1) 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 prototipo del sistema e valutarne i requisiti non funzionali usando i dati storici dall'azienda.

2.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 migliorie apportate dal sistema software ai processi di business dell'azienda mediante simulazione.

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

Risposta : 2

2) Un'azienda decide di organizzare il processo di sviluppo di un grosso software in 101 phasi sequenziali, numerate da 0 a 100. La phase 0 è quella iniziale. La phase 100 è quella finale in cui lo sviluppo è completato. Tutte le fasi hanno circa la stessa durata.

Alla fine di ogni fase viene eseguita una batteria di tests. I risultati del testing possono essere:

a) si può passare alla fase successiva;

b) bisogna ripetere la fase corrente;

c) bisogna rivedere il lavoro fatto nella fase precedente (reworking).

Dai dati storici è noto che la probabilità del caso a) è 0.72, del caso b) è 0.18 e del caso c) è 0.1.

Allo scopo di stimare attraverso una simulazione MonteCarlo il valore atteso del tempo di completamento del progetto viene realizzato un modello Modelica del processo di sviluppo descritto sopra.

Quale dei seguenti modelli Modelica modella correttamente il processo di sviluppo descritto sopra?

1.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() <= 0.9) then if (myrandom() <= 0.8) then x := x + 1; else x := max(0, x - 1); end if; else x := max(0, x - 1); end if;end if;end when;end MarkovChain;

2.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() <= 0.8) then if (myrandom() <= 0.9) then x := x + 1; else x := max(0, x - 1); end if; else x := max(0, x - 1); end if;end if;end when;end MarkovChain;

3.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() <= 0.9) then if (myrandom() <= 0.72) then x := x + 1; else x := max(0, x - 1); end if; else x := max(0, x - 1); end if;end if;end when;end MarkovChain;

Risposta : 1

3) Quale delle seguenti frasi è corretta riguardo al Sequence Diagram in figura ?

Immagine

1.Il paziente richiede al server una visita con uno specifico medico e, dopo una verifica sul database, riceve conferma dal server della disponibilità o meno del medico richiesto.

2.Periodicamente il client comunica ai pazienti le disponibilità dei medici.

3.Il paziente richiede al client una visita con uno specifico medico e, dopo una verifica sul database, riceve conferma dal client della disponibilità o meno del medico richiesto.

Risposta : 3

4) Si consideri un software sviluppato seguendo un approccio plan-driven implementato con due fasi: F1, F2. La fase F1 ha costo A e la fase F2 ha costo il 50% di A. Qual'e' il costo dello sviluppo del software?

1.A

2.1.5*A

3.0.5*A

Risposta : 2

5) Si consideri un software sviluppato seguendo un approccio iterativo implementato con due fasi: F1 seguita da F2. Ciascuna fase ha costo A e deve essere ripetuta una seconda volta con probabilità p. Qual'e' il costo atteso dello sviluppo dell'intero software?

1.2*A*(p + 2)

2.3*A*(p + 1)

3.2*A*(p +1)

Risposta : 3

6) 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

7) Si consideri un software costituito da due fasi F1 ed F2 ciascuna di costo A. Con probabilità p la fase F1 deve essere ripetuta (a causa di change requests) e con probabilità (1 - p) si passa alla fase F2 e poi al

completamento (End) dello sviluppo. Qual'eè il costo atteso per lo sviluppo del software seguendo il processo sopra descritto ?

1.3*A*p

2.A*(1 + p)

3.A*(2 + p)

Risposta : 3

8) Si consideri il monitor seguente che ritorna true appena i requisiti per il sistema monitorato sono violati.

block Monitor

input Real x;

output Boolean y;

Boolean w;

initial equation

y = false;

equation

w = ((x < 0) or (x > 5));

algorithm

when edge(w) then

y := true;

end when;

end Monitor;

Quale delle seguenti affermazioni meglio descrive il requisito monitorato.

1.La variabile x è fuori dall'intervallo [0, 5].

2.La variabile x è nell'intervallo [0, 5].

3.La variable x è minore di 0.

Risposta : 2

9) Si consideri un software sviluppato seguendo un approccio plan-driven implementato con tre fasi: F1, F2, F3. Le "change requests" arrivano con probabilità p dopo ciascuna fase e provocano la ripetizione (con relativo costo) di tutte le fasi che precedono. Quali delle seguenti catene di Markov modella lo sviluppo software descritto.

Immagine

Immagine

Immagine

Risposta: Immagine

10) Si pianifica di sviluppare un software gestionale per una università. Considerando che questo può essere considerato un sistema mission-critical, quali dei seguenti modelli di processi software generici è più adatto per lo sviluppo di tale software.

1.Sviluppo Iterativo

2.Sviluppo Agile.

3.Sviluppo plan-driven.

Risposta : 3

11) Quale pattern architetturale meglio descrive l'architettura in figura ?

Immagine

1.Layred architecture.

2.Pipe and filter architecture.

3.Model View Controller.

Risposta : 1

12) Quale delle seguenti affermazioni è vera riguardo al performance testing?

1.Il performance testing è tipicamente eseguito solo sulle componenti del sistema prima dell'integrazione.

2.Il performance testing è tipicamente eseguito su un prototipo del sistema.

3.Il performance testing è tipicamente eseguito una volta che il sistema è stato completamento integrato.

Risposta : 3

13) La validazione risponde alla seguenete domanda:

1.Stiamo costruendo il sistema giusto ?

2.Sono soddisfatti i requisti funzionali ?

3.Stiamo costruendo il sistema nel modo giusto ?

Risposta : 1

14) Si consideri il seguente requisito:

RQ: Durante l'esecuzione del programma (cioè per tutti gli istanti di tempo positivi) la variabile x è sempre nell'intervallo [0, 5].

Quale dei seguenti monitor meglio descrive il requisito RQ ?

1.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x > 5) or (x < 0));algorithmwhen edge(z) theny := true;end when;end Monitor;

2.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and (x > 0) and (x < 5);algorithmwhen edge(z) theny := true;end when;end Monitor;

3.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x > 0) or (x < 5));algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 1

15) Si consideri il seguente requisito: "Il sistema fornisce l'elenco dei clienti in ordine alfabetico". Di che tipo di requisito si tratta?

1.Requisito di sistema.

2.Requisito non-funzionale.

3.Requisito utente.

Risposta : 3

16) Il component testing si concentra su:

1.Testare l'interazione tra molte componenti (cioè integrazione di molte unità).

2.Testare funzionalità di unità software individuali, oggetti, classi o metodi.

3.Testare le interfacce per ciascun componente.

Risposta : 3

17) "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 di performance.

3.Requisito funzionale.

Risposta : 3

18) Si consideri il seguente requisito:

RQ1: Durante l'esecuzione del programma (cioè per tutti gli istanti di tempo positivi) la variabile x è sempre nell'intervallo [0, 5] oppure [10, 15]

Quale dei seguenti monitor meglio descrive il requisito RQ1 ?

1.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x >= 0) or (x <= 5)) and ((x >= 10) or (x <= 15)));algorithmwhen edge(z) theny := true;end when;end Monitor;

2.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and (((x >= 0) and (x <= 5)) or ((x >= 10) and (x <= 15)));algorithmwhen edge(z) theny := true;end when;end Monitor;

3.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x >= 5) or (x <= 0)) and ((x >= 15) or (x <= 10)) ;algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 3

19) Si consideri un software sviluppato seguendo un approccio plan-driven implementato con tre fasi: F1, F2, F3. Dopo ogni fase c'e' una probabilità p di dover ripeter la fase precedente ed una probabilità (1 - p) di passare alla fase successiva (sino ad arrivare al termine dello sviluppo). Quale delle seguenti catene di Markov modella il processo software descritto sopra?

Immagine

Immagine

Immagine

Risposta: Immagine

20) Un azienda ha un team di sviluppo in cui il 90% dei membri è junior (cioè con poca esperienza) ed il 10% è senior (cioè con molta esperienza). Con l'obiettivo di massimizzare il numero di progetti completati nell'unità di tempo, quale dei seguenti modelli di sviluppo software appare più opportuno.

1.Iterativo

2.Basato sul riuso

3.Plan driven

Risposta : 3

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

1.Accertarsi che i requisiti definiscano un sistema che risolve il problema che l'utente pianifica di risolvere.

2.Accertarsi che il sistema soddisfi i requisiti dati.

3.Accertarsi che l'architettura del sistema soddisfi i requisiti dati.

Risposta : 1

22) L'environment di un sistema software è costituito da uno user che, ogni untià di tempo (ad esempio, un secondo) invia al sistema tre numeri: -1, 0, 1, con probabilità, rispettivamente, 0.2, 0.56, 0.24.

Quale dei seguenti modelli Modelica modella correttamente l'environment descritto sopra.

1.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) then if (myrandom() <= 0.7) then if (myrandom() <= 0.8) then x := 0; else x := 1; end if; else x := -1; end if;end when;end MarkovChain;

2.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) then if (myrandom() <= 0.8) then if (myrandom() <= 0.7) then x := 0; else x := 1; end if; else x := -1; end if;end when;end MarkovChain;

3.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) then if (myrandom() <= 0.8) then if (myrandom() <= 0.7) then x := 1; else x := 0; end if; else x := -1; end if;end when;end MarkovChain;

Risposta : 2

23) Il system testing si concentra su:

1.Testare le funzionalità di unità software individuali, oggetti, classi o metodi.

2.Testare le interfacce per ciascuna componente.

3.Testare l'interazione tra le componenti del sistema (cioè, integrazione di molte unità di sistema).

Risposta : 3

24) Quale delle seguenti frasi meglio descrive il criterio di "requirements verifiability" che è parte della "requirements validation activity".

1.Per ciascun requisito, dovremmo essere in grado di scrivere un inseme di test che può dimostrare che il sistema sviluppato soddisfa il requisito considerato.

2.Per ciascuna coppia di componenti, dovremmo essere in grado di scrivere un insieme di test che può dimostrare che l'interazione tra le componenti soddisfa tutti i requisiti di interfaccia.

3.Per ciascuna componente del sistema, dovremmo essere in grado di scrivere un insieme di test che può dimostrare che essa soddisfa tutti i requisiti.

Risposta : 1

25) Un'azienda decide di organizzare il processo di sviluppo di un grosso software in 101 phasi sequenziali, numerate da 0 a 100. La phase 0 è quella iniziale. La phase 100 è quella finale in cui lo sviluppo è completato. Tutte le fasi hanno circa la stessa durata.

Si decide di realizzare un approccio incrementale in cui, alla fine di ogni fase, si passa alla fase successiva solo nel caso in cui tutti i test per la fase vengono superati. In caso contrario bisogna ripetere la phase. Dai dati storici è noto che la probabilità che il team di sviluppo passi da una fase a quella successiva è 0.8.

Allo scopo di stimare attraverso una simulazione MonteCarlo il valore atteso del tempo di completamento del progetto viene realizzato un modello Modelica delo processo di sviluppo descritto sopra.

Quale dei seguenti modelli Modelica modella correttamente il processo di sviluppo descritto sopra?

1.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x; // Connectoralgorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() >= 0.8) then x := x + 1; end if;end if;end when;end MarkovChain;

2.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x; // Connectoralgorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() <= 0.8) then x := x + 1; else x := x - 1; end if;end if;end when;end MarkovChain;

3.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x; // Connectoralgorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() <= 0.8) then x := x + 1; end if;end if;end when;end MarkovChain;

Risposta : 3

26) Si consideri un software sviluppato seguendo un approccio iterativo implementato con tre fasi: F1, F2, F3. Ciascuna fase ha costo A. Qual'e' il costo dello sviluppo dell'intero software?

1.A

2.2*A

3.3*A

Risposta : 3

27) Si consideri un software sviluppato seguendo un approccio iterativo implementato con due fasi: F1 seguita da F2. Ciascuna fase ha costo A. Con probabilità p potrebbe essere necessario ripetere F1 una seconda volta. Con probabilità q potrebbe essere necessario ripetere F2 una seconda volta. Qual'e' il costo atteso dello sviluppo dell'intero software?

1.A*(3 + p +q)

2.A*(2 + p +q)

3.A*(1 + p +q)

Risposta : 2

28) L'input di un sistema software è costituito da una sequenza di 0 (false) ed 1 (true). Ad uno 0 segue un 1 ed ad un 1 segue uno 0. Il tempo tra un valore di input e l'altro è un valore random compreso tra 1 e 10 unità di tempo.

Quale dei seguenti modelli Modelica modella meglio l'environment descritto sopra.

1.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Real x0 = 0;OutputReal x;Integer countdown;algorithmwhen initial() then x := x0; countdown := 0;elsewhen sample(0, 1) then if (countdown >= 0) then countdown := 1 + integer(floor(10*myrandom())); x := 1 - pre(x); else countdown := countdown - 1; end if;end when;end MarkovChain;

2.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Real x0 = 0;OutputReal x;Integer countdown;algorithmwhen initial() then x := x0; countdown := 0;elsewhen sample(0, 1) then if (countdown <= 0) then countdown := 1 + integer(floor(10*myrandom())); x := 1 - pre(x); else countdown := countdown - 1; end if;end when;end MarkovChain;

3.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Real x0 = 0;OutputReal x;Integer countdown;algorithmwhen initial() then x := x0; countdown := 0;elsewhen sample(0, 10) then if (countdown <= 0) then countdown := 1 + integer(floor(myrandom())); x := 1 - pre(x); else countdown := countdown - 1; end if;end when;end MarkovChain;

Risposta : 2

29) L'input di un sistema software è costituito da una sequenza di valori reali. Ad ogni unità di tempo il valore di input può rimanere uguale al precedente oppure differire di un numero random in [-1, 1]. L'input resta costante per numero random di unità di tempo in [1, 10].

Quale dei seguenti modelli Modelica modella meglio l'environment descritto sopra.

1.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Real x0 = 0;OutputReal x;Integer countdown;algorithmwhen initial() then x := x0; countdown := 0;elsewhen sample(0, 1)

then if (countdown <= 0) then countdown := 1 + integer(floor(10*myrandom())); x := x + (-1 + 4*myrandom()); else countdown := countdown - 1; end if;end when;end MarkovChain;

2.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Real x0 = 0;OutputReal x;Integer countdown;algorithmwhen initial() then x := x0; countdown := 0;elsewhen sample(0, 1) then if (countdown <= 0) then countdown := 1 + integer(floor(10*myrandom())); x := x - myrandom(); else countdown := countdown - 1; end if;end when;end MarkovChain;

3.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Real x0 = 0;OutputReal x;Integer countdown;algorithmwhen initial() then x := x0; countdown := 0;elsewhen sample(0, 1) then if (countdown <= 0) then countdown := 1 + integer(floor(10*myrandom())); x := x + (-1 + 2*myrandom()); else countdown := countdown - 1; end if;end when;end MarkovChain;

Risposta : 3

30) Il rischio R può essere calcolato come R = P*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 = exp(-b*S), dove b è una opportuna costante note da dati storici aziendali. Quale sarà il costo dello sviluppo S di un software il cui costo della failure è C ed il rischio ammesso è R?

1.S = b*ln(R/C)

2.S = (1/b)*ln(R/C)

3.S = (1/b)*ln(C/R)

Risposta : 3

31) Si consideri un software sviluppato seguendo un approccio plan-driven implementato con tre fasi: F1, F2, F3 ciascuna con costo A. Le "change request" possono arrivare solo al fine di una fase e provocano la ripetizione (con relativo costo) di tutte le fasi che precedono. Si assuma che dopo la fase F3 (cioè al termine dello sviluppo) arriva una change request. Qual'e' il costo totale per lo sviluppo del software in questione.

1.6*A

2.4*A

3.5*A

Risposta : 1

32) 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) or (x > 20)) and ((y < 0.5*x) or (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) and (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) and (y <= 0.7*x) ;algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 2

33) 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 nonegativa.

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) or (delay(x, 10) > 1) or (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) and (delay(x, 10) > 1) and (y < 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 3

34) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?

Immagine

Immagine

Immagine

Immagine

Risposta: Immagine

35) 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) or ((x >= 20) and (x <= 30)) ;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 >= 30) or (x <= 20)) ;algorithmwhen edge(z) theny := true;end when;end Monitor;

3.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;

Risposta : 2

36) Quali delle seguenti attività può contribuire a validare i requisiti di un sistema ?

1.Costruire un prototipo e testarlo a fondo per evidenziare subito errori di implementazione.

2.Costruire un prototipo e valutarne attentamente le performance.

3.Costruire un prototipo, metterlo in esercizio ed accertarsi che i porti i benefici attesi.

Risposta : 3

37) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?

Immagine

1.Una volta selezionata la bevanda non è possibile cancellare l'operazione.

2.La macchina non dà resto.

3.Una volta inserite monete per due bevande è possibile ottenerle senza reinserire le monete.

Risposta : 1

38) Si pianifica lo sviluppo di un sistema software per controllare il sistema di anti-lock braking in un automobile. Quale dei seguenti è il tipico processo software usato per questo tipo di sistema software ?

1.Sviluppo Iterativo.

2.Extreme programming.

3.Sviluppo Plan-driven.

Risposta : 3

39) Verification answers the following question:

1.Is the system cost reasonable for the intended market ?

2.Are we building the right system??

3.Are we building the system right??

40) Si consideri il Test-Driven Development (TDD). Quale delle seguenti affermazioni è vera?

1.Scrivi test automatizzati per tutti i requisiti di sistema, esegui i test e rivedi l'implementazione come necessario.

2.Per ciascun incremento di funzionalità, scrivi test automatizzati, implementa la funzionalità, esegui i test e rivedi l'implementazione come necessario.

3.Per ciascun incremento di funzionalità, implementa la funzionalità, scrivi test automatizzati, esegui i test e rivedi l'implementazione come necessario.

Risposta : 2

41) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?

Immagine

1.Se ci sono abbastanza monete è sempre possibile ottenere la bevanda selezionata.

