

3) La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

Test case 1: act0 act2 act1 act2 act2 act2 act0 act1 act2 act2 act2

Test case 2: act0 act1 act2 act2 act1 act2 act0 act2 act2 act2 act0

Test case 3: act2 act2 act0 act2 act1 act0 act2 act0

Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?

Immagine

1.100%

2.80%

3.60%

Risposta : 2

4) Il rischio R può essere calcolato come $R = P \cdot C$, dove P è la probabilità dell'evento avverso (software failure nel nostro contesto) e C è il costo dell'occorrenza dell'evento avverso.

Assumiamo che la probabilità P sia legata al costo di sviluppo S dalla formula

$P = 10^{-b \cdot S}$ (cioè 10 elevato alla (-b*S))

dove b è una opportuna costante nota da dati storici aziendali. Si assuma che $b = 0.0001$, $C = 1000000$, ed il rischio ammesso è $R = 1000$. Quale dei seguenti valori meglio approssima il costo S per lo sviluppo del software in questione.

1.700000 EUR

2.300000 EUR

3.500000 EUR

Risposta : 2

5) La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura

Si consideri il seguente insieme di test cases:

Test case 1: act1 act0 act2 act0

Test case 2: act0 act1 act2 act2 act0

Test case 3: act0 act0 act0

Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra

Immagine

- 1.State coverage: 80%
- 2.State coverage: 75%
- 3.State coverage: 100%

Risposta : 2

6) La state coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura

Si consideri il seguente insieme di test cases:

Test case 1: act1 act1 act0 act1 act1 act2 act0

Test case 2: act2 act0 act0

Test case 3: act1 act1 act2 act0 act0

Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra

Immagine

- 1.State coverage: 90%
- 2.State coverage: 70%
- 3.State coverage: 100%

Risposta : 1

7) La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura

Si consideri il seguente insieme di test cases:

Test case 1: act2 act2 act2 act1 act2 act1 act2 act2 act1

Test case 2: act2 act2 act0 act1 act1 act2 act0 act0 act2 act0 act2 act2 act2 act0 act0 act0 act2 act2 act0 act2 act2 act2 act1 act2 act2 act1

Test case 3: act2 act0 act2 act1 act2 act1 act0 act2 act2 act0 act0 act2 act1

Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra

Immagine

- 1.State coverage: 90%
- 2.State coverage: 60%
- 3.State coverage: 70%

Risposta : 1

8) Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il tempo necessario per completare la fase x è $\text{time}(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta il completamento del processo software, e quindi $\text{time}(1) = 0$.

Il tempo di una istanza del processo software descritto sopra è la somma dei tempi degli stati (fasi) attraversati (tenendo presente che si parte sempre dallo stato 0.

Quindi il costo $\text{Time}(X)$ della sequenza di stati $X = x(0), x(1), x(2), \dots$ è $\text{Time}(X) = \text{time}(x(0)) + \text{time}(x(1)) + \text{time}(x(2)) + \dots$

Ad esempio se $X = 0, 1$ abbiamo $\text{Time}(X) = \text{time}(0) + \text{time}(1) = \text{time}(0)$ (poichè $\text{time}(1) = 0$).

Quale delle seguenti formule calcola il valore atteso del costo per completare il processo software di cui sopra

Immagine

1. $\text{time}(0) \cdot (1 - p)/p$
2. $\text{time}(0)/(1 - p)$
3. $\text{time}(0)/(p \cdot (1 - p))$

Risposta : 2

9) Il partition coverage di un insieme di test cases è la percentuale di elementi della partition inclusi nei test cases. La partition è una partizione finita dell'insieme di input della funzione che si sta testando.

Si consideri la seguente funzione C:

```
int f1(int x) { return (x + 7); }
```

Si vuole testare la funzione $f1()$. A tal fine l'insieme degli interi viene partizionato come segue:

$\{-inf, -101\}, [-100, -1], \{0\}, [1, 500], [501, +inf\}$

Quale dei seguenti test cases consegue una partition coverage del 100% ?

