

1.block SysArch;SC1 sc1SC2 sc2;SC3 sc3:SC4 sc4:connect(sc1.input4. sc2.output4);connect(sc1.output4, sc2.input4):connect(sc1.input5. sc3.output5):connect(sc1.output5. sc3.input5);connect(sc1.input6, sc4.output6);connect(sc1.output6, sc4.input6);connect(sc2.input1);connect(sc3.input2, sc4.output2);connect(sc4.output3, sc2.input3);end SysArch 2.block SvsArch:SC1 sc1SC2 sc2:SC3 sc3;SC4 sc4;connect(sc1.input4, sc2.output4);connect(sc1.output4, sc3.output5);connect(sc1.output5, sc2.input4);connect(sc1.input5, sc3.input5);connect(sc1.input6, sc4.output6);connect(sc1.output6, sc4.input6);connect(sc2.input1, sc3.output1);connect(sc3.input2, sc4.output2);connect(sc4.input3, sc2.ouput3);end SysArch SysArch;SC1 sc1SC2 sc2;SC3 sc3;SC4 sc4;connect(sc1.input4, sc2.output4);connect(sc1.output4, sc2.input4);connect(sc1.input5, sc3.output5);connect(sc1.output5, sc3.input5);connect(sc1.input6, sc4.output6);connect(sc1.output6, sc4.input6);connect(sc2.output1, sc3.input1);connect(sc3.output2, sc4.input2);connect(sc4.output3, sc2.input3);end SysArch

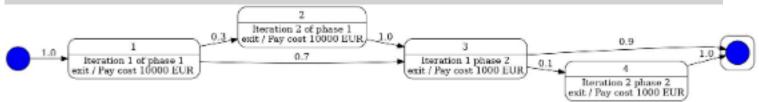
Risposta: 3

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.

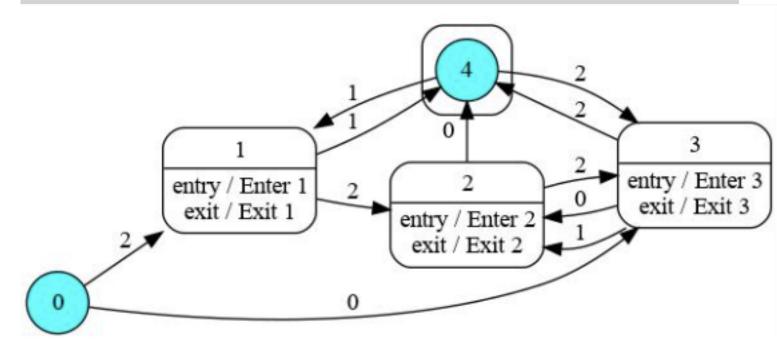
Ad esempio lo state diagram in figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità dello 0.3 di dover essere ripetuta (a causa di errori) ed F2 ha una probabilità 0.1 di dover essere ripetuta (a causa di errori).

Uno scenario è una sequenza di stati.

Qual'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 ?



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



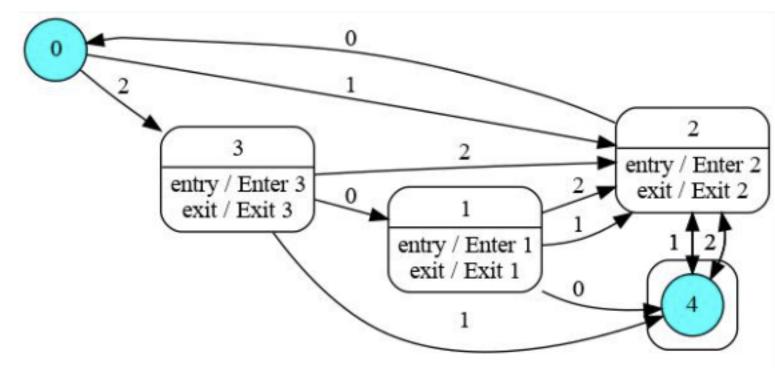
1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer; connector OutputInteger = output Integer; // InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1; algorithmwhen initial() then x := 0; elsewhen sample(0,T)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