2.Una volta inserite le monete bisogna necessariamente consumare almeno una bevanda.

3.Anche se ci sono abbastanza monete potrebbe non essere possibile ottenere la bevanda selezionata.

Risposta : 1

42) Lo state diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti modelli Modelica è plausibile per lo state diagram in figura?

Immagine

1.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer := 0; elsewhen sample(0, T) thenif (pre(state) == 0) and (Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state := 3; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 0; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end CoffeeMachine;

2.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer := 0; elsewhen sample(0, T) thenif (pre(state) == 0) and (Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state := 0; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 3; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end CoffeeMachine;

3.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer := 0; elsewhen sample(0, T) thenif (pre(state) == 0) and (Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state := 3; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 3; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end CoffeeMachine;

Risposta : 3

43) 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) and (delay(x, 10) <= 0) 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 > 60) or (delay(x, 10) > 0) or (y >= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 1

44) Una azienda manifatturiera desidera costruire un sistema software per monitorare (attraverso sensori) la produzione al fine di ridurre gli scarti. Quali delle seguenti attività contribuisce a validare i requisiti del sistema.

1.Costruire un prototipo, eseguirlo usando dati storici dai log di produzione ed identificare errori di implementazione.

2.Costruire un prototipo, eseguirlo usando dati storici dai log di produzione e valutare la capacità del prototipo di ridurre gli scarti.

3.Costruire un prototipo, eseguirlo usando dati storici dai log di produzione e valutarne le performance.

Risposta : 2

45) 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 era stata richiesta una risorsa (variabile x positiva) allora ora è concesso l'accesso alla risorsa (variabile y positiva)

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) and (delay(x, 10) > 0) 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 > 60) or (delay(x, 10) > 0) or (y <= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 1

46) Si consideri il monitor seguente che ritorna true appena il sistema viola il requisito monitorato.

```
block Monitor
input Real x;
output Boolean y;
Boolean w;
initial equation
y = false;
equation
w = ((x < 1) or (x > 4)) and ((x < 15) or (x > 20));
algorithm
when edge(w) then
y := true;
end when;
end Monitor;
```

Quale delle seguenti affermazioni meglio descrive il requisito monitorato?

1.La variabile x è nell'intervallo [1, 4] e fuori dall'intervallo [15, 20].

2.La variabile x è nell'intervallo [1, 4] oppure nell'intervallo [15, 20].

3.La variabile x è fuori dall'intervallo [1, 4] e fuori dall'intervallo [15, 20].

Risposta : 2

47) Quale delle seguenti frasi è corretta riguardo all'ctivity diagram in figura ?

Immagine

1.Una volta selezionato il piatto di mare da preparare, la stessa persona prepara prima il pesce e poi il contorno.

2.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in sequenza.

3.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in parallelo.

Risposta : 3

48) Il rischio R può essere calcolato come R = P*C, dove P è la probabilità dell'evento avverso (software failure nel nostro contesto) e C è il costo dell'occorrenza dell'evento avverso. Si consideri un software il cui costo per la failure è C = 1000000 EUR. Volendo un rischio non superiore a 1000 EUR quale è il valore massimo della probabilità di failure P accettabile?

1.1/10

2.1/100

3.1/1000

Risposta : 3

49) Unit testing si concentra su:

1.Testare le interfacce di ciascuna componente.

2.Testare l'interazione tra componenti.

3.Testare funzionalità di unità software individuali, oggetti, classi o metodi.

Risposta : 3

50) L'input di un sistema software è costituito da un sensore che ogni unità di tempo (ad esempio, un secondo) invia un numero reale. Con probabilità 0.63 il valore inviato in una unità di tempo è maggiore del 10% rispetto quello inviato nell'unità di tempo precedente. Con probabilità 0.1 è inferiore del 27% rispetto al valore inviato nell'unità di tempo precedente. Con probabilità 0.27 è inferiore del 10% rispetto quello inviato nell'unità di tempo precedente.

Quale dei seguenti modelli Modelica modella correttamente l'environment descritto sopra.

1.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Real x0 = 1;OutputReal x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (myrandom() <= 0.9)then if (myrandom() <= 0.7) then x := 1.1*x; else x := 0.9*x; end if;else x := 0.73*x; end if;end when;end MarkovChain;

2.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Real x0 = 1;OutputReal x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (myrandom() <= 0.7)then if (myrandom() <= 0.9) then x := 1.1*x; else x := 0.9*x; end if;else x := 0.73*x; end if;end when;end MarkovChain;

3.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Real x0 = 1;OutputReal x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (myrandom() <= 0.9)then if (myrandom() <= 0.7) then x := 0.9*x; else x := 01.1*x; end if;else x := 0.73*x; end if;end when;end MarkovChain;

Risposta : 1

1) 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 act1 act2

Test case 2: act0 act2 act0

Test case 3: act0 act0 act0 act1 act1

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: 30%

3.Transition coverage: 40%

Risposta : 3

2) Quale delle seguenti frasi meglio descrive il criterio di "requirements verifiability" che è parte della "requirements validation activity".

1.Per ciascuna componente del sistema, dovremmo essere in grado di scrivere un insieme di test che può dimostrare che essa soddisfa tutti i requisiti.

2.Per ciascun requisito, dovremmo essere in grado di scrivere un inseme di test che può dimostrare che il sistema sviluppato soddisfa il requisito considerato.

3.Per ciascuna coppia di componenti, dovremmo essere in grado di scrivere un insieme di test che può dimostrare che l'interazione tra le componenti soddisfa tutti i requisiti di interfaccia.

Risposta : 2

3) 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 >= 20) and (x <= 30) ;algorithmwhen edge(z) theny := true;end when;end Monitor;

2.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;

3.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;

Risposta : 3

4) 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 act0 act1 act0 act2

Test case 2: act0 act2 act2 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.Transition coverage: 35%

2.Transition coverage: 80%

3.Transition coverage: 60%

Risposta : 1

5) 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 <= 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 >= 0) && (b - c - 1 <= 0) )
         return (1);    // punto di uscita 1
     else if ((b - c - 1 <= 0)  || (b + c - 5 >= 0))
          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 = 0).

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

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

6) Il rischio R può essere calcolato come R = P*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*S)}   (cioè 10 elevato alla (-b*S))

dove b è una opportuna costante note 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.

1.500000 EUR

2.300000 EUR

3.700000 EUR

Risposta : 2

7) Un processo software può essere rappesentato 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 probabilita 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 terminti, qual'è la probabilità che non sia necessario ripetere la prima fase (ma non la seconda) ?

Immagine

1.0.12

2.0.28

3.0.42

Risposta : 2

8) 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) and (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) or (x > 20)) and ((y < 0.5*x) or (y > 0.7*x)) ;algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 2

9) 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 act1 act0 act2

Test case 2: act0 act2 act2 act0 act1

Test case 3: act0 act0

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

Immagine

1.State coverage: 50%

2.State coverage: 80%

3.State coverage: 100%

Risposta : 3

10) 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:

{(-inf, -11], [-10, -1], {0}, [1, 50], [51, +inf)}

Si consideri il seguente insieme di test cases:

{x=-20, x= 10, x=60}

Quale delle seguenti è la partition coverage conseguita?

1.60%

2.80%

3.100%

Risposta : 1

11) Un processo software può essere rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del prcoesso 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 probabilita 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'e' la probabilità dello scenario: 1, 3 ? In altri terminti, qual'è la probabilità che non sia necessario ripetere nessuna fase?

Immagine

1.0.56

2.0.14

3.0.24

Risposta : 1

12) Quale delle seguenti frasi meglio descrive l'obiettivo del "validity check" che è parte della "requirements validation activity".

1.Assicurarsi che un sistema che soddisfa i requisiti risolve il problema del "customer".

2.Assicurarsi che non ci siano requisiti in conflitto con altri requisiti.

3.Assicurarsi che i requisiti funzionali descrivano tutte le funzionalità del sistema.

Risposta : 1

13) Un processo software può essere rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del prcoesso 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 probabilita 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'e' la probabilità dello scenario: 1, 3, 4? In altri terminti, qual'è la probabilità che non sia necessario ripetere la seconda fase (ma non la prima) ?

Immagine

1.0.08

2.0.32

3.0.12

Risposta : 3

14) Un processo software può essere rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del prcoesso 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 probabilita 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'e' la probabilità dello scenario: 1, 2, 3, 4 ? In altri terminti, qual'è la probabilità che sia necessario ripetere sia la fase 1 che la fase 2 ?

Immagine

1.0.07

2.0.03

3.0.27

Risposta : 2

15) 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 era stata richiesta una risorsa (variabile x positiva) allora ora è concesso l'accesso alla risorsa (variabile y positiva)

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) and (delay(x, 10) > 0) 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 > 60) or (delay(x, 10) > 0) or (y <= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 1

16) 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 act2 act1 act2 act2

Test case 2: 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: 90%

2.State coverage: 60%

3.State coverage: 70%

Risposta : 1

17) Si consideri il seguente modello Modelica. Quale dei seguenti UML state diagram lo rappresenta correttamente ?

```
block FSA  //  Finite State Automaton


/* connector declarations outside this block:
connector InputInteger = input Integer;
connector OutputInteger = output Integer;
*/


InputInteger u; // external input
OutputInteger x; // state
parameter Real T = 1;

algorithm

when initial() then
x := 0;

elsewhen sample(0,T) then

if (pre(x) == 1) and (pre(u) == 0) then x := 3;
```

elseif (pre(x) == 1) and (pre(u) == 1) then x := 3;

elseif (pre(x) == 1) and (pre(u) == 2) then x := 2;

elseif (pre(x) == 2) and (pre(u) == 0) then x := 0;

elseif (pre(x) == 2) and (pre(u) == 2) then x := 0;

elseif (pre(x) == 3) and (pre(u) == 0) then x := 2;

elseif (pre(x) == 4) and (pre(u) == 0) then x := 3;

elseif (pre(x) == 4) and (pre(u) == 2) then x := 2;

else x := pre(x); // default

end if;

end when;

end FSA;

Immagine

Immagine

Immagine

Risposta: Immagine

18) Si consideri il monitor seguente che ritorna true appena i requisiti per il sistema monitorato sono violati.

block Monitor

input Real x;

output Boolean y;

Boolean w;

initial equation

y = false;

equation

w = ((x < 0) or (x > 5));

algorithm

when edge(w) then

y := true;

end when;

end Monitor;

Quale delle seguenti affermazioni meglio descrive il requisito monitorato.

1.La variabile x è nell'intervallo [0, 5].

2.La variable x è minore di 0.

3.La variabile x è fuori dall'intervallo [0, 5].

Risposta : 1

19) 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 act1

Test case 2: act1 act0 act1 act0 act2

Test case 3: act0 act2 act2 act1

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

Immagine

1.Transition coverage: 40%

2.Transition coverage: 30%

3.Transition coverage: 80%

Risposta : 1

20) Si consideri il seguente modello Modelica. Quale dei seguenti UML state diagram lo rappresenta correttamente ?

block FSA // Finite State Automaton

/* connector declarations outside this block:

connector InputInteger = input Integer;

connector OutputInteger = output Integer;

```
*/


InputInteger u; // external input

OutputInteger x; // state

parameter Real T = 1;


algorithm


when initial() then

x := 0;


elsewhen sample(0,T) then


if (pre(x) == 0) and (pre(u) == 0) then x := 1;

elseif (pre(x) == 0) and (pre(u) == 1) then x := 1;

elseif (pre(x) == 0) and (pre(u) == 2) then x := 1;

elseif (pre(x) == 1) and (pre(u) == 1) then x := 4;

elseif (pre(x) == 1) and (pre(u) == 2) then x := 0;

elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;

elseif (pre(x) == 2) and (pre(u) == 1) then x := 1;

elseif (pre(x) == 3) and (pre(u) == 0) then x := 4;

elseif (pre(x) == 4) and (pre(u) == 0) then x := 2;

elseif (pre(x) == 4) and (pre(u) == 1) then x := 0;

elseif (pre(x) == 4) and (pre(u) == 2) then x := 2;

else x := pre(x); // default

end if;


end when;

end FSA;
```

Immagine

Immagine

Immagine

Risposta: Immagine

21) Quale delle seguenti frasi meglio descrive l'obiettivo del "check di completezza" che è parte della "requirements validation activity".

1.Assicurarsi che i requisisti descrivano tutte le funzionalità e vincoli (e.g., security, performance) del sistema desiderato dal customer.

2.Assicurarsi che per ogni requisito sia stato implementato nel sistema.

3.Assicurarsi che i requisiti funzionali descrivano tutte le funzionalità del sistema.

Risposta : 1

22) Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ?

Immagine

1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 4;elseif (pre(x) == 0) and (pre(u) == 1) then x := 4;elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;elseif (pre(x) == 1) and (pre(u) == 0) then x := 4;elseif (pre(x) == 1) and (pre(u) == 1) then x := 0;elseif (pre(x) == 1) and (pre(u) == 2) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 1;elseif (pre(x) == 2) and (pre(u) == 1) then x := 4;elseif (pre(x) == 3) and (pre(u) == 0) then x := 1;elseif (pre(x) == 3) and (pre(u) == 1) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 0;else x := pre(x); // defaultend if;end when;end FSA;

2.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 1) then x := 4;elseif (pre(x) == 0) and (pre(u) == 2) then x := 4;elseif (pre(x) == 1) and (pre(u) == 2) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 0;elseif (pre(x) == 2) and (pre(u) == 1) then x := 3;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 0) then x := 0;elseif (pre(x) == 3) and (pre(u) == 1) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 1) then x := 2;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 3;elseif (pre(x) == 0) and (pre(u) == 1) then x := 1;elseif (pre(x) == 2) and (pre(u) == 0) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 0;elseif (pre(x) == 4) and (pre(u) == 0) then x := 0;else x := pre(x); // defaultend if;end when;end FSA;

Risposta : 1

23) Quale delle seguenti frasi meglio descrive l'obiettivo del "check di consistenza" che è parte della "requirements validation activity".

1.Assicurarsi che per ogni requisito esista un insieme di test che lo possa verificare.

2.Assicurarsi che i requisiti funzionali descrivano tutte le funzionalità del sistema.

3.Assicurarsi che non ci siano requisiti in conflitto con altri requisiti.

Risposta : 3

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 <= 0)   { if (x + y - 2>= 0)  return (1); else return (2); }
  else {if (2*x + y - 1>= 0)  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

25) 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)/p

2.1/(1 - p)

3.1/(p*(1 - p))

Risposta : 2

26) 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 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 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 Time(X) della sequenza di stati X = x(0), x(1), x(2), ... è Time(X) = time(x(0)) + time(x(1)) + time(x(2)) + ...

Ad esempio se X = 0, 1 abbiamo Time(X) = time(0) + time(1) = time(0) (poichè time(1) = 0).

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

Immagine

1.time(0)*(1 - p)/p

2.time(0)/(p*(1 - p))

3.time(0)/(1 - p)

Risposta : 3

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

1.Accertarsi che i requisiti definiscano un sistema che risolve il problema che l'utente pianifica di risolvere.

2.Accertarsi che il sistema soddisfi i requisiti dati.

3.Accertarsi che l'architettura del sistema soddisfi i requisiti dati.

Risposta : 1

28) Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ?

Immagine

1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 1;elseif (pre(x) == 1) and (pre(u) == 1) then x := 3;elseif (pre(x) == 1) and (pre(u) == 2) then x := 2;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 1) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 4;elseif (pre(x) == 3) and (pre(u) == 0) then x := 0;elseif (pre(x) == 3) and (pre(u) == 1) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 2;elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 0;else x := pre(x); // defaultend if;end when;end FSA;

2.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; //

stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 1;elseif (pre(x) == 0) and (pre(u) == 2) then x := 1;elseif (pre(x) == 1) and (pre(u) == 2) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 1) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 4;elseif (pre(x) == 3) and (pre(u) == 0) then x := 2;elseif (pre(x) == 3) and (pre(u) == 1) then x := 0;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 1) then x := 2;elseif (pre(x) == 4) and (pre(u) == 2) then x := 0;else x := pre(x); // defaultend if;end when;end FSA;

3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 4;elseif (pre(x) == 0) and (pre(u) == 2) then x := 1;elseif (pre(x) == 1) and (pre(u) == 0) then x := 3;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 2) and (pre(u) == 0) then x := 3;elseif (pre(x) == 2) and (pre(u) == 1) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

Risposta : 2

29) 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 : 2

30) 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 <= 0)   { if (x + y - 3 >= 0)  return (1); else return (2); }
  else {if (x + 2*y -15 >= 0)  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=0, y=0}, {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=2, y=1}, {x=15, y=0}, {x=9, y=0}.

Risposta : 1

31) Si consideri il seguente requisito:

RQ: Durante l'esecuzione del programma (cioè per tutti gli istanti di tempo positivi) la variabile x è sempre nell'intervallo [0, 5].

Quale dei seguenti monitor meglio descrive il requisito RQ ?