1. $\{x = -200, x = -50, x = 0, x = 100, x = 500\}$

2.{x = -200, x = -150, x = 0, x = 100, x = 700}

3.{x = -200, x = -50, x = 0, x = 100, x = 700}

Risposta : 3

10) La state coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura

Si consideri il seguente insieme di test cases:

Test case 1: act1 act2 act2 act2 act2 act0 act2

Test case 2: act2 act0 act0 act2 act0

Test case 3: act2 act2 act0 act2 act2 act0

Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra

Immagine

1.State coverage: 100%

2.State coverage: 90%

3.State coverage: 80%

Risposta : 1

11) Si consideri il seguente requisito:

RQ: Dopo 10 unità di tempo dall'inizio dell'esecuzione vale la seguente proprietà: se la variabile x è nell'intervallo [10, 20] allora la variabile y è compresa tra il 50% di x ed il 70% di x.

Quale dei seguenti monitor meglio descrive il requisito RQ ?

1.class MonitorInputReal x, y; // plant outputOutputBoolean wy;Boolean wz;initial equationwy = false;equationwz = (time > 10) and (x >= 10) and (x <= 20) and (y >= 0.5*x) and (y <= 0.7*x)

;algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

2.class MonitorInputReal x, y; // plant outputOutputBoolean wy;Boolean wz;initial equationwy = false;equationwz = (time > 10) and ((x < 10) or (x > 20)) and ((y < 0.5*x) or (y > 0.7*x))

;algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

3.class MonitorInputReal x, y; // plant outputOutputBoolean wy;Boolean wz;initial equationwy = false;equationwz = (time > 10) and (x >= 10) and (x <= 20) and ((y < 0.5*x) or (y > 0.7*x))

;algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 3

12) La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

Test case 1: act0 act2 act1 act2

Test case 2: act2 act2 act1 act2 act2

Test case 3: act2 act1 act0 act2

Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?

Immagine

1.Transition coverage: 80%

2.Transition coverage: 100%

3.Transition coverage: 50%

Risposta : 3

13) Il team di sviluppo di un azienda consiste di un senior software engineer e due sviluppatori junior. Usando un approccio agile, ogni iterazione impegna tutti e tre i membri del team per un mese ed occorrono tre iterazioni per completare lo sviluppo. Si assuma che non ci siano "change requests" e che il membro senior costi A Eur/mese ed i membri junior B Eur/mese.

Qual'e' il costo dello sviluppo usando un approccio agile ?

1.A + 2*B

2.3*A + 2*B

3.3*(A + 2*B)

Risposta : 3

14) Una azienda finanziaria desidera costruire un sistema software per ottimizzare i processi di business. Quali delle seguenti attività può contribuire a validare i requisiti del sistema ?

1.Costruire un modello di simulazione per i principali aspetti dei processi di business dell'azienda e per il sistema software da realizzare e valutare le miglorie apportate dal sistema software ai processi di business dell'azienda mediante simulazione.

2.Costruire un prototipo del sistema e testarlo rispetto ai requisiti funzionali usando i dati storici dall'azienda.

3.Costruire un prototipo del sistema e valutarne i requisiti non funzionali usando i dati storici dall'azienda.

Risposta : 1

15) Il branch coverage di un insieme di test cases è la percentuale di branch del programma che sono attraversati da almeno un test case.

Si consideri la seguente funzione C:

```
-----  
  
int f(int x, int y) {  
  
    if (x - y <= 0) { if (x + y >= 1) return (1); else return (2); }  
  
    else {if (2*x + y >= 5) return (3); else return (4); }  
  
} /* f() */
```

Quale dei seguenti test sets consegue una branch coverage del 100% ?

1.Test set: {x=1, y=1}, {x=2, y=2}, {x=2, y=1}, {x=2, y=0}.

2.Test set: {x=1, y=1}, {x=0, y=0}, {x=2, y=1}, {x=2, y=0}.

3.Test set: {x=1, y=1}, {x=0, y=0}, {x=2, y=1}, {x=2, y=3}.

Risposta : 2

16) Si consideri la Markov chain in figura con stato iniziale 0 e p in (0, 1). Quale delle seguenti formule calcola il valore atteso del numero di transizioni necessarie per lasciare lo stato 0.