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 := 1; elseif (pre(x) == 0) and (pre(u) == 2) then x := 4; elseif (pre(x) == 1) and (pre(u) == 2) then x := 3; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 0; elseif (pre(x) == 3) and (pre(u) == 1) then x := 4; elseif (pre(x) == 4) and (pre(u) == 0) then x := 3; 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; // nputInteger u; // external

inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() then x := 0;elsewhen sample(0,T) then if (pre(x) == 0) and (pre(u) == 0) then x := 3;elseif (pre(x) == 0) and (pre(u) == 0) then x := 3;elseif (pre(x) == 0) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 3) and (pre(y) == 0) then y := 4;elseif (pre(y) == 4) and (pre(y) == 0) then y := 4;elseif (pre(y) == 4) and (pre(y) == 0) then y := 4;elseif (pre(y) == 4) and (pre(y) == 0) then y := 4;elseif (pre(y) == 4) and (pre(y) == 0) then y := 4;elseif (pre(y) == 4) and (pre(y) == 4) and (pre(y) == 2) then y := 4;elseif (pre(y) == 4) and (pre(y) == 3) then y := 4;elseif (pre(y) == 4) and (pre(y) == 4) and (pre(y) == 3) then y := 4;elseif (pre(y) == 4) and (pre(y) == 4) and (pre(y) == 3) then y := 4;elseif (pre(y) == 4) and (pre(y) == 5) then y := 4;elseif (pre(y) == 4) and (pre(y) == 6) then y := 4;elseif (pre(y) == 6) then y

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

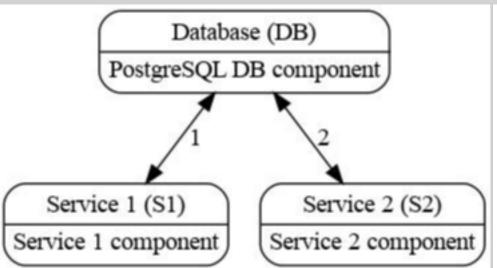


1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer; connector OutputInteger = output Integer; InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1; algorithmwhen initial() thenx := 0; elsewhen sample(0,T)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

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

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() then x := 0; elsewhen sample(0,T) then if (pre(x) == 0) and (pre(u) == 1) then x := 2; elseif (pre(x) == 0) and (pre(u) == 2) then x := 2; elseif (pre(x) == 0) and (pre(u) == 2) then x := 2; elseif (pre(x) == 2) and (pre(u) == 0) then x := 4; elseif (pre(x) == 1) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 3) and (pre(u) == 3) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 5) th

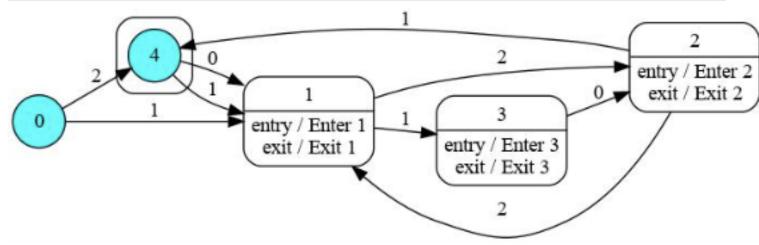
Si consideri la seguente architettura software: Quale dei seguenti modelli Modelica meglio la rappresenta ?



1.block SysArchDB db_c;S1 s1_c;S2 s2_c;connect(db_c.input[1], s1_c.output[1]);connect(db_c.output[1]);connect(s1_c.input[2]);connect(s1_c.output[2]);connect(s1_c.output[2]);end SysArch

2.block SysArchDB db_c;S1 s1_c;S2 s2_c;connect(db_c.input[1], s2_c.input[1]);connect(db_c.output[1], s2_c.input[1]);connect(s1_c.input[2], s2_c.output[2]);connect(s1_c.input[2], s2_c.output[2]);end SysArch

 $3. block \ SysArchDB \ db_c; S1 \ s1_c; S2 \ s2_c; connect (db_c.input[1], \ s1_c.input); connect (db_c.input[2], \ s2_c.input); connect (db_c.input[2], \ s2_c.input]; connect (db_c.input[2], \ s2_c.input[2], \ s3_c.input[2], \ s3_c.input[2], \ s3_c.input[2], \ s3_c.input[2]$