1.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and (x > 0) and (x < 5);algorithmwhen edge(z) theny := true;end when;end Monitor;

2.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x > 5) or (x < 0));algorithmwhen edge(z) theny := true;end when;end Monitor;

3.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x > 0) or (x < 5));algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 2

32) 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 = -150, x = 0, x = 100, x = 700}

2.{x = -150, x = -40, x = 0, x = 200, x = 600}

3.{x = -200, x = -50, x = 0, x = 100, x = 500}

Risposta : 2

33) 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 act1 act2 act1

Test case 2: act1 act0 act2

Test case 3: act2 act1 act0

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

Immagine

1.Transition coverage: 30%

2.Transition coverage: 80%

3.Transition coverage: 50%

Risposta : 3

34) Si consideri il seguente modello Modelica. Quale dei seguenti UML state diagram lo rappresenta correttamente ?

```
block FSA  //  Finite State Automaton


/* connector declarations outside this block:
connector InputInteger = input Integer;
connector OutputInteger = output Integer;
*/


InputInteger u; // external input
OutputInteger x; // state
parameter Real T = 1;


algorithm


when initial() then
x := 0;


elsewhen sample(0,T) then


if (pre(x) == 0) and (pre(u) == 0) then x := 3;
elseif (pre(x) == 1) and (pre(u) == 1) then x := 2;
elseif (pre(x) == 1) and (pre(u) == 2) then x := 4;
elseif (pre(x) == 2) and (pre(u) == 1) then x := 4;
elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;
elseif (pre(x) == 3) and (pre(u) == 0) then x := 1;
elseif (pre(x) == 3) and (pre(u) == 1) then x := 2;
elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;
elseif (pre(x) == 4) and (pre(u) == 0) then x := 3;
elseif (pre(x) == 4) and (pre(u) == 1) then x := 2;
elseif (pre(x) == 4) and (pre(u) == 2) then x := 0;
else x := pre(x); // default
end if;


end when;
end FSA;
```

Immagine

Immagine

Immagine

Risposta: Immagine

35) Si consideri il monitor seguente che ritorna true appena il sistema viola il requisito monitorato.

```
block Monitor
input Real x;
output Boolean y;
Boolean w;
initial equation
```

y = false;

equation

w = ((x < 1) or (x > 4)) and ((x < 15) or (x > 20));

algorithm

when edge(w) then

y := true;

end when;

end Monitor;

Quale delle seguenti affermazioni meglio descrive il requisito monitorato?

1.La variabile x è nell'intervallo [1, 4] e fuori dall'intervallo [15, 20].

2.La variabile x è nell'intervallo [1, 4] oppure nell'intervallo [15, 20].

3.La variabile x è fuori dall'intervallo [1, 4] e fuori dall'intervallo [15, 20].

Risposta : 2

36) 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 act0

Test case 2: act2 act0 act1

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: 60%

3.State coverage: 25%

Risposta : 3

37) 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 rappreenta 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), ... è C(X) = c(x(0)) + c(x(1)) + c(x(2)) + ...

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)

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

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

Risposta : 1

38) 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(int 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 maggiore di 3 ?

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

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

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

39) 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) = if (x > y) then (z == x) else (z == y + 1)

3.F(x, y, z) = if (x > y) then (z == x + 1) else (z == y + 1)

Risposta : 2

40) 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 >= 0)  return (1); else return (2); }
  else {if (2*x + y - 5 >= 0)  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=3}.

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

Risposta : 3

41) 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 nonegativa.

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 : 2

42) 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 <= 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 - 100 >= 0) && (b - c - 1 <= 0) )
        return (1);    // punto di uscita 1
     else if ((b - c - 1 <= 0)  || (b + c - 5 >= 0)
)
         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=200, b = 0, c = 1), (a=50, b = 5, c = 0), (a=50, b = 0, c = 5).

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

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

Risposta : 3

43) Si consideri il seguente modello Modelica. Quale dei seguenti UML state diagram lo rappresenta correttamente ?

```
block FSA  //  Finite State Automaton


/* connector declarations outside this block:
connector InputInteger = input Integer;
connector OutputInteger = output Integer;
*/


InputInteger u; // external input
OutputInteger x; // state
parameter Real T = 1;


algorithm


when initial() then
x := 0;


elsewhen sample(0,T) then


if (pre(x) == 0) and (pre(u) == 0) then x := 2;
elseif (pre(x) == 0) and (pre(u) == 1) then x := 4;
elseif (pre(x) == 0) and (pre(u) == 2) then x := 1;
elseif (pre(x) == 1) and (pre(u) == 0) then x := 2;
elseif (pre(x) == 1) and (pre(u) == 1) then x := 4;
elseif (pre(x) == 1) and (pre(u) == 2) then x := 4;
elseif (pre(x) == 2) and (pre(u) == 0) then x := 1;
elseif (pre(x) == 2) and (pre(u) == 1) then x := 1;
elseif (pre(x) == 2) and (pre(u) == 2) then x := 0;
elseif (pre(x) == 3) and (pre(u) == 0) then x := 1;
elseif (pre(x) == 3) and (pre(u) == 2) then x := 0;
elseif (pre(x) == 4) and (pre(u) == 1) then x := 0;
elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;
else x := pre(x); // default
end if;


end when;
end FSA;
```

Immagine

Immagine

Immagine

Risposta: Immagine

44) Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ?

Immagine

1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 4;elseif (pre(x) == 0) and (pre(u) == 1) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 4;elseif (pre(x) == 1) and (pre(u) == 0) then x := 3;elseif (pre(x) == 1) and (pre(u) == 1) then x := 0;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 2) and (pre(u) == 0) then x := 0;elseif (pre(x) == 2) and (pre(u) == 1) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;elseif (pre(x) == 3) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 3;elseif (pre(x) == 4) and (pre(u) == 1) then x := 0;elseif (pre(x) == 4) and (pre(u) == 2) then x := 0;else x := pre(x); // defaultend if;end when;end FSA;

2.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 2) then x := 4;elseif (pre(x) == 1) and (pre(u) == 1) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 0;elseif (pre(x) == 2) and (pre(u) == 1) then x := 4;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 0) then x := 2;elseif (pre(x) == 3) and (pre(u) == 1) then x := 0;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 1) then x := 2;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 4;elseif (pre(x) == 0) and (pre(u) == 1) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 4;elseif (pre(x) == 1) and (pre(u) == 0) then x := 3;elseif (pre(x) == 1) and (pre(u) == 1) then x := 4;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 2) and (pre(u) == 0) then x := 1;elseif (pre(x) == 2) and (pre(u) == 1) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 0) then x := 4;elseif (pre(x) == 3) and (pre(u) == 1) then x := 0;elseif (pre(x) == 3) and (pre(u) == 2) then x := 0;elseif (pre(x) == 4) and (pre(u) == 1) then x := 3;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

Risposta : 2

45) 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 act1 act2 act2

Test case 2: act2 act0 act2 act0 act2

Test case 3: act2 act0 act2 act0 act1

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

Immagine

1.State coverage: 87%

2.State coverage: 60%

3.State coverage: 100%

Risposta : 1

46) 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(int x, int y) { ..... }

Quale delle seguenti assert esprime l'invariante che le variabili locali z e w di f() hanno somma minore di 1 oppure maggiore di 7 ?

1.int f(in x, int y) { int z, w;assert((z + w < 1) || (z + w > 7));.....}

2.int f(in x, int y) { int z, w;assert((z + w <= 1) || (z + w >= 7));.....}

3.int f(in x, int y) { int z, w;assert((z + w > 1) || (z + w < 7));.....}

47) Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ?

Immagine

1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 1) then x := 4;elseif (pre(x) == 0) and (pre(u) == 2) then x := 4;elseif (pre(x) == 1) and (pre(u) == 0) then x := 4;elseif (pre(x) == 1) and (pre(u) == 2) then x := 2;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 1) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;elseif (pre(x) == 3) and (pre(u) == 0) then x := 0;elseif (pre(x) == 3) and (pre(u) == 2) then x := 0;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

2.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 3;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;elseif (pre(x) == 1) and (pre(u) == 0) then x := 4;elseif (pre(x) == 1) and (pre(u) == 1) then x := 3;elseif (pre(x) == 1) and (pre(u) == 2) then x := 2;elseif (pre(x) == 2) and (pre(u) == 0) then x := 0;elseif (pre(x) == 3) and (pre(u) == 0) then x := 1;elseif (pre(x) == 3) and (pre(u) == 1) then x := 2;elseif (pre(x) == 4) and (pre(u) == 1) then x := 0;else x := pre(x); // defaultend if;end when;end FSA;

3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 4;elseif (pre(x) == 0) and (pre(u) == 1) then x := 4;elseif (pre(x) == 0) and (pre(u) == 2) then x := 4;elseif (pre(x) == 1) and (pre(u) == 0) then x := 2;elseif (pre(x) == 1) and (pre(u) == 1) then x := 3;elseif (pre(x) == 1) and (pre(u) == 2) then x := 0;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 1) then x := 1;elseif (pre(x) == 2) and (pre(u) == 2) then x := 0;elseif (pre(x) == 3) and (pre(u) == 0) then x := 2;elseif (pre(x) == 3) and (pre(u) == 1) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 4;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 2;else x := pre(x); // defaultend if;end when;end FSA;

Risposta : 3

48) Si consideri il seguente requisito:

RQ1: Durante l'esecuzione del programma (cioè per tutti gli istanti di tempo positivi) la variabile x è sempre nell'intervallo [0, 5] oppure [10, 15]

Quale dei seguenti monitor meglio descrive il requisito RQ1 ?

1.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x >= 0) or (x <= 5)) and ((x >= 10) or (x <= 15)));algorithmwhen edge(z) theny := true;end when;end Monitor;

2.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x >= 5) or (x <= 0)) and ((x >= 15) or (x <= 10)) ;algorithmwhen edge(z) theny := true;end when;end Monitor;

3.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and (((x >= 0) and (x <= 5)) or ((x >= 10) and (x <= 15)));algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 2

49) 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 <= 0)   { if (x + y - 1 >= 0)  return (1); else return (2); }
  else {if (x + 2*y - 5 >= 0)  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?

1.80%

2.100%

3.50%

Risposta : 2

50) "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 di performance.

2.Requisito non-funzionale.

3.Requisito funzionale.

Risposta : 3

1) 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) and (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) or (x > 20)) and ((y < 0.5*x) or (y > 0.7*x)) ;algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 2

2) Si consideri il seguente requisito:

RQ1: Durante l'esecuzione del programma (cioè per tutti gli istanti di tempo positivi) la variabile x è sempre nell'intervallo [0, 5] oppure [10, 15]

Quale dei seguenti monitor meglio descrive il requisito RQ1 ?

1.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x >= 5) or (x <= 0)) and ((x >= 15) or (x <= 10)) ;algorithmwhen edge(z) theny := true;end when;end Monitor;

2.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x >= 0) or (x <= 5)) and ((x >= 10) or (x <= 15)));algorithmwhen edge(z) theny := true;end when;end Monitor;

3.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and (((x >= 0) and (x <= 5)) or ((x >= 10) and (x <= 15)));algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 1

3) Un processo software può essere rappesentato 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 probabilita 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 terminti, qual'è la probabilità che non sia necessario ripetere la prima fase (ma non la seconda) ?

Immagine

1.0.28

2.0.12

3.0.42

Risposta : 1

4) 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 rappreenta 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), ... è C(X) = c(x(0)) + c(x(1)) + c(x(2)) + ...

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)

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

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

Risposta : 1

5) Si consideri il seguente modello Modelica. Quale dei seguenti state diagram lo rappresenta correttamente ?

```
block FSA  //  Finite State Automaton


/* connector declarations outside this block:
connector InputInteger = input Integer;
connector OutputInteger = output Integer;
*/


InputInteger u; // external input
OutputInteger x; // state
parameter Real T = 1;

algorithm

when initial() then
x := 0;

elsewhen sample(0,T) then

if (pre(x) == 0) and (pre(u) == 0) then x := 1;
elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;
elseif (pre(x) == 1) and (pre(u) == 0) then x := 0;
elseif (pre(x) == 1) and (pre(u) == 1) then x := 0;
elseif (pre(x) == 1) and (pre(u) == 2) then x := 2;
elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;
elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;
elseif (pre(x) == 3) and (pre(u) == 1) then x := 0;
elseif (pre(x) == 4) and (pre(u) == 0) then x := 0;
elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;
elseif (pre(x) == 4) and (pre(u) == 2) then x := 0;
else x := pre(x); // default
end if;


end when;
end FSA;
```

Immagine

Immagine

Immagine

Risposta: Immagine

6) 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 act2 act0

Test case 2: act0 act1 act0 act0

Test case 3: act1 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: 75%

2.State coverage: 50%

3.State coverage: 100%

Risposta : 2

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 act1 act1

Test case 2: act0 act0 act2 act1

Test case 3: 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: 60%

2.State coverage: 100%

3.State coverage: 75%

Risposta : 3

8) Il rischio R può essere calcolato come R = P*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*S)} (cioè 10 elevato alla (-b*S))

dove b è una opportuna costante note 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.

1.500000 EUR

2.300000 EUR

3.700000 EUR

Risposta : 2

9) 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 == 0)

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

Risposta : 1

10) 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 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 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 Time(X) della sequenza di stati X = x(0), x(1), x(2), ... è Time(X) = time(x(0)) + time(x(1)) + time(x(2)) + ...

Ad esempio se X = 0, 1 abbiamo Time(X) = time(0) + time(1) = time(0) (poichè time(1) = 0).

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

Immagine

1.time(0)*(1 - p)/p

2.time(0)/(p*(1 - p))

3.time(0)/(1 - p)

Risposta : 3

11) "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

12) 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 era stata richiesta una risorsa (variabile x positiva) allora ora è concesso l'accesso alla risorsa (variabile y positiva)

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) or (delay(x, 10) > 0) or (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) and (delay(x, 10) > 0) 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 > 60) and (delay(x, 10) > 0) and (y > 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 2

13) 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 act2 act0 act2 act2 act0 act2 act0 act0 act2 act2 act2 act2 act1

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

Test case 3: act0 act1 act0 act0 act0 act2 act1 act0 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: 50%

2.State coverage: 87%

3.State coverage: 100%

Risposta : 2

14) Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ?

Immagine

1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 3;elseif (pre(x) == 0) and (pre(u) == 1) then x := 4;elseif (pre(x) == 0) and (pre(u) == 2) then x := 1;elseif (pre(x) == 1) and (pre(u) == 1) then x := 3;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 1) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 0;elseif (pre(x) == 3) and (pre(u) == 0) then x := 2;elseif (pre(x) == 3) and (pre(u) == 1) then x := 0;elseif (pre(x) == 3) and (pre(u) == 2) then x := 4;elseif (pre(x) == 4) and (pre(u) == 0) then x := 3;elseif (pre(x) == 4) and (pre(u) == 1) then x := 0;elseif (pre(x) == 4) and (pre(u) == 2) then x := 2;else x := pre(x); // defaultend if;end when;end FSA;

2.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 1) and (pre(u) == 1) then x := 0;elseif (pre(x) == 1) and (pre(u) == 2) then x := 0;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 2) then x := 4;elseif (pre(x) == 3) and (pre(u) == 0) then x := 1;elseif (pre(x) == 3) and (pre(u) == 1) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 0) then x := 3;elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;else x := pre(x); // defaultend if;end when;end FSA;

3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 1) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;elseif (pre(x) == 1) and (pre(u) == 1) then x := 4;elseif (pre(x) == 2) and (pre(u) == 1) then x := 4;elseif (pre(x) == 3) and (pre(u) == 0) then x := 2;elseif (pre(x) == 3) and (pre(u) == 1) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 0) then x := 2;elseif (pre(x) == 4) and (pre(u) == 1) then x := 0;elseif (pre(x) == 4) and (pre(u) == 2) then x := 2;else x := pre(x); // defaultend if;end when;end FSA;

15) 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 : 1

16) 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:

{(-inf, -11], [-10, -1], {0}, [1, 50], [51, +inf)}

Si consideri il seguente insieme di test cases:

{x=-20, x= 10, x=60}

Quale delle seguenti è la partition coverage conseguita?

1.60%

2.80%

3.100%

Risposta : 1

17) 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 <= 0)   { if (x + y - 3 >= 0) return (1); else return (2); }

  else {if (x + 2*y -15 >= 0) 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=0, y=0}, {x=15, y=0}, {x=10, y=3}.

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

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

Risposta : 3

18) 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(int x, int y) { ..... }

Quale delle seguenti assert esprime l'invariante che le variabili locali z e w di f() hanno somma minore di 1 oppure maggiore di 7 ?

1.int f(in x, int y) { int z, w;assert((z + w > 1) || (z + w < 7));.....}

2.int f(in x, int y) { int z, w;assert((z + w <= 1) || (z + w >= 7));.....}

3.int f(in x, int y) { int z, w;assert((z + w < 1) || (z + w > 7));.....}

19) 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 <= 0)   { if (x + y - 1 >= 0) return (1); else return (2); }

  else {if (x + 2*y - 5 >= 0) 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?

1.50%

2.80%

3.100%

Risposta : 3

20) Un processo software può essere rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del prcoesso 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 probabilita 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'e' la probabilità dello scenario: 1, 2, 3, 4 ? In altri terminti, qual'è la probabilità che sia necessario ripetere sia la fase 1 che la fase 2 ?

Immagine

1.0.03

2.0.27

3.0.07

Risposta : 1

21) Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ?

Immagine

1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 1) then x := 1;elseif (pre(x) == 0) and (pre(u) == 2) then x := 1;elseif (pre(x) == 1) and (pre(u) == 0) then x := 2;elseif (pre(x) == 1) and (pre(u) == 1) then x := 4;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 2) and (pre(u) == 0) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 4;elseif (pre(x) == 3) and (pre(u) == 0) then x := 0;elseif (pre(x) == 3) and (pre(u) == 1) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 4;elseif (pre(x) == 4) and (pre(u) == 0) then x := 3;elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

2.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 1) then x := 1;elseif (pre(x) == 0) and (pre(u) == 2) then x := 1;elseif (pre(x) == 1) and (pre(u) == 0) then x := 2;elseif (pre(x) == 1) and (pre(u) == 1) then x := 3;elseif (pre(x) == 1) and (pre(u) == 2) then x := 0;elseif (pre(x) == 2) and (pre(u) == 0) then x := 0;elseif (pre(x) == 2) and (pre(u) == 1) then x := 3;elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;elseif (pre(x) == 3) and (pre(u) == 0) then x := 4;elseif (pre(x) == 3) and (pre(u) == 1) then x := 4;elseif (pre(x) == 3) and (pre(u) == 2) then x := 0;elseif (pre(x) == 4) and (pre(u) == 0) then x := 3;elseif (pre(x) == 4) and (pre(u) == 1) then x := 3;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 3;elseif (pre(x) == 1) and (pre(u) == 0) then x := 2;elseif (pre(x) == 1) and (pre(u) == 1) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 3;elseif (pre(x) == 2) and (pre(u) == 2) then x := 0;elseif (pre(x) == 3) and (pre(u) == 1) then x := 2;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

Risposta : 1

22) 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) = if (x > y) then (z == x + 1) else (z == y + 1)

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

3.F(x, y, z) = if (x > y) then (z == x) else (z == y + 1)

Risposta : 3

23) 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 >= 0)  return (1); else return (2); }
  else {if (2*x + y - 5 >= 0)  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=0, y=0}, {x=2, y=1}, {x=2, y=3}.