Immagine

1. $1/(p*(1 - p))$

2. $1/(1 - p)$

3. $(1 - p)/p$

Risposta : 2

17) La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

Test case 1: act2 act0 act2 act2 act2

Test case 2: act0 act2 act2 act1 act2 act1 act1 act1 act2 act2 act2 act2

Test case 3: act2 act2 act2 act0 act1 act0

Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?

Immagine

1.Transition coverage: 60%

2.Transition coverage: 70%

3.Transition coverage: 100%

Risposta : 2

18) Si consideri il seguente requisito:

RQ: Dopo 50 unità di tempo dall'inizio dell'esecuzione vale la seguente proprietà:

se la variabile x è minore del 60% della variabile y allora la somma di x ed y è maggiore del 30% della variabile z

Quale dei seguenti monitor meglio descrive il requisito RQ ?

1.class MonitorInputReal x, y, z; // plant outputOutputBoolean wy;Boolean wz;initial equationwy = false;equationwz = (time > 50) and (x >= 0.6*y) and (x + y <= 0.3*z);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

2.class MonitorInputReal x, y, z; // plant outputOutputBoolean wy;Boolean wz;initial equationwy = false;equationwz = (time > 50) and (x < 0.6*y) and (x + y > 0.3*z);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

3.class MonitorInputReal x, y, z; // plant outputOutputBoolean wy;Boolean wz;initial equationwy = false;equationwz = (time > 50) and (x < 0.6*y) and (x + y <= 0.3*z);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 3

19) Si consideri il seguente requisito:

RQ1: Dopo 20 unità di tempo dall'inizio dell'esecuzione la variabile x è sempre nell'intervallo [20, 30] .

Quale dei seguenti monitor meglio descrive il requisito RQ1 ?

1.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 20) and ((x >= 30) or (x <= 20)) ;algorithmwhen edge(z) theny := true;end when;end Monitor;

2.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 20) and (x >= 20) and (x <= 30) ;algorithmwhen edge(z) theny := true;end when;end Monitor;

```
3.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 20) or ((x >= 20) and (x <= 30)) ;algorithmwhen edge(z) theny := true;end when;end Monitor;
```

Risposta : 1

20) Una Condition è una proposizione booleana, cioè una espressione con valore booleano che non può essere decomposta in espressioni boolean più semplici. Ad esempio, $(x + y \leq 3)$ è una condition.

Una Decision è una espressione booleana composta da conditions e zero o più operatori booleani. Ad esempio, sono decisions:

```
(x + y <= 3)
((x + y <= 3) || (x - y > 7))
```

Un insieme di test cases T soddisfa il criterio di Condition/Decision coverage se tutte le seguenti condizioni sono soddisfatte:

- 1) Ciascun punto di entrata ed uscita nel programma è eseguito in almeno un test;
- 2) Per ogni decision d nel programma, per ogni condition c in d, esiste un test in T in cui c è true ed un test in T in cui c è false.
- 3) Per ogni decision d nel programma, esiste un test in T in cui d è true ed un test in T in cui d è false.

Si consideri la seguente funzione:

```
int f(int a, int b, int c)
{
    if ( (a + b >= 6) && (b - c <= 1) )
        return (1); // punto di uscita 1
    else if ((b - c <= 1) || (b + c >= 5))
        then return (2); // punto di uscita 2
    else return (3); // punto di uscita 3
}
```

Quale dei seguenti test set soddisfa il criterio della Condition/Decision coverage ?

- 1.(a = 6, b = 0, c = 1), (a = 0, b = 5, c = 0), (a = 0, b = 3, c = 2).
- 2.(a = 6, b = 0, c = 1), (a = 0, b = 5, c = 0), (a = 0, b = 3, c = 0).
- 3.(a = 5, b = 0, c = 1), (a = 0, b = 5, c = 0), (a = 0, b = 3, c = 0).

21) Un test oracle per un programma P è una funzione booleana che ha come inputs gli inputs ed outputs di P e ritorna true se e solo se il valore di output di P (con i dati inputs) è quello atteso dalle specifiche.