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) == 0) then x := 3;elseif (pre(x) == 1) and (pre(u) == 0) then x := 3;elseif (pre(x) == 1) and (pre(u) == 0) then x := 1;elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;elseif (pre(x) == 3) and (pre(u) == 1) then x := 2;elseif (pre(x) == 3) and (pre(u) == 1) then x := 4;elseif (pre(x) == 3) and (pre(u) == 1) then x := 2;elseif (pre(x) == 3) and (pre(u) == 1) then x := 3;elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;elseif (pre(x) == 4) and (pre(u) == 4) and (pre(u) == 4) and (p

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 := 1;elseif (pre(x) == 0) and (pre(u) == 2) then x := 4;elseif (pre(x) == 2) and (pre(u) == 2) 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) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) and (pre(u) == 0) then x := 1;elseif (pre(x) == 0) then x := 1;elseif (pre(

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.

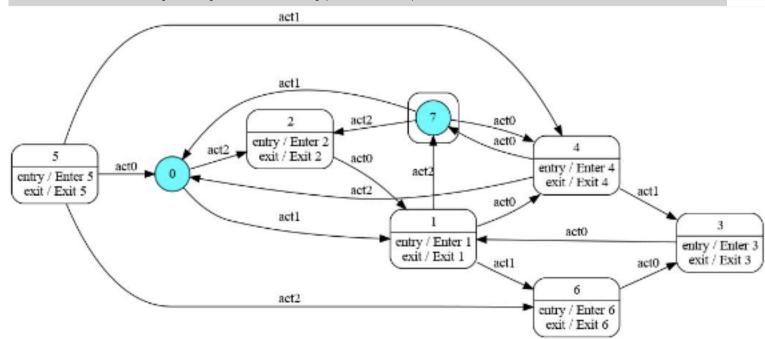
Si consideri lo state diagram in figura

Si consideri il sequente insieme di test cases:

Test case 1: act1 act2

Test case 2: act2 act0 act1 act0 act

Test case 3: act2 act0 act2 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



- 1.State coverage: 75%
- 2.State coverage: 80%
- 3.State coverage: 70%

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.

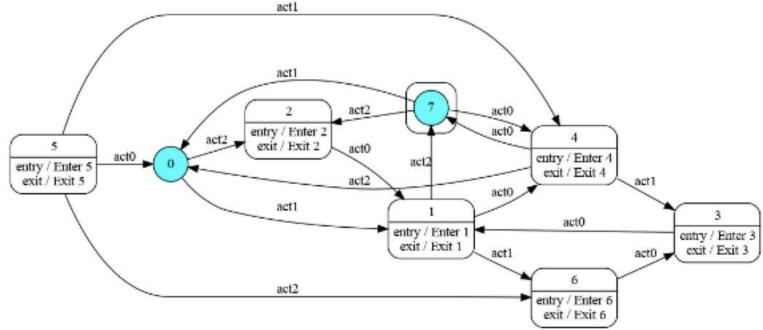
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act1 act2

Test case 2: act2 act0 act1 act0 act0

Test case 3: act2 act0 act2

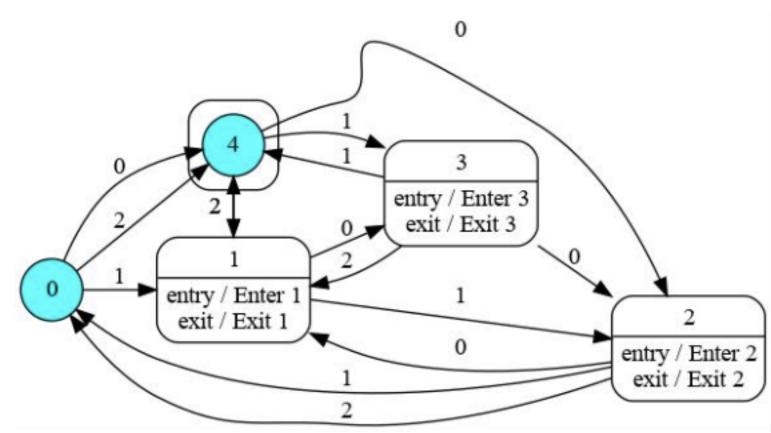
Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?