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

Risposta : 3

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 <= 0)   { if (x + y - 2>= 0)  return (1); else return (2); }
  else {if (2*x + y - 1>= 0)  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%

2.80%

3.50%

Risposta : 1

25) 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 <= 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 - 100 >= 0) && (b - c - 1 <= 0) )
         return (1);    // punto di uscita 1
     else if ((b - c - 1 <= 0)  || (b + c - 5 >= 0)
)
          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=200, b = 0, c = 1), (a=50, b = 5, c = 0), (a=50, b = 3, c = 0).

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

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

Risposta : 1

26) 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 = -150, x = 0, x = 100, x = 700}

2.{x = -150, x = -40, x = 0, x = 200, x = 600}

3.{x = -200, x = -50, x = 0, x = 100, x = 500}

Risposta : 2

27) Quali delle seguenti attività può contribuire a validare i requisiti di un sistema ?

1.Costruire un prototipo, metterlo in esercizio ed accertarsi che i porti i benefici attesi.

2.Costruire un prototipo e testarlo a fondo per evidenziare subito errori di implementazione.

3.Costruire un prototipo e valutarne attentamente le performance.

Risposta : 1

28) Si consideri il seguente requisito:

RQ: Durante l'esecuzione del programma (cioè per tutti gli istanti di tempo positivi) la variabile x è sempre nell'intervallo [0, 5].

Quale dei seguenti monitor meglio descrive il requisito RQ ?

1.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x > 5) or (x < 0));algorithmwhen edge(z) theny := true;end when;end Monitor;

2.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x > 0) or (x < 5));algorithmwhen edge(z) theny := true;end when;end Monitor;

3.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and (x > 0) and (x < 5);algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 1

29) 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 nonegativa.

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) or (delay(x, 10) > 1) 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 > 40) and (delay(x, 10) > 1) and (y >= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 1

30) 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 act0 act2 act0 act0 act0 act2 act1 act1 act0 act2 act0 act2 act2 act1 act1 act0 act2 act2 act2 act1 act1 act2 act0 act1 act0 act1 act2

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

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

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

Immagine

1.Transition coverage: 50%

2.Transition coverage: 75%

3.Transition coverage: 100%

Risposta : 2

31) Si consideri il monitor seguente che ritorna true appena il sistema viola il requisito monitorato.

```
block Monitor
input Real x;
output Boolean y;
Boolean w;
initial equation
y = false;
equation
w = ((x < 1) or (x > 4)) and ((x < 15) or (x > 20));
algorithm
when edge(w) then
y := true;
end when;
end Monitor;
```

Quale delle seguenti affermazioni meglio descrive il requisito monitorato?

1.La variabile x è nell'intervallo [1, 4] e fuori dall'intervallo [15, 20].

2.La variabile x è fuori dall'intervallo [1, 4] e fuori dall'intervallo [15, 20].

3.La variabile x è nell'intervallo [1, 4] oppure nell'intervallo [15, 20].

Risposta : 3

32) 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(int 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 maggiore di 3 ?

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

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

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

33) Si consideri il monitor seguente che ritorna true appena i requisiti per il sistema monitorato sono violati.

block Monitor

input Real x;

output Boolean y;

Boolean w;

initial equation

y = false;

equation

w = ((x < 0) or (x > 5));

algorithm

when edge(w) then

y := true;

end when;

end Monitor;

Quale delle seguenti affermazioni meglio descrive il requisito monitorato.

1.La variabile x è fuori dall'intervallo [0, 5].

2.La variabile x è nell'intervallo [0, 5].

3.La variable x è minore di 0.

Risposta : 2

34) Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ?

Immagine

1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 2) then x := 1;elseif (pre(x) == 1) and (pre(u) == 0) then x := 0;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 2) and (pre(u) == 0) then x := 1;elseif (pre(x) == 2) and (pre(u) == 1) then x := 1;elseif (pre(x) == 2) and (pre(u) == 2) then x := 4;elseif (pre(x) == 3) and (pre(u) == 0) then x := 2;elseif (pre(x) == 3) and (pre(u) == 1) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 4;elseif (pre(x) == 4) and (pre(u) == 0) then x := 3;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

2.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 4;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 2) then x := 2;elseif (pre(x) == 1) and (pre(u) == 1) then x := 2;elseif (pre(x) == 2) and (pre(u) == 1) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 0) then x := 1;elseif (pre(x) == 3) and (pre(u) == 1) then x := 0;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 3;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 4;elseif (pre(x) == 1) and (pre(u) == 0) then x := 4;elseif (pre(x) == 1) and (pre(u) == 1) then x := 0;elseif (pre(x) == 2) and (pre(u) == 0) then x := 1;elseif (pre(x) == 2) and (pre(u) == 1) then x := 3;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 1) then x := 0;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 1) then x := 3;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

Risposta : 2

35) Si consideri il seguente modello Modelica. Quale dei seguenti state diagram lo rappresenta correttamente ?

block FSA // Finite State Automaton

/* connector declarations outside this block:

connector InputInteger = input Integer;

connector OutputInteger = output Integer;

*/

InputInteger u; // external input

OutputInteger x; // state

parameter Real T = 1;

algorithm

when initial() then

x := 0;

elsewhen sample(0,T) then

if (pre(x) == 0) and (pre(u) == 0) then x := 1;

elseif (pre(x) == 0) and (pre(u) == 1) then x := 4;

elseif (pre(x) == 0) and (pre(u) == 2) then x := 4;

elseif (pre(x) == 1) and (pre(u) == 1) then x := 3;

elseif (pre(x) == 1) and (pre(u) == 2) then x := 4;

elseif (pre(x) == 3) and (pre(u) == 0) then x := 4;

elseif (pre(x) == 3) and (pre(u) == 1) then x := 4;

elseif (pre(x) == 4) and (pre(u) == 0) then x := 0;

elseif (pre(x) == 4) and (pre(u) == 1) then x := 2;

elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;

else x := pre(x); // default

end if;

end when;

end FSA;

Immagine

Immagine

Immagine

Risposta: Immagine

36) 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) or ((x >= 20) and (x <= 30)) ;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) and ((x >= 30) or (x <= 20)) ;algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 3

37) Un processo software può essere rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del prcoesso 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 probabilita 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'e' la probabilità dello scenario: 1, 3, 4? In altri terminti, qual'è la probabilità che non sia necessario ripetere la seconda fase (ma non la prima) ?

Immagine

1.0.12

2.0.32

3.0.08

Risposta : 1

38) 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)/p

2.1/(1 - p)

3.1/(p*(1 - p))

Risposta : 2

39) 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 act2 act0 act1

Test case 2: act1 act0 act1 act1 act2 act2 act0

Test case 3: act1 act2 act0 act0

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: 40%

3.Transition coverage: 100%

Risposta : 2

40) Quale delle seguenti frasi meglio descrive il criterio di "requirements verifiability" che è parte della "requirements validation activity".

1.Per ciascuna coppia di componenti, dovremmo essere in grado di scrivere un insieme di test che può dimostrare che l'interazione tra le componenti soddisfa tutti i requisiti di interfaccia.

2.Per ciascun requisito, dovremmo essere in grado di scrivere un inseme di test che può dimostrare che il sistema sviluppato soddisfa il requisito considerato.

3.Per ciascuna componente del sistema, dovremmo essere in grado di scrivere un insieme di test che può dimostrare che essa soddisfa tutti i requisiti.

Risposta : 2

41) 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 act2 act0

Test case 2: act2 act2 act2 act2 act2 act2 act0

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

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

Immagine

1.Transition coverage: 90%

2.Transition coverage: 35%

3.Transition coverage: 50%

Risposta : 3

42) Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ?

Immagine

1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 3;elseif (pre(x) == 0) and (pre(u) == 1) then x := 1;elseif (pre(x) == 1) and (pre(u) == 0) then x := 3;elseif (pre(x) == 1) and (pre(u) == 1) then x := 0;elseif (pre(x) == 1) and (pre(u) == 2) then x := 0;elseif (pre(x) == 2) and (pre(u) == 0) then x := 3;elseif (pre(x) == 2) and (pre(u) == 2) then x := 4;elseif (pre(x) == 3) and (pre(u) == 0) then x := 4;elseif (pre(x) == 3) and (pre(u) == 1) then x := 4;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 0;else x := pre(x); // defaultend if;end when;end FSA;

2.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 4;elseif (pre(x) == 0) and (pre(u) == 1) then x := 4;elseif (pre(x) == 0) and (pre(u) == 2) then x := 4;elseif (pre(x) == 1) and (pre(u) == 1) then x := 4;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 1) then x := 3;elseif (pre(x) == 2) and (pre(u) == 2) then x := 0;elseif (pre(x) == 3) and (pre(u) == 0) then x := 4;elseif (pre(x) == 3) and (pre(u) == 1) then x := 4;elseif (pre(x) == 3) and (pre(u) == 2) then x := 0;elseif (pre(x) == 4) and (pre(u) == 0) then x := 0;elseif (pre(x) == 4) and (pre(u) == 1) then x := 0;elseif (pre(x) == 4) and (pre(u) == 2) then x := 2;else x := pre(x); // defaultend if;end when;end FSA;

3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;elseif (pre(x) == 1) and (pre(u) == 0) then x := 3;elseif (pre(x) == 1) and (pre(u) == 1) then x := 2;elseif (pre(x) == 1) and (pre(u) == 2) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 1) then x := 1;elseif (pre(x) == 3) and (pre(u) == 1) then x := 0;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 1) then x := 0;elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;else x := pre(x); // defaultend if;end when;end FSA;

43) 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 act1 act1 act2 act1 act1 act0

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

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

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

Immagine

1.State coverage: 50%

2.State coverage: 87%

3.State coverage: 100%

Risposta : 2

44) Si consideri il seguente modello Modelica. Quale dei seguenti state diagram lo rappresenta correttamente ?

```
block FSA  //  Finite State Automaton


/* connector declarations outside this block:
connector InputInteger = input Integer;
connector OutputInteger = output Integer;
*/


InputInteger u; // external input
OutputInteger x; // state
parameter Real T = 1;


algorithm


when initial() then
x := 0;


elsewhen sample(0,T) then


if (pre(x) == 0) and (pre(u) == 0) then x := 1;
elseif (pre(x) == 0) and (pre(u) == 1) then x := 1;
elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;
elseif (pre(x) == 1) and (pre(u) == 0) then x := 4;
elseif (pre(x) == 1) and (pre(u) == 1) then x := 2;
elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;
elseif (pre(x) == 2) and (pre(u) == 1) then x := 0;
elseif (pre(x) == 2) and (pre(u) == 2) then x := 0;
elseif (pre(x) == 3) and (pre(u) == 0) then x := 0;
elseif (pre(x) == 3) and (pre(u) == 1) then x := 2;
elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;
elseif (pre(x) == 4) and (pre(u) == 0) then x := 0;
elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;
else x := pre(x); // default
end if;


end when;
end FSA;
```

Immagine

Immagine

Immagine

Risposta: Immagine

45) 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

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

Test case 3: act0 act0

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

Immagine

1.Transition coverage: 25%

2.Transition coverage: 75%

3.Transition coverage: 50%

Risposta : 1

46) 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 <= 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 >= 0) && (b - c - 1 <= 0) )
         return (1);    // punto di uscita 1
     else if ((b - c - 1 <= 0)  || (b + c - 5 >= 0))
          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 = 5, b = 0, c = 1), (a = 0, b = 5, c = 0), (a = 0, b = 3, c = 0).

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

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

47) Un processo software può essere rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del prcoesso 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 probabilita 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'e' la probabilità dello scenario: 1, 3 ? In altri terminti, qual'è la probabilità che non sia necessario ripetere nessuna fase?

Immagine

1.0.14

2.0.56

3.0.24

Risposta : 2

48) Si consideri il seguente modello Modelica. Quale dei seguenti state diagram lo rappresenta correttamente ?

```
block FSA  //  Finite State Automaton

/* connector declarations outside this block:
connector InputInteger = input Integer;
connector OutputInteger = output Integer;
*/
```

InputInteger u; // external input

OutputInteger x; // state

parameter Real T = 1;

algorithm

when initial() then

x := 0;

elsewhen sample(0,T) then

if (pre(x) == 0) and (pre(u) == 0) then x := 1;

elseif (pre(x) == 0) and (pre(u) == 1) then x := 1;

elseif (pre(x) == 0) and (pre(u) == 2) then x := 2;

elseif (pre(x) == 1) and (pre(u) == 0) then x := 0;

elseif (pre(x) == 1) and (pre(u) == 2) then x := 0;

elseif (pre(x) == 2) and (pre(u) == 0) then x := 3;

elseif (pre(x) == 2) and (pre(u) == 1) then x := 1;

elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;

elseif (pre(x) == 3) and (pre(u) == 0) then x := 1;

elseif (pre(x) == 3) and (pre(u) == 1) then x := 4;

elseif (pre(x) == 3) and (pre(u) == 2) then x := 0;

elseif (pre(x) == 4) and (pre(u) == 1) then x := 2;

elseif (pre(x) == 4) and (pre(u) == 2) then x := 0;

else x := pre(x); // default

end if;

end when;

end FSA;

Immagine

Immagine

Immagine

Risposta: Immagine

49) 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 : 3

50) 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(int *A, 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 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);}

3.#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);}

Risposta : 1

1) 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

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

Test case 3: act2 act0

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

Immagine

1.90%

2.60%

3.75%

Risposta : 2

2) 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

3) Un processo software può essere rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del prcoesso 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 probabilita 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'e' la probabilità dello scenario: 1, 3, 4? In altri terminti, 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

4) Quale delle seguenti frasi meglio descrive l'obiettivo del "check di realismo" (realizability) che è parte della "requirements validation activity".

1.Assicurarsi che le performance richieste al sistema siano necessarie per soddisfare le necessità del customer.

2.Assicurarsi che le funzionalità richieste al sistema siano necessarie per soddisfare le necessità del customer.

3.Assicurarsi che, tenedo conto della tecnologia, budget e tempo disponibili, sia possibile realizzare un sistema che soddisfa i requisisti.

Risposta : 3

5) 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 rappreenta 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), ... è C(X) = c(x(0)) + c(x(1)) + c(x(2)) + ...

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

6) 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) = if (x > y) then (z == x + 1) else (z == y + 1)

3.F(x, y, z) = if (x > y) then (z == x) else (z == y + 1)

Risposta : 3

7) Un processo software può essere rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del prcoesso 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 probabilita 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'e' la probabilità dello scenario: 1, 3 ? In altri terminti, qual'è la probabilità che non sia necessario ripetere nessuna fase?

Immagine

1.0.56

2.0.14

3.0.24

Risposta : 1

8) Si consideri il seguente modello Modelica. Quale dei seguenti state diagram lo rappresenta correttamente ?

```
block FSA  //  Finite State Automaton

/* connector declarations outside this block:
connector InputInteger = input Integer;
connector OutputInteger = output Integer;
*/

InputInteger u; // external input
OutputInteger x; // state
parameter Real T = 1;

algorithm

when initial() then
x := 0;

elsewhen sample(0,T) then

if (pre(x) == 0) and (pre(u) == 0) then x := 2;
elseif (pre(x) == 0) and (pre(u) == 1) then x := 1;
elseif (pre(x) == 0) and (pre(u) == 2) then x := 4;
elseif (pre(x) == 1) and (pre(u) == 0) then x := 3;
elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;
elseif (pre(x) == 2) and (pre(u) == 1) then x := 4;
elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;
elseif (pre(x) == 3) and (pre(u) == 0) then x := 0;
elseif (pre(x) == 3) and (pre(u) == 1) then x := 4;
elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;
elseif (pre(x) == 4) and (pre(u) == 0) then x := 2;
elseif (pre(x) == 4) and (pre(u) == 2) then x := 0;
else x := pre(x); // default
end if;
```

end when;

end FSA;

Immagine

Immagine

Immagine

Risposta: Immagine

9) Si consideri il seguente modello Modelica. Quale dei seguenti state diagram lo rappresenta correttamente ?

```
block FSA  //  Finite State Automaton

/* connector declarations outside this block:
connector InputInteger = input Integer;
connector OutputInteger = output Integer;
*/

InputInteger u; // external input
OutputInteger x; // state
parameter Real T = 1;

algorithm

when initial() then
x := 0;

elsewhen sample(0,T) then

if (pre(x) == 0) and (pre(u) == 0) then x := 4;
elseif (pre(x) == 0) and (pre(u) == 1) then x := 4;
elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;
elseif (pre(x) == 1) and (pre(u) == 1) then x := 2;
elseif (pre(x) == 1) and (pre(u) == 2) then x := 0;
elseif (pre(x) == 2) and (pre(u) == 0) then x := 0;
elseif (pre(x) == 2) and (pre(u) == 1) then x := 1;
elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;
elseif (pre(x) == 3) and (pre(u) == 0) then x := 2;
elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;
elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;
elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;
elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;
else x := pre(x); // default
end if;

end when;
end FSA;
```

Immagine

Immagine

Immagine

Risposta: Immagine

10) 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

11) 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 <= 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 >= 100) && (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=200, b = 0, c = 1), (a=50, b = 5, c = 0), (a=50, b = 3, 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 = 4, c = 0), (a=200, b = 4, c = 0)

Risposta : 1

12) Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ?