Si consideri la seguente specifica funzionale per la funzione f.

La funzione $f(\text{int } *A, \text{int } *B)$ prende come input un vettore A di dimensione n ritorna come output un vettore B ottenuto ordinando gli elementi di A in ordine crescente.

Quale delle seguenti funzioni è un test oracle per la funzione f ?

- 1.#define n 1000int TestOracle1(int *A, int *B){int i, j, D[n];//initfor (i = 0; i < n; i++) D[i] = -1;// B is orderedfor (i = 0; i < n; i++) { for (j = i+1; j < n; j++) {if (B[j] < B[i]) {retun (0);}}// B is a permutation of Afor (i = 0; i < n; i++) { for (j = 0; j < n; j++) {if ((A[i] == B[j]) && (D[j] == -1)) {C[i][j] = 1; D[j] = 1; break;}}for (i = 0; i < n; i++) {if (D[i] == -1) return (0);} B okreturn (1);}
- 2.#define n 1000int TestOracle3(int *A, int *B){int i, j, D[n];//initfor (i = 0; i < n; i++) D[i] = -1;// B is orderedfor (i = 0; i < n; i++) { for (j = i+1; j < n; j++) {if (B[j] < B[i]) {retun (0);}}// B is a permutation of Afor (i = 0; i < n; i++) { for (j = 0; j < n; j++) {if (A[i] == B[j]) {C[i][j] = 1; D[j] = 1; break;}}for (i = 0; i < n; i++) {if (D[i] == -1) return (0);} B okreturn (1);}
- 3.#define n 1000int TestOracle2(int *A, int *B){int i, j, D[n];//initfor (i = 0; i < n; i++) D[i] = -1;// B is orderedfor (i = 0; i < n; i++) { for (j = i+1; j < n; j++) {if (B[j] < B[i]) {retun (0);}}// B is a permutation of Afor (i = 0; i < n; i++) { for (j = 0; j < n; j++) {if ((A[i] == B[j]) && (D[j] == -1)) {C[i][j] = 1; break;}}for (i = 0; i < n; i++) {if (D[i] == -1) return (0);} B okreturn (1);}

Risposta : 1

22) Il branch coverage di un insieme di test cases è la percentuale di branch del programma che sono attraversati da almeno un test case.

Si consideri la seguente funzione C:

```
-----
int f(int x, int y) {
    if (x - y <= 0) { if (x + y >= 2) return (1); else return (2); }
    else {if (2*x + y >= 1) return (3); else return (4); }
} /* f() */
```

Si considerino i seguenti test cases: {x=1, y=1}, {x=0, y=0}, {x=1, y=0}, {x=0, y=-1}.

Quale delle seguenti è la branch coverage conseguita?

- 1.100%
- 2.50%
- 3.80%

Risposta : 1

23) La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura

Si consideri il seguente insieme di test cases:

Test case 1: act0 act1 act2 act2 act2 act1 act1 act0 act0 act0 act0 act0 act1

Test case 2: act1

Test case 3: act0 act1 act2 act0 act2 act2 act2 act2 act0 act1

Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra

Immagine

1.State coverage: 60%

2.State coverage: 75%

3.State coverage: 100%

Risposta : 2

24) Il branch coverage di un insieme di test cases è la percentuale di branch del programma che sono attraversati da almeno un test case.

Si consideri la seguente funzione C:

```
-----  
  
int f(int x, int y) {  
    if (x - y <= 6) { if (x + y >= 3) return (1); else return (2); }  
    else {if (x + 2*y >= 15) return (3); else return (4); }  
} /* f() */
```

Quale dei seguenti test sets consegue una branch coverage del 100% ?

1.Test set: {x=3, y=6}, {x=2, y=1}, {x=15, y=0}, {x=9, y=0}.

2.Test set: {x=3, y=6}, {x=0, y=0}, {x=15, y=0}, {x=10, y=3}.

3.Test set: {x=3, y=6}, {x=0, y=0}, {x=15, y=0}, {x=9, y=0}.

Risposta : 3

25) Focalizzandosi sui metodi agile di sviluppo del software, quale delle seguenti affermazioni è vera?