- 1.Transition coverage: 60%
- 2.Transition coverage: 50%
- 3.Transition coverage: 35%

Risposta: 3

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



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 := 4; elseif (pre(x) == 0) then x := 0; elseif (pre(x) == 1) 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) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 3) and (pre(u)

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() then x := 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) == 1) and (pre(u) == 2) then x := 4;elseif (pre(x) == 1) 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) == 2) then x := 1;elseif (pre(x) == 2) and (pre(u) == 2) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 3) and (pre(u) == 3) then x := 4;elseif (pre(x) == 3) and (pre(u) == 3) then x := 4;elseif (pre(x) == 3) and (pre(u) == 3) then x := 4;elseif (pre(x) == 3) and (pre(u) == 3) then x := 4;elseif (pre(x) == 3) then x := 4;elseif (pre(x) == 4) then x := 4;elseif (pre(x) == 3) then x := 4;elseif (pre(x) == 4) then x := 4;elseif (pre(x) == 3) then x := 4;elseif (pre(x) == 4) then x := 4;elseif (pre(x) == 3) then x := 4;elseif (pre(x) == 3) then x := 4;elseif (pre(x) == 4) then

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

Risposta : 2

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

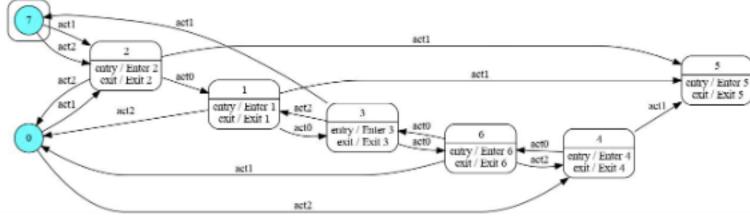
ed il seguente insieme di test cases:

Test case 1: act2 act0 act1 act2 act0

Test case 2: act1 act2 act1

Test case 3: act1 act2 act1 act0 act0

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



- 1.Transition coverage: 60%
- 2.Transition coverage: 40%
- 3.Transition coverage: 80%

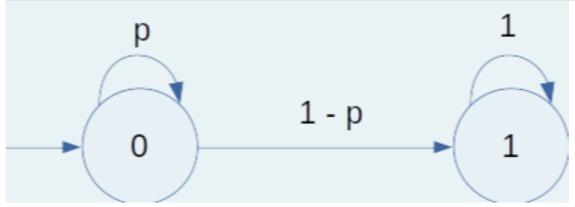
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il tempo necessario per completare la fase x è time(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta 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 $X = x(0), x(1), x(2), \dots$ è Time(X) = time(x(0)) + time(x(1)) + time(x(2)) + \dots

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



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

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

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

Risposta: 1

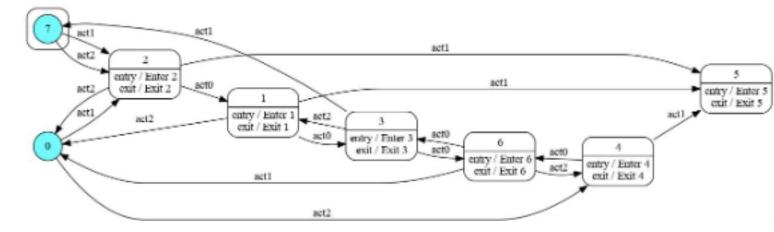
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

Test case 1: act2 act0 act1 act2 act0

Test case 2: act1 act2 act1

Test case 3: act1 act2 act1 act0 act0 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



- 1.State coverage: 100%
- 2.State coverage: 50%
- 3.State coverage: 75%

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.

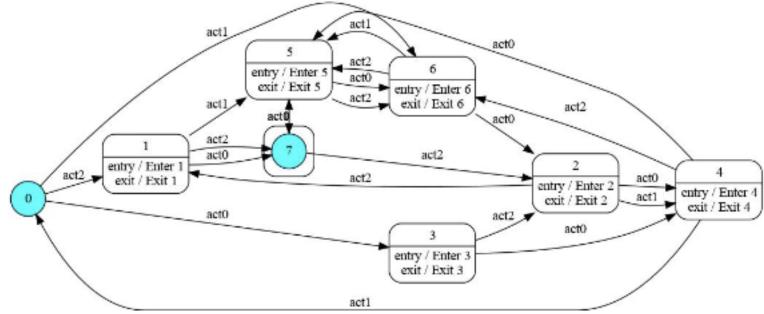
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

Test case 1: act1 act0 act0 act0 act0

Test case 2: act2 act0

Test case 3: act0 act0 act1 act0 act2

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



- 1.State coverage: 100%
- 2.State coverage: 90%
- 3.State coverage: 80%

Risposta : 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.