Immagine

1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 2) then x := 1;elseif (pre(x) == 1) and (pre(u) == 0) then x := 2;elseif (pre(x) == 1) and (pre(u) == 1) then x := 0;elseif (pre(x) == 1) and (pre(u) == 2) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 0;elseif (pre(x) == 2) and (pre(u) == 1) then x := 4;elseif (pre(x) == 2) and (pre(u) == 2) then x := 4;elseif (pre(x) == 3) and (pre(u) == 0) then x := 1;elseif (pre(x) == 3) and (pre(u) == 1) then x := 0;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 0) then x := 2;elseif (pre(x) == 4) and (pre(u) == 1) then x := 2;else x := pre(x); // defaultend if;end when;end FSA;

2.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 1;elseif (pre(x) == 0) and (pre(u) == 2) then x := 1;elseif (pre(x) == 1) and (pre(u) == 1) then x := 2;elseif (pre(x) == 2) and (pre(u) == 0) then x := 1;elseif (pre(x) == 2) and (pre(u) == 1) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 0;elseif (pre(x) == 3) and (pre(u) == 0) then x := 2;elseif (pre(x) == 3) and (pre(u) == 1) then x := 0;elseif (pre(x) == 3) and (pre(u) == 2) then x := 4;elseif (pre(x) == 4) and (pre(u) == 0) then x := 2;elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 4;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 2) then x := 2;elseif (pre(x) == 1) and (pre(u) == 0) then x := 4;elseif (pre(x) == 1) and (pre(u) == 1) then x := 2;elseif (pre(x) == 1) and (pre(u) == 2) then x := 2;elseif (pre(x) == 2) and (pre(u) == 0) then x := 3;elseif (pre(x) == 2) and (pre(u) == 1) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 0) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

Risposta : 3

13) 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

14) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?

Immagine

1.Se ci sono abbastanza monete è sempre possibile ottenere la bevanda selezionata.

2.Anche se ci sono abbastanza monete potrebbe non essere possibile ottenere la bevanda selezionata.

3.Una volta inserite le monete bisogna necessariamente consumare almeno una bevanda.

Risposta : 1

15) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?

Immagine

Immagine

Immagine

Immagine

Risposta: Immagine

16) 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 act1 act2 act2

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

Test case 3: act2 act2 act0

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 : 2

17) 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 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 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 Time(X) della sequenza di stati X = x(0), x(1), x(2), ... è Time(X) = time(x(0)) + time(x(1)) + time(x(2)) + ...

Ad esempio se X = 0, 1 abbiamo Time(X) = time(0) + time(1) = time(0) (poichè time(1) = 0).

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

Immagine

1.time(0)/(1 - p)

2.time(0)*(1 - p)/p

3.time(0)/(p*(1 - p))

Risposta : 1

18) "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 funzionale.

2.Requisito di performance.

3.Requisito non-funzionale.

Risposta : 1

19) 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 era stata richiesta una risorsa (variabile x positiva) allora ora è concesso l'accesso alla risorsa (variabile y positiva)

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) or (delay(x, 10) > 0) or (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) and (delay(x, 10) > 0) 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 > 60) and (delay(x, 10) > 0) and (y <= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 3

20) 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 act1 act2 act0 act2 act0 act0 act0 act0 act0 act2

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

Test case 3: 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: 60%

3.State coverage: 75%

Risposta : 3

21) Quale delle seguenti frasi meglio descrive l'obiettivo del "check di consistenza" che è parte della "requirements validation activity".

1.Assicurarsi che i requisiti funzionali descrivano tutte le funzionalità del sistema.

2.Assicurarsi che non ci siano requisiti in conflitto con altri requisiti.

3.Assicurarsi che per ogni requisito esista un insieme di test che lo possa verificare.

Risposta : 2

22) 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) and (delay(x, 10) > 0) 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 > 60) or (delay(x, 10) > 0) or (y >= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 2

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 act2

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

Test case 3: act2 act0 act1 act2 act2

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

Immagine

1.State coverage: 75%

2.State coverage: 60%

3.State coverage: 90%

Risposta : 2

24) Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ?

Immagine

1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;elseif (pre(x) == 1) and (pre(u) == 0) then x := 3;elseif (pre(x) == 2) and (pre(u) == 0) then x := 3;elseif (pre(x) == 2) and (pre(u) == 1) then x := 1;elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;elseif (pre(x) == 3) and (pre(u) == 0) then x := 4;elseif (pre(x) == 3) and (pre(u) == 1) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 0;elseif (pre(x) == 4) and (pre(u) == 1) then x := 0;elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;else x := pre(x); //

defaultend if;end when;end FSA;

2.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 4;elseif (pre(x) == 0) and (pre(u) == 2) then x := 2;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 2) and (pre(u) == 0) then x := 0;elseif (pre(x) == 2) and (pre(u) == 1) then x := 3;elseif (pre(x) == 3) and (pre(u) == 0) then x := 4;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 1) then x := 2;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 2) then x := 2;elseif (pre(x) == 1) and (pre(u) == 0) then x := 3;elseif (pre(x) == 1) and (pre(u) == 1) then x := 3;elseif (pre(x) == 1) and (pre(u) == 2) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 1;elseif (pre(x) == 2) and (pre(u) == 1) then x := 4;elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;elseif (pre(x) == 3) and (pre(u) == 0) then x := 4;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 1) then x := 0;elseif (pre(x) == 4) and (pre(u) == 2) then x := 0;else x := pre(x); // defaultend if;end when;end FSA;

Risposta : 2

25) Il rischio R può essere calcolato come R = P*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*S)} (cioè 10 elevato alla (-b*S))

dove b è una opportuna costante note 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

26) Un processo software può essere rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del prcoesso 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 probabilita 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'e' la probabilità dello scenario: 1, 2, 3, 4 ? In altri terminti, qual'è la probabilità che sia necessario ripetere sia la fase 1 che la fase 2 ?

Immagine

1.0.03

2.0.07

3.0.27

Risposta : 1

27) Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ?

Immagine

1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 1) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;elseif (pre(x) == 1) and (pre(u) == 0) then x := 0;elseif (pre(x) == 1) and (pre(u) == 1) then x := 3;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 2) and (pre(u) == 0) then x := 3;elseif (pre(x) == 2) and (pre(u) == 1) then x := 3;elseif (pre(x) == 3) and (pre(u) == 0) then x := 2;elseif (pre(x) == 3) and (pre(u) == 1) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 0;elseif (pre(x) == 4) and (pre(u) == 1) then x := 0;elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;else x := pre(x); // defaultend if;end when;end FSA;

2.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 1) then x := 4;elseif (pre(x) == 0) and (pre(u) == 2) then x := 4;elseif (pre(x) == 1) and (pre(u) == 0) then x := 2;elseif (pre(x) == 1) and (pre(u) == 1) then x := 3;elseif (pre(x) == 1) and (pre(u) == 2) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 0;elseif (pre(x) == 2) and (pre(u) == 1) then x := 1;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 1) then x := 4;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 1) then x := 3;elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;else x := pre(x); // defaultend if;end when;end FSA;

3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 1) then x := 4;elseif (pre(x) == 1) and (pre(u) == 0) then x := 0;elseif (pre(x) == 1) and (pre(u) == 1) then x := 0;elseif (pre(x) == 1) and (pre(u) == 2) then x := 2;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 1) then x := 3;elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;elseif (pre(x) == 3) and (pre(u) == 0) then x := 4;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 3;elseif (pre(x) == 4) and (pre(u) == 1) then x := 2;elseif (pre(x) == 4) and (pre(u) == 2) then x := 2;else x := pre(x); // defaultend if;end when;end FSA;

Risposta : 3

28) Quale delle seguenti frasi è corretta riguardo al Sequence Diagram in figura ?

Immagine

1.Il paziente richiede al server una visita con uno specifico medico e, dopo una verifica sul database, riceve conferma dal server della disponibilità o meno del medico richiesto.

2.Periodicamente il client comunica ai pazienti le disponibilità dei medici.

3.Il paziente richiede al client una visita con uno specifico medico e, dopo una verifica sul database, riceve conferma dal client della disponibilità o meno del medico richiesto.

Risposta : 3

29) 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?

1.80%

2.50%

3.100%

Risposta : 3

30) 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

31) 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

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

Test case 3: act2 act0

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

Immagine

1.50%

2.100%

3.35%

Risposta : 1

32) Quale delle seguenti frasi è corretta riguardo all'ctivity diagram in figura ?

Immagine

1.Una volta selezionato il piatto di mare da preparare, la stessa persona prepara prima il pesce e poi il contorno.

2.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in sequenza.

3.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in parallelo.

Risposta : 3

33) 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 act0 act1 act1 act0 act2 act1 act2 act2 act1 act2 act0 act1 act2 act0 act2 act2 act0 act1 act1 act2 act2 act0 act0 act2 act2 act2 act0 act2 act2 act0 act1 act1 act0 act2 act1 act2 act1 act0 act0 act0 act0 act2 act2 act1 act1 act1 act1 act0

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

Test case 3: act1 act1 act2 act0 act1 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: 75%

3.State coverage: 60%

Risposta : 2

34) 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 act2 act2 act1 act2 act1 act1 act0 act1 act2 act0 act1 act2 act1 act2 act1 act0 act0 act2 act2 act0 act1 act1 act2 act2 act2 act0 act1 act2 act2 act1

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

Test case 3: act1 act1

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

Immagine

1.State coverage: 85%

2.State coverage: 60%

3.State coverage: 100%

Risposta : 1

35) Si consideri il seguente modello Modelica. Quale dei seguenti state diagram lo rappresenta correttamente ?

```
block FSA  //  Finite State Automaton


/* connector declarations outside this block:
connector InputInteger = input Integer;
connector OutputInteger = output Integer;
*/


InputInteger u; // external input
OutputInteger x; // state
parameter Real T = 1;

algorithm

when initial() then
x := 0;

elsewhen sample(0,T) then

if (pre(x) == 0) and (pre(u) == 0) then x := 4;
elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;
elseif (pre(x) == 0) and (pre(u) == 2) then x := 2;
elseif (pre(x) == 1) and (pre(u) == 0) then x := 2;
elseif (pre(x) == 1) and (pre(u) == 1) then x := 0;
elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;
elseif (pre(x) == 2) and (pre(u) == 1) then x := 1;
elseif (pre(x) == 2) and (pre(u) == 2) then x := 0;
elseif (pre(x) == 3) and (pre(u) == 0) then x := 4;
elseif (pre(x) == 3) and (pre(u) == 1) then x := 1;
elseif (pre(x) == 3) and (pre(u) == 2) then x := 0;
elseif (pre(x) == 4) and (pre(u) == 0) then x := 2;
elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;
elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;
else x := pre(x); // default
end if;


end when;
```

end FSA;

Immagine

Immagine

Immagine

Risposta: Immagine

36) 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(int *A, 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 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);}

2.#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);}

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 : 2

37) Si consideri il seguente modello Modelica. Quale dei seguenti state diagram lo rappresenta correttamente ?

```
block FSA  //  Finite State Automaton

/* connector declarations outside this block:
connector InputInteger = input Integer;
connector OutputInteger = output Integer;
*/

InputInteger u; // external input
OutputInteger x; // state
parameter Real T = 1;

algorithm

when initial() then
x := 0;

elsewhen sample(0,T) then

if (pre(x) == 0) and (pre(u) == 0) then x := 1;
elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;
elseif (pre(x) == 1) and (pre(u) == 0) then x := 0;
elseif (pre(x) == 1) and (pre(u) == 1) then x := 0;
elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;
elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;
elseif (pre(x) == 2) and (pre(u) == 1) then x := 3;
elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;
elseif (pre(x) == 3) and (pre(u) == 0) then x := 4;
elseif (pre(x) == 3) and (pre(u) == 1) then x := 4;
elseif (pre(x) == 3) and (pre(u) == 2) then x := 4;
elseif (pre(x) == 4) and (pre(u) == 0) then x := 0;
elseif (pre(x) == 4) and (pre(u) == 1) then x := 2;
elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;
else x := pre(x); // default
end if;
```

end when;

end FSA;

Immagine

Immagine

Immagine

Risposta: Immagine

38) 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 == y)

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

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

Risposta : 2

39) 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=0, y=0}, {x=2, y=1}, {x=2, y=0}.

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

40) 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 act0 act1 act1 act0 act0 act2 act0 act2 act2 act2 act1 act2 act2 act0 act0 act2 act1 act0 act0 act2 act2 act2 act0 act2 act2 act0 act2 act0 act1 act2 act1 act1 act1 act1 act0 act1 act0 act1 act2 act1 act2 act0

Test case 2: act0

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

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

Immagine

1.100%

2.90%

3.50%

Risposta : 2

41) 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

42) 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/(1 - p)

2.(1 - p)/p

3.1/(p*(1 - p))

Risposta : 1

43) 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)));.....}

44) 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 : 1

45) 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)));.....}

46) Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ?

Immagine

1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 2;elseif (pre(x) == 0) and (pre(u) == 2) then x := 2;elseif (pre(x) == 1) and (pre(u) == 0) then x := 4;elseif (pre(x) == 1) and (pre(u) == 1) then x := 0;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 0) then x := 0;elseif (pre(x) == 3) and (pre(u) == 1) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 0) then x := 0;elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

2.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 4;elseif (pre(x) == 0) and (pre(u) == 2) then x := 2;elseif (pre(x) == 1) and (pre(u) == 0) then x := 3;elseif (pre(x) == 1) and (pre(u) == 1) then x := 0;elseif (pre(x) == 1) and (pre(u) == 2) then x := 0;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;elseif (pre(x) == 3) and (pre(u) == 1) then x := 4;elseif (pre(x) == 3) and (pre(u) == 2) then x := 4;elseif (pre(x) == 4) and (pre(u) == 2) then x := 0;else x := pre(x); // defaultend if;end when;end FSA;

3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 2) then x := 2;elseif (pre(x) == 1) and (pre(u) == 0) then x := 4;elseif (pre(x) == 1) and (pre(u) == 2) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 1;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 1) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 4;elseif (pre(x) == 4) and (pre(u) == 0) then x := 0;elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;

Risposta : 3

47) Si consideri il seguente requisito:

RQ: Durante l'esecuzione del programma (cioè per tutti gli istanti di tempo positivi) la variabile x è sempre nell'intervallo [0, 5].

Quale dei seguenti monitor meglio descrive il requisito RQ ?

1.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x > 0) or (x < 5));algorithmwhen edge(z) theny := true;end when;end Monitor;

2.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x > 5) or (x < 0));algorithmwhen edge(z) theny := true;end when;end Monitor;

3.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and (x > 0) and (x < 5);algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 2

48) Un processo software può essere rappesentato 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 probabilita 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 terminti, 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

49) 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 <= 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).

50) 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:

{(-inf, -11], [-10, -1], {0}, [1, 50], [51, +inf)}

Si consideri il seguente insieme di test cases:

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

Quale delle seguenti è la partition coverage conseguita?

1.100%

2.60%

3.80%

Risposta : 2

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 rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del prcoesso 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 probabilita 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'e' la probabilità dello scenario: 1, 3, 4? In altri terminti, 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*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*S)}   (cioè 10 elevato alla (-b*S))

dove b è una opportuna costante note 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 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 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 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 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 Time(X) della sequenza di stati X = x(0), x(1), x(2), ... è Time(X) = time(x(0)) + time(x(1)) + time(x(2)) + ...

Ad esempio se X = 0, 1 abbiamo Time(X) = time(0) + time(1) = time(0) (poichè time(1) = 0).

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

Immagine

1.time(0)*(1 - p)/p

2.time(0)/(1 - p)

3.time(0)/(p*(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 migliorie 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 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 <= 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(int *A, 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 act0 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 boolean più semplici. Ad esempio, (x + y <= 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 >= 100) && (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=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'e' 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 rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del prcoesso 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 probabilita 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'e' la probabilità dello scenario: 1, 2, 3, 4 ? In altri terminti, 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:

{(-inf, -11], [-10, -1], {0}, [1, 50], [51, +inf)}

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*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'e' 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) = if (x > y) then (z == x + 1) else (z == y + 1)

3.F(x, y, z) = if (x > y) then (z == x) 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 rappreenta 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), ... è C(X) = c(x(0)) + c(x(1)) + c(x(2)) + ...

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 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 rappesentato 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 probabilita 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 terminti, 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 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 rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del prcoesso 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 probabilita 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'e' la probabilità dello scenario: 1, 3 ? In altri terminti, 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 nonegativa.

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

1) Quale delle seguenti frasi meglio descrive l'obiettivo del "check di consistenza" che è parte della "requirements validation activity".

1.Assicurarsi che i requisiti funzionali descrivano tutte le funzionalità del sistema.

2.Assicurarsi che non ci siano requisiti in conflitto con altri requisiti.

3.Assicurarsi che per ogni requisito esista un insieme di test che lo possa verificare.

Risposta : 2

2) 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 >= 20) and (x <= 30) ;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 >= 30) or (x <= 20)) ;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 : 2

3) Quale delle seguenti frasi meglio descrive il criterio di "requirements verifiability" che è parte della "requirements validation activity".

1.Per ciascuna coppia di componenti, dovremmo essere in grado di scrivere un insieme di test che può dimostrare che l'interazione tra le componenti soddisfa tutti i requisiti di interfaccia.

2.Per ciascuna componente del sistema, dovremmo essere in grado di scrivere un insieme di test che può dimostrare che essa soddisfa tutti i requisiti.

3.Per ciascun requisito, dovremmo essere in grado di scrivere un inseme di test che può dimostrare che il sistema sviluppato soddisfa il requisito considerato.

Risposta : 3

4) 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)));.....}

5) 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=9, y=0}.