1.Per evitare di sprecare tempo durante la fase di sviluppo del software, il customer non è mai coinvolto nel processo di sviluppo del software.

2.Per evitare di sprecare tempo durante la fase di sviluppo del software, questa inizia solo quando i requisiti sono stati completamente definiti.

3.Le attività di definizione dei requisiti e di sviluppo sono interleaved.

Risposta : 3

26) La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura

Si consideri il seguente insieme di test cases:

Test case 1: act0 act0 act2 act0 act0 act0 act2 act1 act2 act0 act2 act2 act2 act2 act0 act0 act1 act2 act2 act0 act2 act0 act2 act1 act0 act2 act1 act2 act2 act0 act2

Test case 2: act2 act2 act1 act0 act0 act0 act0 act2 act2 act1 act2

Test case 3: act2 act2 act2 act1 act0 act2 act2 act0 act2

Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra

Immagine

1.State coverage: 60%

2.State coverage: 100%

3.State coverage: 80%

Risposta : 2

27) La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

Test case 1: act2 act2 act2 act0

Test case 2: act0 act1 act2 act0 act2

Test case 3: act2 act2 act2 act2

Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?

Immagine

1.Transition coverage: 70%

2.Transition coverage: 100%

3.Transition coverage: 40%

Risposta : 3

28) Una Condition è una proposizione booleana, cioè una espressione con valore booleano che non può essere decomposta in espressioni booleane più semplici. Ad esempio, $(x + y \leq 3)$ è una condition.

Una Decision è una espressione booleana composta da conditions e zero o più operatori booleani. Ad esempio, sono decisions:

$(x + y \leq 3)$

$((x + y \leq 3) \parallel (x - y > 7))$

Un insieme di test cases T soddisfa il criterio di Condition/Decision coverage se tutte le seguenti condizioni sono soddisfatte:

- 1) Ciascun punto di entrata ed uscita nel programma è eseguito in almeno un test;
- 2) Per ogni decision d nel programma, per ogni condition c in d, esiste un test in T in cui c è true ed un test in T in cui c è false.
- 3) Per ogni decision d nel programma, esiste un test in T in cui d è true ed un test in T in cui d è false.

Si consideri la seguente funzione:

```
int f(int a, int b, int c)
{
    if ( (a >= 100) && (b - c <= 1) )
        return (1); // punto di uscita 1
    else if ((b - c <= 1) || (b + c >= 5))
        return (2); // punto di uscita 2
    else return (3); // punto di uscita 3
}
```

Quale dei seguenti test set soddisfa il criterio della Condition/Decision coverage ?

- 1.(a=200, b = 0, c = 1), (a=50, b = 4, c = 0), (a=200, b = 4, c = 0)
- 2.(a=200, b = 0, c = 1), (a=50, b = 5, c = 0), (a=50, b = 0, c = 5).
- 3.(a=200, b = 0, c = 1), (a=50, b = 5, c = 0), (a=50, b = 3, c = 0).

Risposta : 3

29) Il team di sviluppo di un'azienda consiste di un senior software engineer e due sviluppatori junior. Usando un approccio plan-driven (ad esempio, water-fall) la fase di design impegna solo il membro senior per tre mesi e la fase di sviluppo e testing solo i due membri junior per tre mesi. Si assuma che non ci siano "change requests" e che il membro senior costi A Eur/mese ed i membri junior B Eur/mese. Qual'è il costo dello sviluppo usando un approccio plan-driven come sopra ?

- 1.3*A + 6*B
- 2.3*A + 3*B
- 3.A + 2*B

Risposta : 1

30) Un processo software può essere rappresentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del processo software e gli archi le transizioni da una fase all'altra. Gli archi sono etichettati con le probabilità della transizione e gli stati sono etichettati con il costo per lasciare lo stato.

Ad esempio lo state diagram in figura

Rappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità dello 0.3 di dover essere ripetuta (a causa di errori) ed F2 ha una probabilità 0.1 di dover essere ripetuta (a causa di errori).

Uno scenario è una sequenza di stati.