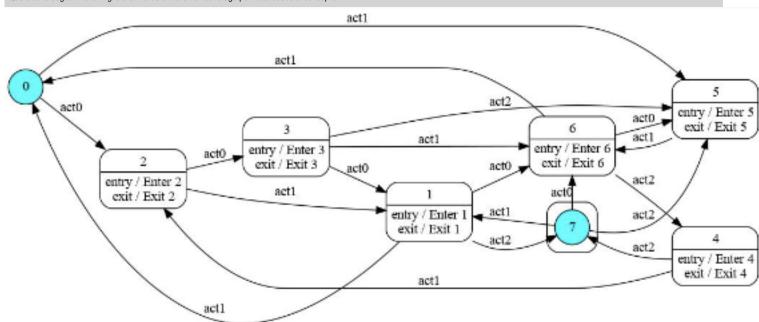
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act1 act1 act2 act2

Test case 2: act1 act1 act0 act1

Test case 3: act0 act0 act2 act1 act0

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



- 1.Transition coverage: 60%
- 2.Transition coverage: 80%
- 3.Transition coverage: 40%

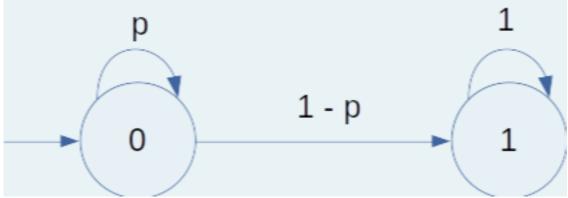
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il costo dello stato (fase) x è c(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta il completamento 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 $X = x(0), x(1), x(2), \\ e C(X) = c(x(0)) + c(x(1)) + c(x(2)) + ...$

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



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

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

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

Risposta: 3

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

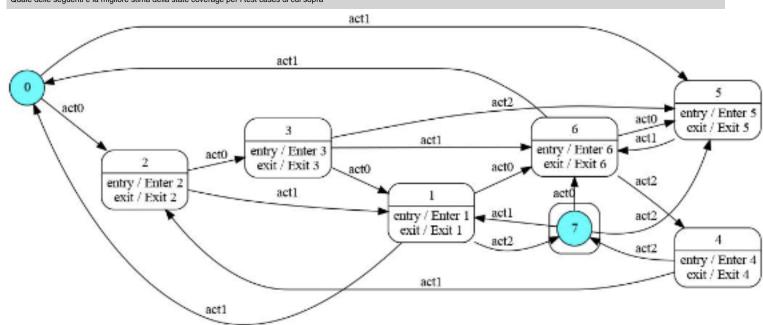
Si consideri il seguente insieme di test cases:

Test case 1: act1 act1 act2 act2

Test case 2: act1 act1 act0 act1

Test case 3: act0 act0 act2 act1 act0

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



1.State coverage: 70%

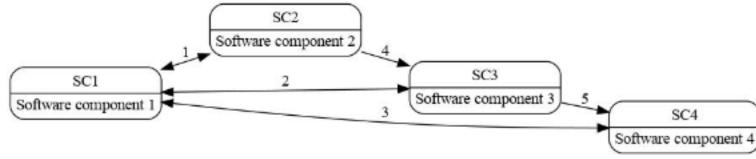
2.State coverage: 80%

3.State coverage: 90%

Risposta: 3

Si consideri la seguente architettura software:

Quale dei seguenti modelli Modelica meglio la rappresenta?



sc1SC2 sc2;SC3 sc3;SC4 sc3.output2);connect(sc1.output2, 1.block SysArch;SC1 sc4;connect(sc1.input1, sc2.output1);connect(sc1.output1, sc2.input1);connect(sc1.input2, sc3.input2);connect(sc1.input3, sc4.output3);connect(sc1.output3, sc4.input3);connect(sc2.input4, sc3.output4);connect(sc3.input5, sc4.output5);end SysArch