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

Risposta : 2

6) "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 funzionale.

2.Requisito di performance.

3.Requisito non-funzionale.

Risposta : 1

7) Si consideri un software sviluppato seguendo un approccio iterativo implementato con due fasi: F1 seguita da F2. Ciascuna fase ha costo 10000 Eur e deve essere ripetuta una seconda volta con probabilità 0.1. Qual'e' il costo atteso dello sviluppo dell'intero software?

1.22000

2.30000

3.25000

Risposta : 1

8) 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: 90%

2.State coverage: 100%

3.State coverage: 80%

Risposta : 2

9) 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 <= 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 >= 100) && (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=200, b = 0, c = 1), (a=50, b = 5, c = 0), (a=50, b = 3, 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 = 4, c = 0), (a=200, b = 4, c = 0)

Risposta : 1

10) 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: 40%

2.Transition coverage: 70%

3.Transition coverage: 100%

Risposta : 1

11) 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 <= 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 = 0).

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

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

12) 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) = if (x > y) then (z == x) else (z == y + 1)

3.F(x, y, z) = if (x > y) then (z == x + 1) else (z == y + 1)

Risposta : 2

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

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

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

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

Risposta : 1

14) 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:

{(-inf, -11], [-10, -1], {0}, [1, 50], [51, +inf)}

Si consideri il seguente insieme di test cases:

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

Quale delle seguenti è la partition coverage conseguita?

1.80%

2.100%

3.60%

Risposta : 3

15) Si consideri un software sviluppato seguendo un approccio iterativo implementato con due fasi: F1 seguita da F2. Ciascuna fase ha costo 1000. Con probabilità 0.5 potrebbe essere necessario ripetere F1 una seconda volta. Con probabilità 0.2 potrebbe essere necessario ripetere F2 una seconda volta. Qual'e' il costo atteso dello sviluppo dell'intero software?

1.4000

2.5000

3.2700

Risposta : 3

16) 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(int *A, 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 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);}

2.#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);}

3.#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);}

Risposta : 3

17) 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 nonegativa.

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) or (delay(x, 10) > 1) or (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) and (delay(x, 10) > 1) and (y < 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 3

18) Un'azienda decide di organizzare il processo di sviluppo di un grosso software in 101 phasi sequenziali, numerate da 0 a 100. La phase 0 è quella iniziale. La phase 100 è quella finale in cui lo sviluppo è completato. Tutte le fasi hanno circa la stessa durata.

Alla fine di ogni fase viene eseguita una batteria di tests. I risultati del testing possono essere:

a) si può passare alla fase successiva;

b) bisogna ripetere la fase corrente;

c) bisogna rivedere il lavoro fatto nella fase precedente (reworking).

Dai dati storici è noto che la probabilità del caso a) è 0.72, del caso b) è 0.18 e del caso c) è 0.1.

Allo scopo di stimare attraverso una simulazione MonteCarlo il valore atteso del tempo di completamento del progetto viene realizzato un modello Modelica del processo di sviluppo descritto sopra.

Quale dei seguenti modelli Modelica modella correttamente il processo di sviluppo descritto sopra?

1.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() <= 0.9) then if (myrandom() <= 0.8) then x := x + 1; else x := max(0, x - 1); end if; else x := max(0, x - 1); end if;end if;end when;end MarkovChain;

2.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() <= 0.9) then if (myrandom() <= 0.72) then x := x + 1; else x := max(0, x - 1); end if; else x := max(0, x - 1); end if;end if;end when;end MarkovChain;

3.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() <= 0.8) then if (myrandom() <= 0.9) then x := x + 1; else x := max(0, x - 1); end if; else x := max(0, x - 1); end if;end if;end when;end MarkovChain;

Risposta : 1

19) 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

20) Quale delle seguenti frasi meglio descrive l'obiettivo del "check di realismo" (realizability) che è parte della "requirements validation activity".

1.Assicurarsi che le performance richieste al sistema siano necessarie per soddisfare le necessità del customer.

2.Assicurarsi che, tenedo conto della tecnologia, budget e tempo disponibili, sia possibile realizzare un sistema che soddisfa i requisisti.

3.Assicurarsi che le funzionalità richieste al sistema siano necessarie per soddisfare le necessità del customer.

Risposta : 2

21) Quale delle seguenti frasi meglio descrive l'obiettivo del "check di completezza" che è parte della "requirements validation activity".

1.Assicurarsi che i requisisti descrivano tutte le funzionalità e vincoli (e.g., security, performance) del sistema desiderato dal customer.

2.Assicurarsi che i requisiti funzionali descrivano tutte le funzionalità del sistema.

3.Assicurarsi che per ogni requisito sia stato implementato nel sistema.

Risposta : 1

22) Quale delle seguenti frasi è corretta riguardo all'ctivity diagram in figura ?

Immagine

1.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in sequenza.

2.Una volta selezionato il piatto di mare da preparare, la stessa persona prepara prima il pesce e poi il contorno.

3.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in parallelo.

Risposta : 3

23) Si consideri un processo software costituito da due fasi F1 ed F2 ciascuna di costo 10000. Con probabilità p = 0.4 la fase F1 deve essere ripetuta (a causa di change requests) e con probabilità (1 - p) si passa alla fase F2 e poi al completamento (End) dello sviluppo. Qual'è il costo atteso per lo sviluppo del software seguendo il processo sopra descritto ?

1.24000

2.30000

3.35000

Risposta : 1

24) 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

1.60%

2.100%

3.80%

Risposta : 3

25) 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=0, y=0}, {x=2, y=1}, {x=2, y=3}.

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

Risposta : 3

26) 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?

1.80%

2.50%

3.100%

Risposta : 3

27) Si consideri il seguente requisito:

RQ1: Durante l'esecuzione del programma (cioè per tutti gli istanti di tempo positivi) la variabile x è sempre nell'intervallo [0, 5] oppure [10, 15]

Quale dei seguenti monitor meglio descrive il requisito RQ1 ?

1.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and (((x >= 0) and (x <= 5)) or ((x >= 10) and (x <= 15)));algorithmwhen edge(z) theny := true;end when;end Monitor;

2.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x >= 5) or (x <= 0)) and ((x >= 15) or (x <= 10)) ;algorithmwhen edge(z) theny := true;end when;end Monitor;

3.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x >= 0) or (x <= 5)) and ((x >= 10) or (x <= 15)));algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 2

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

1.I metodi agile sono metodi di sviluppo plan-driven.

2.I metodi agile sono metodi di sviluppo incrementale.

3.I metodi agile sono metodi di sviluppo orientato al riuso.

Risposta : 2

29) 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 : 2

30) Si consideri un software sviluppato seguendo un approccio iterativo implementato con due fasi: F1 seguita da F2. Ciascuna fase ha costo 1000 Eur e deve essere ripetuta una seconda volta con probabilità 0.5. Qual'e' il costo atteso dello sviluppo dell'intero software?

1.3000

2.5000

3.2000

Risposta : 1

31) Il rischio R può essere calcolato come R = P*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 = exp(-b*S), dove b è una opportuna costante note da dati storici aziendali. Si assuma che b = 0.00001, C = 1000000, ed il rischio ammesso è R = 1000. Quale delle seguenti opzioni meglio approssima il costo S per lo sviluppo del software in questione.

1.700000 EUR

2.300000 EUR

3.500000 EUR

Risposta : 1

32) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?

Immagine

1.Una volta selezionata la bevanda non è possibile cancellare l'operazione.

2.La macchina non dà resto.

3.Una volta inserite monete per due bevande è possibile ottenerle senza reinserire le monete.

Risposta : 1

33) 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 era stata richiesta una risorsa (variabile x positiva) allora ora è concesso l'accesso alla risorsa (variabile y positiva)

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

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 act0 act2 act2 act2

Test case 2: act0 act2 act2 act1 act2 act1 act1 act1 act2 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: 100%

3.Transition coverage: 70%

Risposta : 3

35) 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?

1.50%

2.100%

3.80%

Risposta : 2

36) 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

37) Si consideri un software sviluppato seguendo un approccio iterativo implementato con due fasi: F1 seguita da F2. Ciascuna fase ha costo 10000. Con probabilità 0.1 potrebbe essere necessario ripetere F1 una seconda volta. Con probabilità 0.2 potrebbe essere necessario ripetere F2 una seconda volta. Qual'e' il costo atteso dello sviluppo dell'intero software?

1.25000

2.30000

3.23000

Risposta : 3

38) Quale delle seguenti frasi meglio descrive l'obiettivo del "validity check" che è parte della "requirements validation activity".

1.Assicurarsi che i requisiti funzionali descrivano tutte le funzionalità del sistema.

2.Assicurarsi che un sistema che soddisfa i requisiti risolve il problema del "customer".

3.Assicurarsi che non ci siano requisiti in conflitto con altri requisiti.

Risposta : 2

39) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?

Immagine

Immagine

Immagine

Immagine

Risposta: Immagine

40) Si consideri un processo software costituito da due fasi F1 ed F2 ciascuna di costo 10000. Con probabilità p = 0.1 la fase F1 deve essere ripetuta (a causa di change requests) e con probabilità (1 - p) si passa alla fase F2 e poi al completamento (End) dello sviluppo. Qual'è il costo atteso per lo sviluppo del software seguendo il processo sopra descritto ?

1.25000

2.30000

3.21000

Risposta : 3

41) Il rischio R può essere calcolato come R = P*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 = exp(-b*S), dove b è una opportuna costante note da dati storici aziendali. Si assuma che b = 0.00001, C = 1000000, ed il rischio ammesso è R = 100. Quale delle seguenti opzioni meglio approssima il costo S per lo sviluppo del software in questione.

1.750000 EUR

2.950000 EUR

3.850000 EUR

Risposta : 2

42) 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: 100%

2.State coverage: 90%

3.State coverage: 70%

Risposta : 2

43) Lo state diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti modelli Modelica è plausibile per lo state diagram in figura?

Immagine

1.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer := 0; elsewhen sample(0, T) thenif (pre(state) == 0) and (Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state := 3; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 0; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end CoffeeMachine;

2.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer := 0; elsewhen sample(0, T) thenif (pre(state) == 0) and (Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state := 3; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 3; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end CoffeeMachine;

3.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer := 0; elsewhen sample(0, T) thenif (pre(state) == 0) and (Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state := 0; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 3; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end CoffeeMachine;

Risposta : 2

44) Quale delle seguenti frasi è corretta riguardo al Sequence Diagram in figura ?

Immagine

1.Il paziente richiede al server una visita con uno specifico medico e, dopo una verifica sul database, riceve conferma dal server della disponibilità o meno del medico richiesto.

2.Periodicamente il client comunica ai pazienti le disponibilità dei medici.

3.Il paziente richiede al client una visita con uno specifico medico e, dopo una verifica sul database, riceve conferma dal client della disponibilità o meno del medico richiesto.

Risposta : 3

45) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?

Immagine

1.Anche se ci sono abbastanza monete potrebbe non essere possibile ottenere la bevanda selezionata.

2.Se ci sono abbastanza monete è sempre possibile ottenere la bevanda selezionata.

3.Una volta inserite le monete bisogna necessariamente consumare almeno una bevanda.

Risposta : 2

46) 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 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: 75%

2.State coverage: 100%

3.State coverage: 80%

Risposta : 1

47) 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: 100%

2.State coverage: 90%

3.State coverage: 70%

Risposta : 2

48) 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) and (delay(x, 10) <= 0) 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 > 60) or (delay(x, 10) > 0) or (y >= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 1

49) 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 = -150, x = 0, x = 100, x = 700}

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

3.{x = -200, x = -50, x = 0, x = 100, x = 500}

Risposta : 2

50) 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

1) 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 il seguente programma C:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#define N  4   /* number of test cases */


int f(int x1, int x2)
{
 if (x1 + x2 <= 2)
 return (1);
 else return (2);
}


int main() {   int  i, y;   int x1[N], x2[N];
 // define test cases
   x1[0] = 3; x2[0] = -2;    x1[1] = 4; x2[1] = -3;    x1[2] = 7; x2[2] = -4;    x1[3] = 8; x2[3] = -5;
   // testing
 for (i = 0; i < N; i++) {
     y = f(x1[i], x2[i]);       // function under testing
     assert(y ==(x1[i], x2[i] <= 2) ? 1 : 2);   // oracle
   }
  printf("All %d test cases passed\n", N);
   return (0);
}
```

Il programma main() sopra realizza il nostro testing per la funzione f1(). I test cases sono i valori in x1[i] ed x2[i].

Quale delle seguenti è la branch coverage conseguita?

1.100%

2.50%

3.80%

Risposta : 1

2) 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) or ((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) and (x >= 20) and (x <= 30) ;algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 1

3) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?

Immagine

1.Una volta selezionata la bevanda non è possibile cancellare l'operazione.

2.Una volta inserite monete per due bevande è possibile ottenerle senza reinserire le monete.

3.La macchina non dà resto.

Risposta : 1

4) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?

Immagine

Immagine

Immagine

Immagine

Risposta: Immagine

5) 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) rggiunti almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2

2) Start PIN validation, card inserted, PIN Entered, Cancel 2, End PIN Validation 2

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

Immagine

1.60%

2.80%

3.90%

Risposta : 3

6) Si consideri il monitor seguente che ritorna true appena il sistema viola il requisito monitorato.

block Monitor

input Real x;

output Boolean y;

Boolean w;

initial equation

y = false;

equation

w = ((x < 1) or (x > 4)) and ((x < 15) or (x > 20));

algorithm

when edge(w) then

y := true;

end when;

end Monitor;

Quale delle seguenti affermazioni meglio descrive il requisito monitorato?

1.La variabile x è fuori dall'intervallo [1, 4] e fuori dall'intervallo [15, 20].

2.La variabile x è nell'intervallo [1, 4] e fuori dall'intervallo [15, 20].

3.La variabile x è nell'intervallo [1, 4] oppure nell'intervallo [15, 20].

Risposta : 3

7) Si consideri il seguente requisito:

RQ1: Durante l'esecuzione del programma (cioè per tutti gli istanti di tempo positivi) la variabile x è sempre nell'intervallo [0, 5] oppure [10, 15]

Quale dei seguenti monitor meglio descrive il requisito RQ1 ?

1.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x >= 5) or (x <= 0)) and ((x >= 15) or (x <= 10)) ;algorithmwhen edge(z) theny := true;end when;end Monitor;

2.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and (((x >= 0) and (x <= 5)) or ((x >= 10) and (x <= 15)));algorithmwhen edge(z) theny := true;end when;end Monitor;

3.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x >= 0) or (x <= 5)) and ((x >= 10) or (x <= 15)));algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 1

8) Si consideri il seguente requisito:

RQ: Durante l'esecuzione del programma (cioè per tutti gli istanti di tempo positivi) la variabile x è sempre nell'intervallo [0, 5].

Quale dei seguenti monitor meglio descrive il requisito RQ ?

1.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and (x > 0) and (x < 5);algorithmwhen edge(z) theny := true;end when;end Monitor;

2.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x > 0) or (x < 5));algorithmwhen edge(z) theny := true;end when;end Monitor;

3.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x > 5) or (x < 0));algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 3

9) Il system testing si concentra su:

1.Testare le funzionalità di unità software individuali, oggetti, classi o metodi.

2.Testare l'interazione tra le componenti del sistema (cioè, integrazione di molte unità di sistema).

3.Testare le interfacce per ciascuna componente.

Risposta : 2

10) 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 : 1

11) La transition coverage di un insieme di test cases (cioè sequeze 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:

1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2

2) Start PIN validation, card inserted, PIN Entered, Cancel 2, End PIN Validation 2

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

Immagine

1.80%

2.60%

3.40%

Risposta : 2

12) 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 era stata richiesta una risorsa (variabile x positiva) allora ora è concesso l'accesso alla risorsa (variabile y positiva)

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) and (delay(x, 10) > 0) 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 > 60) or (delay(x, 10) > 0) or (y <= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 2

13) 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) rggiunti almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2

2) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 2, End PIN Validation 2

3) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, More than 3 failed..., END PIN validation 1;

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

Immagine

1.100%

2.90%

3.80%

Risposta : 1

14) Un'azienda decide di organizzare il processo di sviluppo di un grosso software in 101 phasi sequenziali, numerate da 0 a 100. La phase 0 è quella iniziale. La phase 100 è quella finale in cui lo sviluppo è completato. Tutte le fasi hanno circa la stessa durata.

Alla fine di ogni fase viene eseguita una batteria di tests. I risultati del testing possono essere:

a) si può passare alla fase successiva;

b) bisogna ripetere la fase corrente;

c) bisogna rivedere il lavoro fatto nella fase precedente (reworking).

Dai dati storici è noto che la probabilità del caso a) è 0.72, del caso b) è 0.18 e del caso c) è 0.1.

Allo scopo di stimare attraverso una simulazione MonteCarlo il valore atteso del tempo di completamento del progetto viene realizzato un modello Modelica del processo di sviluppo descritto sopra.

Quale dei seguenti modelli Modelica modella correttamente il processo di sviluppo descritto sopra?

1.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() <= 0.9) then if (myrandom() <= 0.72) then x := x + 1; else x := max(0, x - 1); end if; else x := max(0, x - 1); end if;end if;end when;end MarkovChain;

2.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() <= 0.9) then if (myrandom() <= 0.8) then x := x + 1; else x := max(0, x - 1); end if; else x := max(0, x - 1); end if;end if;end when;end MarkovChain;

3.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() <= 0.8) then if (myrandom() <= 0.9) then x := x + 1; else x := max(0, x - 1); end if; else x := max(0, x - 1); end if;end if;end when;end MarkovChain;

Risposta : 2

15) La transition coverage di un insieme di test cases (cioè sequeze 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:

1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2;

2) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2;

3) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2.

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

Immagine

1.40%

2.80%

3.60%

Risposta : 3

16) Quale delle seguenti affermazioni è vera riguardo al performance testing?

1.Il performance testing è tipicamente eseguito su un prototipo del sistema.

2.Il performance testing è tipicamente eseguito una volta che il sistema è stato completamento integrato.

3.Il performance testing è tipicamente eseguito solo sulle componenti del sistema prima dell'integrazione.

Risposta : 2

17) Si consideri il Test-Driven Development (TDD). Quale delle seguenti affermazioni è vera?

1.Scrivi test automatizzati per tutti i requisiti di sistema, esegui i test e rivedi l'implementazione come necessario.

2.Per ciascun incremento di funzionalità, scrivi test automatizzati, implementa la funzionalità, esegui i test e rivedi l'implementazione come necessario.

3.Per ciascun incremento di funzionalità, implementa la funzionalità, scrivi test automatizzati, esegui i test e rivedi l'implementazione come necessario.

Risposta : 2

18) 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 il seguente programma C:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#define N  4    /* number of test cases */

int f(int x1, int x2)
{
  if (x1 + x2 <= 2)
  return (1);
  else return (2);
}

int main() {    int  i, y;   int x1[N], x2[N];
  // define test cases
    x1[0] = 5; x2[0] = -2;    x1[1] = 6; x2[1] = -3;    x1[2] = 7; x2[2] = -4;    x1[3] = 8; x2[3] = -5;
    // testing
  for (i = 0; i < N; i++)  {
      y = f(x1[i], x2[i]);       // function under testing
      assert(y ==(x1[i], x2[i] <= 2) ? 1 : 2);   // oracle
    }
   printf("All %d test cases passed\n", N);
    return (0);
}
```

Il programma main() sopra realizza il nostro testing per la funzione f1(). I test cases sono i valori in x1[i] ed x2[i].