Qual'è la probabilità dello scenario: 1, 2, 3, 4 ? In altri termini, qual'è la probabilità che sia necessario ripetere sia la fase 1 che la fase 2 ?

Immagine

- 1.0.27
- 2.0.03
- 3.0.07

Risposta : 2

31) Il partition coverage di un insieme di test cases è la percentuale di elementi della partition inclusi nei test cases. La partition è una partizione finita dell'insieme di input della funzione che si sta testando.

Si consideri la seguente funzione C:

```
int f1(int x) { return (2*x); }
```

Si vuole testare la funzione f1(). A tal fine l'insieme degli interi viene partizionato come segue:

$\{(-\infty, -11], [-10, -1], \{0\}, [1, 50], [51, +\infty)\}$

Si consideri il seguente insieme di test cases:

$\{x=-100, x=40, x=100\}$

Quale delle seguenti è la partition coverage conseguita?

1.60%

2.100%

3.80%

Risposta : 1

32) Una azienda vende software utilizzando un contratto di Service Level Agreement (SLA) per cui l'utente paga 1000 Eur al mese di licenza e l'azienda garantisce che il software sia "up and running". Questo vuol dire che failures del software generano un costo (quello del repair). Sia $C = 10000$ Eur il costo del repair di una failure e $R = P \cdot C$ il valore atteso (rischio) del costo dovuto alle failures (dove P è la probabilità di una software failure). Ovviamente affinché il business sia profittevole deve essere che R sia al più 1000 Eur. Qual'è il valore massimo di P che garantisce la validità del modello di business di cui sopra ?

1. $P = 1/10$

2. $P = 1/10000$

3. $P = 1/1000$

Risposta : 1

33) Un test oracle per un programma P è una funzione booleana che ha come inputs gli inputs ed outputs di P e ritorna true se e solo se il valore di output di P (con i dati inputs) è quello atteso dalle specifiche.

Si consideri la seguente funzione C :

```
int f(int x, int y) {  
    int z = x;  
    while ( (x <= z) && (z <= y) ) { z = z + 1; }  
    return (z);  
}
```

Siano x, y , gli inputs del programma (f nel nostro caso) e z l'output. Assumendo il programma corretto, quale delle seguenti funzioni booleane $F(x, y, z)$ è un test oracle per la funzione f .

1. $F(x, y, z) = (z == y + 1)$

2. $F(x, y, z) = \text{if } (x > y) \text{ then } (z == x + 1) \text{ else } (z == y + 1)$

3. $F(x, y, z) = \text{if } (x > y) \text{ then } (z == x) \text{ else } (z == y + 1)$

Risposta : 3

34) La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

Test case 1: act2 act2

Test case 2: act0 act1 act1 act1 act2 act2 act1 act0 act1

Test case 3: act0 act0

Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?

Immagine

1.90%

2.70%

3.50%

Risposta : 3

35) Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il costo dello stato (fase) x è $c(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappresenta il completamento del processo software, e quindi $c(1) = 0$.

Il costo di una istanza del processo software descritto sopra è la somma dei costi degli stati attraversati (tenendo presente che si parte sempre dallo stato 0).

Quindi il costo $C(X)$ della sequenza di stati $X = x(0), x(1), x(2), \dots$ è $C(X) = c(x(0)) + c(x(1)) + c(x(2)) + \dots$

Ad esempio se $X = 0, 1$ abbiamo $C(X) = c(0) + c(1) = c(0)$ (poichè $c(1) = 0$).

Quale delle seguenti formule calcola il valore atteso del costo per completare il processo software di cui sopra

Immagine

1. $c(0)^*(1 - p)/p$

2. $c(0)/(1 - p)$

3. $c(0)/(p*(1 - p))$

Risposta : 2

36) La state coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura

Si consideri il seguente insieme di test cases:

Test case 1: act2 act1 act2 act2 act1 act0 act1 act2 act2

Test case 2: act0 act0 act2

Test case 3: act2 act0 act2

Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra

Immagine

1.State coverage: 90%

2.State coverage: 100%

3.State coverage: 70%

Risposta : 1

37) La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

Test case 1: act1 act1 act1

Test case 2: act1 act2 act1 act1 act0 act0 act0 act1 act2 act1 act2 act1 act2 act2 act0 act2 act0 act1 act2 act2 act0 act2 act2 act2

Test case 3: act0 act1 act1 act0 act2 act2 act0 act2 act0 act2 act0 act2 act0 act0 act0 act0 act0 act1 act1 act2

Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?