SysArch;SC1 sc1SC2 sc2:SC3 sc3;SC4 sc4;connect(sc1.input1, sc2.output1);connect(sc1.output1, sc3.output2);connect(sc1.output2, 2.block sc2.input1);connect(sc1.input2, sc3.input2);connect(sc1.input3, sc4.output3);connect(sc1.output3);connect(sc2.input4, sc3.output4);connect(sc2.output4, sc3.input4);connect(sc3.input4);connect(sc3.input4);connect(sc3.input5, sc4.output5);connect(sc3.ou sc4.input5);end SysArch

3.block SysArch;SC1 sc1SC2 sc2;SC3 sc3;SC4 sc4;connect(sc1.input1, sc2.output1);connect(sc1.output1, sc2.input1);connect(sc1.input2, sc3.output2);connect(sc1.output2, sc3.input2);connect(sc1.input3, sc4.output3);connect(sc1.output3, sc4.input3);connect(sc2.output4, sc3.input4);connect(sc3.output5, sc4.input5);end SysArch

Risposta: 3

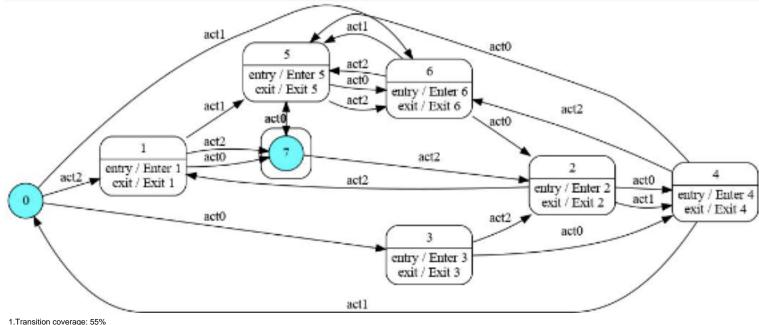
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta. Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act1 act0 act0 act0 act0

Test case 2: act2 act0

Test case 3: act0 act0 act1 act0 act2

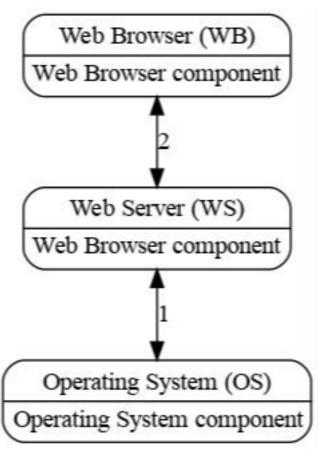
Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?



- 2.Transition coverage: 35%
- 3.Transition coverage: 45%

Risposta: 3

Si consideri la seguente architettura software:Quale dei seguneti modelli Modelica meglio la rappresenta.



1.block SysArchOS os_c;WS ws_c;WB wb_c;connect(os_c.input1, ws_c.output1);connect(os_c.input1, ws_c.input1);connect(wb_c.input2, ws_c.output2);connect(wb_c.input2);connect(wb_c. SysArch

2.block SysArchOS os_c;WS ws_c;WB wb_c;connect(os_c.cinput1, ws_c.cinput1);connect(wb_c.cinput1);connect(wb_c.cinput2);connect(wb_c. SysArch

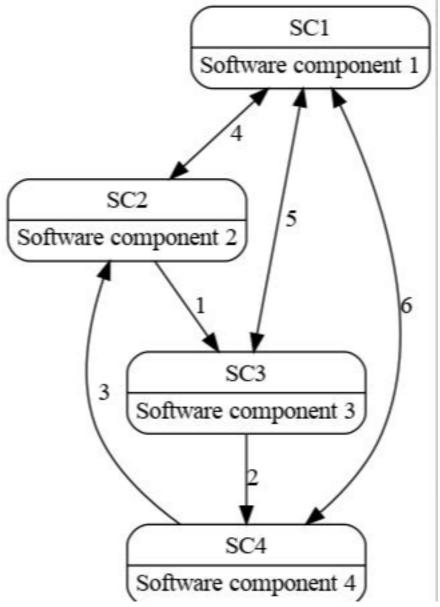
3.block SysArchOS os_c;WS ws_c;WB wb_c;connect(os_c.input1, wb_c.output1);connect(os_c.output1);connect(wb_c.input2);end SysArch

Risposta: 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 1 /* number of test cases */
int f(int x) { int y = 0;
 LOOP: if (abs(x) - y \le 2)
          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]) \le 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?