Quale delle seguenti è la branch coverage conseguita?

1.50%

2.80%

3.100%

Risposta : 1

19) 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 il seguente programma C:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#define N  1   /* number of test cases */
int f(int x)  {   int y = 0;
  LOOP:  if (abs(x) - y <= 2)
                    {return   ;}
               else {y = y + 1; goto LOOP;}
}  /* f()  */
int main() {  int  i, y;   int x[N];
// define test cases
    x[0] = 3;
// testing
  for (i = 0; i < N; i++)   {
      y = f(x[i]);       // function under testing
      assert(y == (abs(x[i]) <= 2) ? 0 : (abs(x[i]) - 2));   // oracle
    }
```

```
  printf("All %d test cases passed\n", N);
 return (0);
}
```

Il programma main() sopra realizza il nostro testing per la funzione f(). I test cases sono i valori in x1[i] ed x2[i].

Quale delle seguenti è la branch coverage conseguita?

Immagine

1.80%

2.100%

3.50%

Risposta : 2

20) 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 il seguente programma C:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#define N  4    /* number of test cases */


int f(int x1, int x2)
{
 if (x1 + x2 <= 2)
 return (1);
 else return (2);
}


int main() {    int  i, y;   int x1[N], x2[N];
 // define test cases
   x1[0] = 3; x2[0] = -2;    x1[1] = 4; x2[1] = -3;    x1[2] = 5; x2[2] = -4;    x1[3] = 6; x2[3] = -5;
   // testing
 for (i = 0; i < N; i++)  {
     y = f(x1[i], x2[i]);       // function under testing
    assert(y ==(x1[i], x2[i] <= 2) ? 1 : 2);   // oracle
   }
  printf("All %d test cases passed\n", N);
   return (0);
}
```

Il programma main() sopra realizza il nostro testing per la funzione f1(). I test cases sono i valori in x1[i] ed x2[i].

Quale delle seguenti è la branch coverage conseguita?

1.80%

2.100%

3.50%

Risposta : 3

21) L'environment di un sistema software è costituito da uno user che, ogni untià di tempo (ad esempio, un secondo) invia al sistema tre numeri: -1, 0, 1, con probabilità, rispettivamente, 0.2, 0.56, 0.24.

Quale dei seguenti modelli Modelica modella correttamente l'environment descritto sopra.

1.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) then if (myrandom() <= 0.7) then if (myrandom() <= 0.8) then x := 0; else x := 1; end if; else x := -1; end if;end when;end MarkovChain;

2.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) then if (myrandom() <= 0.8) then if (myrandom() <= 0.7) then x := 0; else x := 1; end if; else x := -1; end if;end when;end MarkovChain;

3.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;OutputInteger x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) then if (myrandom() <= 0.8) then if (myrandom() <= 0.7) then x := 1; else x := 0; end if; else x := -1; end if;end when;end MarkovChain;

Risposta : 2

22) Quale delle seguenti frasi è corretta riguardo all'ctivity diagram in figura ?

Immagine

1.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in sequenza.

2.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in parallelo.

3.Una volta selezionato il piatto di mare da preparare, la stessa persona prepara prima il pesce e poi il contorno.

Risposta : 2

23) Quale delle seguenti frasi è corretta riguardo al Sequence Diagram in figura ?

Immagine

1.Il paziente richiede al server una visita con uno specifico medico e, dopo una verifica sul database, riceve conferma dal server della disponibilità o meno del medico richiesto.

2.Periodicamente il client comunica ai pazienti le disponibilità dei medici.

3.Il paziente richiede al client una visita con uno specifico medico e, dopo una verifica sul database, riceve conferma dal client della disponibilità o meno del medico richiesto.

Risposta : 3

24) Quale delle seguenti frasi meglio descrive il criterio di "requirements verifiability" che è parte della "requirements validation activity".

1.Per ciascuna componente del sistema, dovremmo essere in grado di scrivere un insieme di test che può dimostrare che essa soddisfa tutti i requisiti.

2.Per ciascuna coppia di componenti, dovremmo essere in grado di scrivere un insieme di test che può dimostrare che l'interazione tra le componenti soddisfa tutti i requisiti di interfaccia.

3.Per ciascun requisito, dovremmo essere in grado di scrivere un inseme di test che può dimostrare che il sistema sviluppato soddisfa il requisito considerato.

Risposta : 3

25) 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 il seguente programma C:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#define N  5    /* number of test cases */
int f1(int x)  { return (2*x); }
int main() {  int  i, y;    int x[N];
 // define test cases
  x[0] = 0; x[1] = 1; x[2] = -1; x[3] = 10; x[4] = -10;
// testing
for (i = 0; i < N; i++)   {
      y = f1(x[i]);       // function under testing
      assert(y == 2*x[i]);   // oracle
 }
   printf("All %d test cases passed\n", N);
    return (0);
}
```

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

{0, {-1}, {1}, {tutti glli interi negativi diversi da -1}, {tutti glli interi positivi diversi da 1}}

Il programma main() sopra realizza il nostro testing per la funzione f1(). I test cases sono i valori in x[i].

Quale delle seguenti è la partition coverage conseguita?

1.80%

2.100%

3.50%

Risposta : 2

26) 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 il seguente programma C:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#define N  5    /* number of test cases */
int f1(int x)  { return (2*x); }
int main() {  int  i, y;    int x[N];
```

```
// define test cases
 x[0] = 0; x[1] = 1; x[2] = -1; x[3] = 10; x[4] = -10;
// testing
for (i = 0; i < N; i++)  {
    y = f1(x[i]);      // function under testing
    assert(y == 2*x[i]);   // oracle
 }
  printf("All %d test cases passed\n", N);
   return (0);
}
```

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

{(-inf, -21], [-20, -1], {0}, [1, 20], [21, +inf)}

Il programma main() sopra realizza il nostro testing per la funzione f1(). I test cases sono i valori in x[i].

Quale delle seguenti è la partition coverage conseguita?

1.100%

2.80%

3.60%

Risposta : 3

27) Lo state diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti modelli Modelica è plausibile per lo state diagram in figura?

Immagine

1.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer := 0; elsewhen sample(0, T) thenif (pre(state) == 0) and (Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state := 3; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 0; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end CoffeeMachine;

2.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer := 0; elsewhen sample(0, T) thenif (pre(state) == 0) and (Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state := 0; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 3; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end CoffeeMachine;

3.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer := 0; elsewhen sample(0, T) thenif (pre(state) == 0) and (Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state := 3; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 3; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end CoffeeMachine;

Risposta : 3

28) Quale delle seguenti frasi meglio descrive l'obiettivo del "check di realismo" (realizability) che è parte della "requirements validation activity".

1.Assicurarsi che le performance richieste al sistema siano necessarie per soddisfare le necessità del customer.

2.Assicurarsi che le funzionalità richieste al sistema siano necessarie per soddisfare le necessità del customer.

3.Assicurarsi che, tenedo conto della tecnologia, budget e tempo disponibili, sia possibile realizzare un sistema che soddisfa i requisisti.

Risposta : 3

29) 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

30) L'input di un sistema software è costituito da un sensore che ogni unità di tempo (ad esempio, un secondo) invia un numero reale. Con probabilità 0.63 il valore inviato in una unità di tempo è maggiore del 10% rispetto quello inviato nell'unità di tempo precedente. Con probabilità 0.1 è inferiore del 27% rispetto al valore inviato nell'unità di tempo precedente. Con probabilità 0.27 è inferiore del 10% rispetto quello inviato nell'unità di tempo precedente.

Quale dei seguenti modelli Modelica modella correttamente l'environment descritto sopra.

1.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Real x0 = 1;OutputReal x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (myrandom() <= 0.9)then if (myrandom() <= 0.7) then x := 1.1*x; else x := 0.9*x; end if;else x := 0.73*x; end if;end when;end MarkovChain;

2.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Real x0 = 1;OutputReal x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (myrandom() <= 0.9)then if (myrandom() <= 0.7) then x := 0.9*x; else x := 01.1*x; end if;else x := 0.73*x; end if;end when;end MarkovChain;

3.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Real x0 = 1;OutputReal x;algorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (myrandom() <= 0.7)then if (myrandom() <= 0.9) then x := 1.1*x; else x := 0.9*x; end if;else x := 0.73*x; end if;end when;end MarkovChain;

Risposta : 1

31) Quale delle seguenti frasi meglio descrive l'obiettivo del "validity check" che è parte della "requirements validation activity".

1.Assicurarsi che un sistema che soddisfa i requisiti risolve il problema del "customer".

2.Assicurarsi che i requisiti funzionali descrivano tutte le funzionalità del sistema.

3.Assicurarsi che non ci siano requisiti in conflitto con altri requisiti.

Risposta : 1

32) Quale delle seguenti frasi meglio descrive l'obiettivo del "check di completezza" che è parte della "requirements validation activity".

1.Assicurarsi che per ogni requisito sia stato implementato nel sistema.

2.Assicurarsi che i requisiti funzionali descrivano tutte le funzionalità del sistema.

3.Assicurarsi che i requisisti descrivano tutte le funzionalità e vincoli (e.g., security, performance) del sistema desiderato dal customer.

Risposta : 3

33) Un'azienda decide di organizzare il processo di sviluppo di un grosso software in 101 phasi sequenziali, numerate da 0 a 100. La phase 0 è quella iniziale. La phase 100 è quella finale in cui lo sviluppo è completato. Tutte le fasi hanno circa la stessa durata.

Si decide di realizzare un approccio incrementale in cui, alla fine di ogni fase, si passa alla fase successiva solo nel caso in cui tutti i test per la fase vengono superati. In caso contrario bisogna ripetere la phase. Dai dati storici è noto che la probabilità che il team di sviluppo passi da una fase a quella successiva è 0.8.

Allo scopo di stimare attraverso una simulazione MonteCarlo il valore atteso del tempo di completamento del progetto viene realizzato un modello Modelica delo processo di sviluppo descritto sopra.

Quale dei seguenti modelli Modelica modella correttamente il processo di sviluppo descritto sopra?

1.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x; // Connectoralgorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() <= 0.8) then x := x + 1; else x := x - 1; end if;end if;end when;end MarkovChain;

2.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x; // Connectoralgorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() <= 0.8) then x := x + 1; end if;end if;end when;end MarkovChain;

3.block MarkovChain//external function myrandom() returns a random real number in [0, 1]parameter Integer x0 = 0;parameter Integer xmax = 100;OutputInteger x; // Connectoralgorithmwhen initial() thenx := x0;elsewhen sample(0, 1) thenif (x < xmax)then if (myrandom() >= 0.8) then x := x + 1; end if;end if;end when;end MarkovChain;

Risposta : 2

34) Quali delle seguenti attività può contribuire a validare i requisiti di un sistema ?

1.Costruire un prototipo e valutarne attentamente le performance.

2.Costruire un prototipo, metterlo in esercizio ed accertarsi che i porti i benefici attesi.

3.Costruire un prototipo e testarlo a fondo per evidenziare subito errori di implementazione.

Risposta : 2

35) Una azienda manifatturiera desidera costruire un sistema software per monitorare (attraverso sensori) la produzione al fine di ridurre gli scarti. Quali delle seguenti attività contribuisce a validare i requisiti del sistema.

1.Costruire un prototipo, eseguirlo usando dati storici dai log di produzione e valutarne le performance.

2.Costruire un prototipo, eseguirlo usando dati storici dai log di produzione ed identificare errori di implementazione.

3.Costruire un prototipo, eseguirlo usando dati storici dai log di produzione e valutare la capacità del prototipo di ridurre gli scarti.

Risposta : 3

36) Il component testing si concentra su:

1.Testare le interfacce per ciascun componente.

2.Testare funzionalità di unità software individuali, oggetti, classi o metodi.

3.Testare l'interazione tra molte componenti (cioè integrazione di molte unità).

Risposta : 1

37) 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 nonegativa.

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) or (delay(x, 10) > 1) 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 > 40) and (delay(x, 10) > 1) and (y >= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 1

38) 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) rggiunti almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2;

2) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2;

3) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2.

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

Immagine

1.80%

2.60%

3.90%

Risposta : 3

39) 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) or (x > 20)) and ((y < 0.5*x) or (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) and (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) and (y <= 0.7*x) ;algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 2

40) 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 <= 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 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 in 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, bool b, bool c)
{   if ( (a == 100) && (b || c) )
      { return (1); }
    else { return (2);}
}
```

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

1.(a=100, b=false, c=true), (a=90, b=false, c=true)

2.(a=100, b=false, c=false), (a=90, b=true, c=true)

3.(a=100, b=false, c=true), (a=90, b=true, c=false)

Risposta : 3

41) La validazione risponde alla seguenete domanda:

1.Sono soddisfatti i requisti funzionali ?

2.Stiamo costruendo il sistema giusto ?

3.Stiamo costruendo il sistema nel modo giusto ?

Risposta : 2

42) "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 di performance.

2.Requisito non-funzionale.

3.Requisito funzionale.

Risposta : 3

43) Quale delle seguenti frasi meglio descrive l'obiettivo del "check di consistenza" che è parte della "requirements validation activity".

1.Assicurarsi che i requisiti funzionali descrivano tutte le funzionalità del sistema.

2.Assicurarsi che per ogni requisito esista un insieme di test che lo possa verificare.

3.Assicurarsi che non ci siano requisiti in conflitto con altri requisiti.

Risposta : 3

44) 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 <= 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 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 in 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, bool b, bool c)
{    if ( (a == 100) && b )
         return (1);    // punto di uscita 1
     else if (b || c)
         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=100, b=true, c=false), (a=90, b=false, c=false), (a=100, b=false, c=false)

2.(a=100, b=true, c=false), (a=90, b=false, c=true), (a=90, b=false, c=false)

3.(a=100, b=true, c=false), (a=90, b=false, c=true), (a=100, b=true, c=true)

Risposta : 2

45) Si consideri il monitor seguente che ritorna true appena i requisiti per il sistema monitorato sono violati.

```
block Monitor
input Real x;
output Boolean y;
Boolean w;
initial equation
y = false;
equation
w = ((x < 0) or (x > 5));
algorithm
when edge(w) then
y := true;
end when;
end Monitor;
```

Quale delle seguenti affermazioni meglio descrive il requisito monitorato.

1.La variabile x è fuori dall'intervallo [0, 5].

2.La variabile x è nell'intervallo [0, 5].

3.La variable x è minore di 0.

Risposta : 2

46) 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

47) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?

Immagine

1.Anche se ci sono abbastanza monete potrebbe non essere possibile ottenere la bevanda selezionata.

2.Una volta inserite le monete bisogna necessariamente consumare almeno una bevanda.

3.Se ci sono abbastanza monete è sempre possibile ottenere la bevanda selezionata.

Risposta : 3

48) 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 prototipo del sistema e valutarne i requisiti non funzionali usando i dati storici dall'azienda.

2.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 migliorie apportate dal sistema software ai processi di business dell'azienda mediante simulazione.

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

Risposta : 2

49) La transition coverage di un insieme di test cases (cioè sequeze 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:

1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2

2) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 2, End PIN Validation 2

3) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, More than 3 failed..., END PIN validation 1;

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

Immagine

1.80%

2.90%

3.100%

Risposta : 2

50) Unit testing si concentra su:

1.Testare l'interazione tra componenti.

2.Testare le interfacce di ciascuna componente.

3.Testare funzionalità di unità software individuali, oggetti, classi o metodi.

Risposta : 3

1) 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) rggiunti almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2

2) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 2, End PIN Validation 2

3) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, More than 3 failed..., END PIN validation 1;

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

Immagine

1.100%

2.90%

3.80%

Risposta : 1

2) La validazione risponde alla seguenete domanda:

1.Stiamo costruendo il sistema nel modo giusto ?

2.Sono soddisfatti i requisti funzionali ?

3.Stiamo costruendo il sistema giusto ?

Risposta : 3

3) La transition coverage di un insieme di test cases (cioè sequeze 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:

1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2;

2) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2;

3) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2.

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

Immagine

1.40%

2.60%

3.80%

Risposta : 2

4) 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 il seguente programma C:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#define N  5    /* number of test cases */
int f1(int x)  { return (2*x); }
int main() {  int  i, y;    int x[N];
 // define test cases
  x[0] = 0; x[1] = 1; x[2] = -1; x[3] = 10; x[4] = -10;
// testing
for (i = 0; i < N; i++)   {
      y = f1(x[i]);       // function under testing
      assert(y == 2*x[i]);   // oracle
 }
   printf("All %d test cases passed\n", N);
    return (0);
}
```

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

{(-inf, -21], [-20, -1], {0}, [1, 20], [21, +inf)}

Il programma main() sopra realizza il nostro testing per la funzione f1(). I test cases sono i valori in x[i].