Immagine

1.100%

2.60%

3.80%

Risposta : 3

38) Il branch coverage di un insieme di test cases è la percentuale di branch del programma che sono attraversati da almeno un test case.

Si consideri la seguente funzione C:

```
-----  
  
int f(int x, int y) {  
    if (x - y <= 2) { if (x + y >= 1) return (1); else return (2); }  
    else {if (x + 2*y >= 5) return (3); else return (4); }  
} /* f() */
```

Si considerino i seguenti test cases: {x=1, y=2}, {x=0, y=0}, {x=5, y=0}, {x=3, y=0}.

Quale delle seguenti è la branch coverage conseguita?

1.100%

2.50%

3.80%

Risposta : 1

39) Pre-condizioni, invarianti e post-condizioni di un programma possono essere definiti usando la macro del C assert() (in <assert.h>). In particolare, assert(expre) non fa nulla se l'espressione expre vale TRUE (cioè non è 0), stampa un messaggio di errore su stderr e abortisce l'esecuzione del programma altrimenti.

Si consideri la funzione C

```
int f(in x, int y) { ..... }
```

Quale delle seguenti assert esprime la pre-condizione che entrambi gli argomenti di f sono non-negativi ed almeno uno di loro è positivo ?

1.int f(in x, int y) { assert((x >= 0) && (y >= 0) && ((x > 0) || (y > 0)));.....}

2.int f(in x, int y) { assert((x > 0) && (y > 0) && ((x > 1) || (y > 1)));.....}

3.int f(in x, int y) { assert((x >= 0) && (y >= 0) && ((x > 1) || (y > 1)));.....}

40) Quale delle seguenti affermazioni è vera riguardo ai metodi agile ?

- 1.I metodi agile sono metodi di sviluppo incrementale.
- 2.I metodi agile sono metodi di sviluppo plan-driven.
- 3.I metodi agile sono metodi di sviluppo orientato al riuso.

Risposta : 1

41) Pre-condizioni, invarianti e post-condizioni di un programma possono essere definiti usando la macro del C assert() (in <assert.h>). In particolare, assert(expre) non fa nulla se l'espressione expre vale TRUE (cioè non è 0), stampa un messaggio di errore su stderr e abortisce l'esecuzione del programma altrimenti.

Si consideri la funzione C

```
int f(in x, int y) { ..... }
```

Quale delle seguenti assert esprime la pre-condizione che entrambi gli argomenti di f sono positivi ed almeno uno di loro è maggiore di 1 ?

- 1.int f(in x, int y) { assert((x >= 0) && (y >= 0) && ((x > 1) || (y > 1)));.....}
- 2.int f(in x, int y) { assert((x > 0) && (y > 0) && (x > 1) && (y > 1));.....}
- 3.int f(in x, int y) { assert((x > 0) && (y > 0) && ((x > 1) || (y > 1)));.....}

42) La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura

Si consideri il seguente insieme di test cases:

Test case 1: act0 act2 act2 act1 act2 act1 act2 act0 act1

Test case 2: act0 act2 act0

Test case 3: act1 act1 act2

Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra

Immagine

- 1.State coverage: 60%
- 2.State coverage: 90%
- 3.State coverage: 40%

Risposta : 1

43) Un processo software può essere rappresentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del processo software e gli archi le transizioni da una fase all'altra. Gli archi sono etichettati con le probabilità della transizione e gli stati sono etichettati con il costo per lasciare lo stato.

Ad esempio lo state diagram in figura

Rappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità dello 0.4 di dover essere ripetuta (a causa di errori) ed F2 ha una probabilità 0.3 di dover essere ripetuta (a causa di errori).

Uno scenario è una sequenza di stati.