Quale delle seguenti è la partition coverage conseguita?

1.80%

2.60%

3.100%

Risposta : 2

5) 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) rggiunti almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2

2) Start PIN validation, card inserted, PIN Entered, Cancel 2, End PIN Validation 2

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

Immagine

1.90%

2.60%

3.80%

Risposta : 1

6) 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'e' il costo dello sviluppo usando un approccio plan-driven come sopra ?

1.A + 2*B

2.3*A + 6*B

3.3*A + 3*B

Risposta : 2

7) Quale pattern architetturale meglio descrive l'architettura in figura ?

Immagine

1.Layred architecture.

2.Pipe and filter architecture.

3.Model View Controller.

Risposta : 1

8) Quale delle seguenti frasi è corretta riguardo all'ctivity diagram in figura ?

Immagine

1.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in parallelo.

2.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in sequenza.

3.Una volta selezionato il piatto di mare da preparare, la stessa persona prepara prima il pesce e poi il contorno.

Risposta : 1

9) Quale delle seguenti frasi meglio descrive il criterio di "requirements verifiability" che è parte della "requirements validation activity".

1.Per ciascuna componente del sistema, dovremmo essere in grado di scrivere un insieme di test che può dimostrare che essa soddisfa tutti i requisiti.

2.Per ciascun requisito, dovremmo essere in grado di scrivere un inseme di test che può dimostrare che il sistema sviluppato soddisfa il requisito considerato.

3.Per ciascuna coppia di componenti, dovremmo essere in grado di scrivere un insieme di test che può dimostrare che l'interazione tra le componenti soddisfa tutti i requisiti di interfaccia.

Risposta : 2

10) Quale delle seguenti affermazioni è vera riguardo al performance testing?

1.Il performance testing è tipicamente eseguito su un prototipo del sistema.

2.Il performance testing è tipicamente eseguito una volta che il sistema è stato completamento integrato.

3.Il performance testing è tipicamente eseguito solo sulle componenti del sistema prima dell'integrazione.

Risposta : 2

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) or (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) and (x <= 20) and (y >= 0.5*x) and (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) or (x > 20)) and ((y < 0.5*x) or (y > 0.7*x)) ;algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 1

12) Quale delle seguenti frasi è corretta riguardo al Sequence Diagram in figura ?

Immagine

1.Il paziente richiede al client una visita con uno specifico medico e, dopo una verifica sul database, riceve conferma dal client della disponibilità o meno del medico richiesto.

2.Periodicamente il client comunica ai pazienti le disponibilità dei medici.

3.Il paziente richiede al server una visita con uno specifico medico e, dopo una verifica sul database, riceve conferma dal server della disponibilità o meno del medico richiesto.

Risposta : 1

13) 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 prototipo del sistema e testarlo rispetto ai requisiti funzionali usando i dati storici dall'azienda.

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

3.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 migliorie apportate dal sistema software ai processi di business dell'azienda mediante simulazione.

Risposta : 3

14) Un processo di sviluppo plan-driven consiste di 2 fasi F1, F2, ciascuna costo A. Alla fine di ogni fase vengono prese in considerazione le "change requests" e, se ve ne sono, lo sviluppo viene ripetuto a partire dalla prima iterazione. Quindi con nessuna change request si hanno le fasi: F1, F2 e costo 2A. Con una "change request" dopo la prima fase si ha: F1, F1, F2 e costo 3A. Con una change request dopo la fase 2 si ha: F1, F2, F1, F2 e costo 4A. Qual'è il costo nel caso in cui ci siano change requests sia dopo la fase 1 che dopo la fase 2.

1.5*A

2.6*A

3.7*A

Risposta : 1

15) Si consideri il monitor seguente che ritorna true appena i requisiti per il sistema monitorato sono violati.

```
block Monitor
input Real x;
output Boolean y;
Boolean w;
initial equation
y = false;
equation
w = ((x < 0) or (x > 5));
algorithm
when edge(w) then
y := true;
end when;
end Monitor;
```

Quale delle seguenti affermazioni meglio descrive il requisito monitorato.

1.La variabile x è fuori dall'intervallo [0, 5].

2.La variable x è minore di 0.

3.La variabile x è nell'intervallo [0, 5].

Risposta : 3

16) 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) and (delay(x, 10) > 0) 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 > 60) or (delay(x, 10) > 0) or (y >= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 2

17) 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 il seguente programma C:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#define N  4    /* number of test cases */
```

```
int f(int x1, int x2)
{
 if (x1 + x2 <= 2)
 return (1);
 else return (2);
}

int main() {   int  i, y;   int x1[N], x2[N];
 // define test cases
   x1[0] = 5; x2[0] = -2;   x1[1] = 6; x2[1] = -3;   x1[2] = 7; x2[2] = -4;   x1[3] = 8; x2[3] = -5;
   // testing
 for (i = 0; i < N; i++)  {
     y = f(x1[i], x2[i]);       // function under testing
     assert(y ==(x1[i], x2[i] <= 2) ? 1 : 2);   // oracle
   }
  printf("All %d test cases passed\n", N);
   return (0);
}
```

Il programma main() sopra realizza il nostro testing per la funzione f1(). I test cases sono i valori in x1[i] ed x2[i].

Quale delle seguenti è la branch coverage conseguita?

1.80%

2.50%

3.100%

Risposta : 2

18) La transition coverage di un insieme di test cases (cioè sequeze 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:

1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2

2) Start PIN validation, card inserted, PIN Entered, Cancel 2, End PIN Validation 2

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

Immagine

1.40%

2.80%

3.60%

Risposta : 3

19) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?

Immagine

Immagine

Immagine

Immagine

Risposta: Immagine

20) Verification answers the following question:

1.Is the system cost reasonable for the intended market ?

2.Are we building the right system??

3.Are we building the system right??

21) 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 il seguente programma C:

```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <assert.h>
#define N  5    /* number of test cases */
int f1(int x)  { return (2*x); }
int main() {  int  i, y;    int x[N];
 // define test cases
  x[0] = 0; x[1] = 1; x[2] = -1; x[3] = 10; x[4] = -10;
// testing
for (i = 0; i < N; i++)   {
      y = f1(x[i]);       // function under testing
      assert(y == 2*x[i]);   // oracle
 }
   printf("All %d test cases passed\n", N);
    return (0);
}
```

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

{0, {-1}, {1}, {tutti glli interi negativi diversi da -1}, {tutti glli interi positivi diversi da 1}}

Il programma main() sopra realizza il nostro testing per la funzione f1(). I test cases sono i valori in x[i].

Quale delle seguenti è la partition coverage conseguita?

1.80%

2.50%

3.100%

Risposta : 3

22) Si consideri il seguente requisito:

RQ: Durante l'esecuzione del programma (cioè per tutti gli istanti di tempo positivi) la variabile x è sempre nell'intervallo [0, 5].

Quale dei seguenti monitor meglio descrive il requisito RQ ?

1.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x > 0) or (x < 5));algorithmwhen edge(z) theny := true;end when;end Monitor;

2.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and (x > 0) and (x < 5);algorithmwhen edge(z) theny := true;end when;end Monitor;

3.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x > 5) or (x < 0));algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 3

23) Un processo di sviluppo agile consiste di varie iterazioni. Alla fine di ogni iterazione vengono prese in considerazione le "change requests" e, se ve ne sono, l'iterazione viene ripetuta. Sia p la probabilità che ci siano "change requests" all fine di una iterazione e sia A il costo di una iterazione. Il valore atteso del costo per l'iterazione è:

1.p*A

2.A

3.(1 + p)*A

Risposta : 3

24) Si consideri il monitor seguente che ritorna true appena il sistema viola il requisito monitorato.

```
block Monitor
input Real x;
output Boolean y;
Boolean w;
initial equation
y = false;
equation
w = ((x < 1) or (x > 4)) and ((x < 15) or (x > 20));
algorithm
when edge(w) then
y := true;
end when;
end Monitor;
```

Quale delle seguenti affermazioni meglio descrive il requisito monitorato?

1.La variabile x è nell'intervallo [1, 4] oppure nell'intervallo [15, 20].

2.La variabile x è nell'intervallo [1, 4] e fuori dall'intervallo [15, 20].

3.La variabile x è fuori dall'intervallo [1, 4] e fuori dall'intervallo [15, 20].

Risposta : 1

25) Si consideri il seguente requisito:

RQ1: Durante l'esecuzione del programma (cioè per tutti gli istanti di tempo positivi) la variabile x è sempre nell'intervallo [0, 5] oppure [10, 15]

Quale dei seguenti monitor meglio descrive il requisito RQ1 ?

1.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x >= 0) or (x <= 5)) and ((x >= 10) or (x <= 15)));algorithmwhen edge(z) theny := true;end when;end Monitor;

2.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and ((x >= 5) or (x <= 0)) and ((x >= 15) or (x <= 10)) ;algorithmwhen edge(z) theny := true;end when;end Monitor;

3.class MonitorInputReal x; // plant outputOutputBoolean y;Boolean z;initial equationy = false;equationz = (time > 0) and (((x >= 0) and (x <= 5)) or ((x >= 10) and (x <= 15)));algorithmwhen edge(z) theny := true;end when;end Monitor;

Risposta : 2

26) Unit testing si concentra su:

1.Testare funzionalità di unità software individuali, oggetti, classi o metodi.

2.Testare l'interazione tra componenti.

3.Testare le interfacce di ciascuna componente.

Risposta : 1

27) "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 di performance.

2.Requisito funzionale.

3.Requisito non-funzionale.

Risposta : 2

28) 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) rggiunti almeno una volta.

Si consideri lo state diagram in figura

ed il seguente insieme di test cases:

1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2;

2) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2;

3) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2.

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

Immagine

1.90%

2.60%

3.80%

Risposta : 1

29) 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 il seguente programma C:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#define N  1   /* number of test cases */
int f(int x)  {   int y = 0;
  LOOP:  if (abs(x) - y <= 2)
                    {return   ;}
               else {y = y + 1; goto LOOP;}
}  /* f()  */
int main() {  int  i, y;   int x[N];
// define test cases
    x[0] = 3;
```

```
// testing
  for (i = 0; i < N; i++)  {
      y = f(x[i]);       // function under testing
      assert(y == (abs(x[i]) <= 2) ? 0 : (abs(x[i]) - 2));   // oracle
    }
   printf("All %d test cases passed\n", N);
  return (0);
}
```

Il programma main() sopra realizza il nostro testing per la funzione f(). I test cases sono i valori in x1[i] ed x2[i].

Quale delle seguenti è la branch coverage conseguita?

Immagine

1.100%

2.80%

3.50%

Risposta : 1

30) 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 era stata richiesta una risorsa (variabile x positiva) allora ora è concesso l'accesso alla risorsa (variabile y positiva)

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) and (delay(x, 10) > 0) 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 > 60) or (delay(x, 10) > 0) or (y <= 0);algorithmwhen edge(wz) thenwy := true;end when;end Monitor;

Risposta : 1

31) 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 il seguente programma C:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#define N  4    /* number of test cases */


int f(int x1, int x2)
{
  if (x1 + x2 <= 2)
  return (1);
  else return (2);
}


int main() {    int  i, y;   int x1[N], x2[N];
  // define test cases
    x1[0] = 3; x2[0] = -2;    x1[1] = 4; x2[1] = -3;    x1[2] = 5; x2[2] = -4;    x1[3] = 6; x2[3] = -5;
    // testing
  for (i = 0; i < N; i++)  {
      y = f(x1[i], x2[i]);       // function under testing
      assert(y ==(x1[i], x2[i] <= 2) ? 1 : 2);   // oracle
    }
   printf("All %d test cases passed\n", N);
    return (0);
}
```

Il programma main() sopra realizza il nostro testing per la funzione f1(). I test cases sono i valori in x1[i] ed x2[i].

Quale delle seguenti è la branch coverage conseguita?

1.50%

2.80%

3.100%

Risposta : 1

32) Il system testing si concentra su:

1.Testare le interfacce per ciascuna componente.

2.Testare le funzionalità di unità software individuali, oggetti, classi o metodi.

3.Testare l'interazione tra le componenti del sistema (cioè, integrazione di molte unità di sistema).

Risposta : 3

33) 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

34) 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*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'e' il valore massimo di P che garantisce la validità del modello di business di cui sopra ?

1.P = 1/1000

2.P = 1/10

3.P=1/10000

Risposta : 2

35) 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

36) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?

Immagine

1.Se ci sono abbastanza monete è sempre possibile ottenere la bevanda selezionata.

2.Una volta inserite le monete bisogna necessariamente consumare almeno una bevanda.

3.Anche se ci sono abbastanza monete potrebbe non essere possibile ottenere la bevanda selezionata.

Risposta : 1

37) 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

38) Si consideri il Test-Driven Development (TDD). Quale delle seguenti affermazioni è vera?

1.Per ciascun incremento di funzionalità, scrivi test automatizzati, implementa la funzionalità, esegui i test e rivedi l'implementazione come necessario.

2.Per ciascun incremento di funzionalità, implementa la funzionalità, scrivi test automatizzati, esegui i test e rivedi l'implementazione come necessario.

3.Scrivi test automatizzati per tutti i requisiti di sistema, esegui i test e rivedi l'implementazione come necessario.

Risposta : 1

39) Quale delle seguenti affermazione è vera riguardo al beta testing ?

1.Test automatizzato sono eseguiti sulla versione finale del sistema presso il cliente.

2.Una release del software è resa disponibile agli utenti (beta users) per permettergli di sperimentare e quindi segnalare eventuali problemi rilevati agli sviluppatori.

3.Test automatizzato sono eseguiti sulla versione finale del sistema presso il sito di sviluppo del software.

Risposta : 2

40) 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 nonegativa.

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 : 2

41) Il component testing si concentra su:

1.Testare funzionalità di unità software individuali, oggetti, classi o metodi.

2.Testare le interfacce per ciascun componente.

3.Testare l'interazione tra molte componenti (cioè integrazione di molte unità).

Risposta : 2

42) Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?

Immagine

1.La macchina non dà resto.

2.Una volta selezionata la bevanda non è possibile cancellare l'operazione.

3.Una volta inserite monete per due bevande è possibile ottenerle senza reinserire le monete.

Risposta : 2

43) La transition coverage di un insieme di test cases (cioè sequeze 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:

1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2

2) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 2, End PIN Validation 2

3) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, More than 3 failed..., END PIN validation 1;

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

Immagine

1.100%

2.80%

3.90%

Risposta : 3

44) Component testing focuses on:

1.Testing functionalities of individual program units, object classes or methods.

2.Testing interfaces for each component (i.e., integration of several units).

3.Testing interactions among components (i.e., integration of several units).

Risposta : 2

45) Quale delle seguenti affermazioni è vera riguardo all'alpha testing ?

1.Gli utenti del sistema lavorano insieme al team di sviluppo per testare il software nel sito di sviluppo.

2.Test automatizzati sono eseguiti su una versione preliminare del sistema.

3.Test automatizzati sono eseguiti sulla prima release del sistema.

Risposta : 1

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

1.Accertarsi che il sistema soddisfi i requisiti dati.

2.Accertarsi che l'architettura del sistema soddisfi i requisiti dati.

3.Accertarsi che i requisiti definiscano un sistema che risolve il problema che l'utente pianifica di risolvere.

Risposta : 3

47) 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.Le attività di definizione dei requisiti e di sviluppo sono interleaved.

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

Risposta : 2

48) Lo state diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti modelli Modelica è plausibile per lo state diagram in figura?

Immagine

1.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer := 0; elsewhen sample(0, T) thenif (pre(state) == 0) and (Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state := 3; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 0; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end CoffeeMachine;

2.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer := 0; elsewhen sample(0, T) thenif (pre(state) == 0) and (Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state := 0; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 3; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end CoffeeMachine;

3.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer := 0; elsewhen sample(0, T) thenif (pre(state) == 0) and (Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state := 3; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 3; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end CoffeeMachine;

Risposta : 3

49) 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 il seguente programma C:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#define N  4    /* number of test cases */


int f(int x1, int x2)
{
  if (x1 + x2 <= 2)
  return (1);
  else return (2);
}


int main() {    int  i, y;   int x1[N], x2[N];
  // define test cases
    x1[0] = 3; x2[0] = -2;    x1[1] = 4; x2[1] = -3;    x1[2] = 7; x2[2] = -4;    x1[3] = 8; x2[3] = -5;
    // testing
  for (i = 0; i < N; i++)  {
      y = f(x1[i], x2[i]);       // function under testing
      assert(y ==(x1[i], x2[i] <= 2) ? 1 : 2);   // oracle
    }
   printf("All %d test cases passed\n", N);
    return (0);
```

}

Il programma main() sopra realizza il nostro testing per la funzione f1(). I test cases sono i valori in x1[i] ed x2[i].

Quale delle seguenti è la branch coverage conseguita?

1.50%

2.100%

3.80%

Risposta : 2

50) Un processo di sviluppo agile consiste di 3 iterazioni identiche di costo A. Alla fine di ogni iterazione vengono prese in considerazione le "change requests" e, se ve ne sono, l'iterazione viene ripetuta. Sia p la probabilità che ci siano "change requests" all fine di una iterazione. Il valore atteso del costo del progetto è:

1.3*(1 + p)*A

2.3*p*A

3.3*(A + p)

Risposta : 1