Qual'e' la probabilità dello scenario: 1, 2, 3? In altri termini, qual'è la probabilità che non sia necessario ripetere la prima fase (ma non la seconda) ?

Immagine

- 1.0.28
- 2.0.42
- 3.0.12

Risposta : 1

44) La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

Test case 1: act2 act2 act2 act2 act0 act1 act2 act0

Test case 2: act1 act2

Test case 3: act2 act0

Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?

Immagine

- 1.40%
- 2.90%
- 3.70%

Risposta : 1

45) Un test oracle per un programma P è una funzione booleana che ha come inputs gli inputs ed outputs di P e ritorna true se e solo se il valore di output di P (con i dati inputs) è quello

atteso dalle specifiche.

Si consideri la seguente funzione C:

```
-----  
  
int f(int x, int y) {  
  
    int z, k;  
  
    z = 1;  k = 0;  
  
    while (k < x) { z = y*z; k = k + 1; }  
  
    return (z);  
  
}
```

Siano x, y, gli inputs del programma (f nel nostro caso) e z l'output. Assumendo il programma corretto, quale delle seguenti funzioni booleane F(x, y, z) è un test oracle per la funzione f.

1.F(x, y, z) = if (x >= 0) then (z == pow(y, x)) else (z == 1)

2.F(x, y, z) = if (x >= 0) then (z == pow(y, x)) else (z == y)

3.F(x, y, z) = if (x >= 0) then (z == pow(y, x)) else (z == 0)

Risposta : 1

46) La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

Test case 1: act0 act1 act0 act2 act2 act1 act2 act2 act2 act2 act0 act0

Test case 2: act2

Test case 3: act2

Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?

Immagine

1.60%

2.90%

3.45%

Risposta : 3

47) Un processo software può essere rappresentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del processo software e gli archi le transizioni da una fase all'altra. Gli archi sono etichettati con le probabilità della transizione e gli stati sono etichettati con il costo per lasciare lo stato.

Ad esempio lo state diagram in figura

Rappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità dello 0.3 di dover essere ripetuta (a causa di errori) ed F2 ha una probabilità 0.2 di dover essere ripetuta (a causa di errori).

Uno scenario è una sequenza di stati.

Qual'è la probabilità dello scenario: 1, 3 ? In altri termini, qual'è la probabilità che non sia necessario ripetere nessuna fase?

Immagine

1.0.56

2.0.24

3.0.14

Risposta : 1

48) Si consideri il seguente requisito:

RQ: Dopo 40 unità di tempo dall'inizio dell'esecuzione vale la seguente proprietà:

se 10 unità di tempo nel passato x era maggiore di 1 allora ora y è nonnegativa.

Tenendo presente che, al tempo time, delay(z, w) ritorna 0 se time <= w e ritorna il valore che z aveva al tempo (time - w), se time = w.

Quale dei seguenti monitor meglio descrive il requisito RQ ?

1.class MonitorInputReal x, y; OutputBoolean wy;Boolean wz;initial equationwy = false;equationwz = (time > 40) and (delay(x, 10) > 1) and (y < 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

2.class MonitorInputReal x, y; OutputBoolean wy;Boolean wz;initial equationwy = false;equationwz = (time > 40) and (delay(x, 10) > 1) and (y >= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

3.class MonitorInputReal x, y; OutputBoolean wy;Boolean wz;initial equationwy = false;equationwz = (time > 40) or (delay(x, 10) > 1) or (y < 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 1

49) "Ogni giorno, per ciascuna clinica, il sistema genererà una lista dei pazienti che hanno un appuntamento quel giorno."

La frase precedente è un esempio di:

- 1.Requisito non-funzionale.
- 2.Requisito funzionale.
- 3.Requisito di performance.

Risposta : 2

50) Quali delle seguenti attività è parte del processo di validazione dei requisiti ?

- 1.Accertarsi che il sistema soddisfi i requisiti dati.
- 2.Accertarsi che i requisiti definiscano un sistema che risolve il problema che l'utente pianifica di risolvere.
- 3.Accertarsi che l'architettura del sistema soddisfi i requisiti dati.

Risposta : 2