1.80%

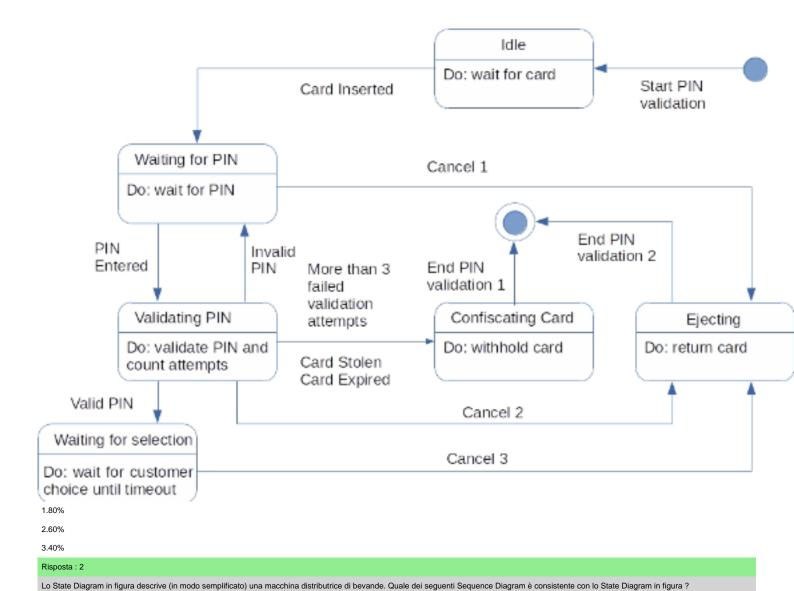
2.50%

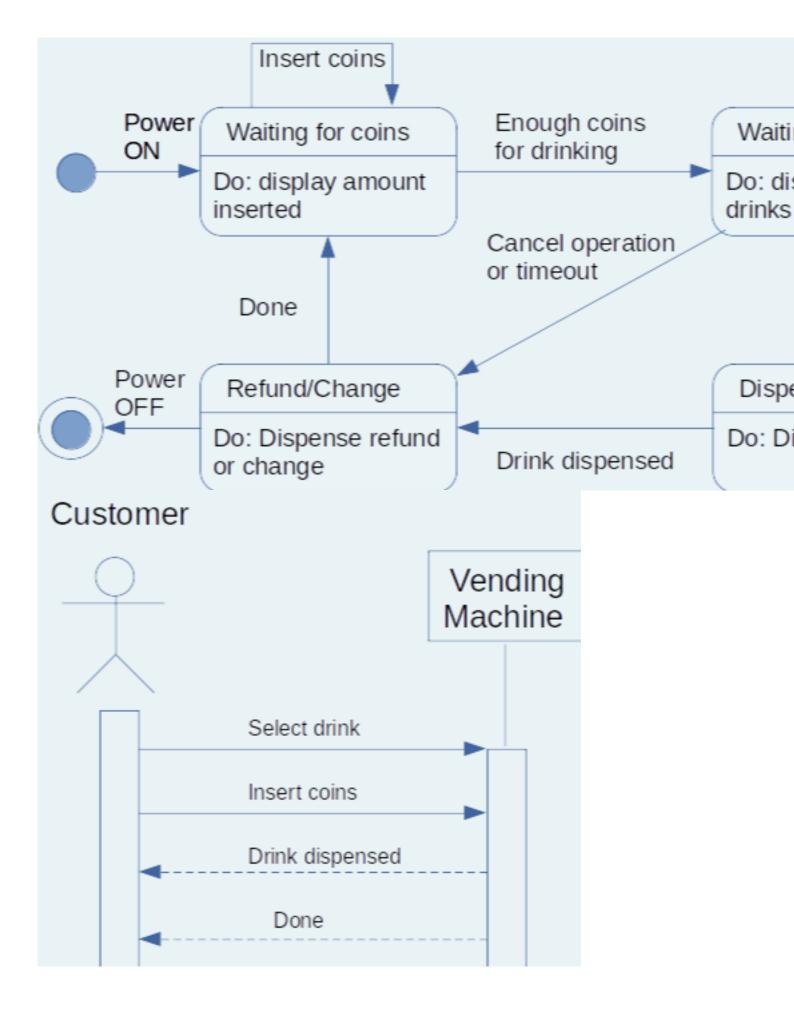
3.100%

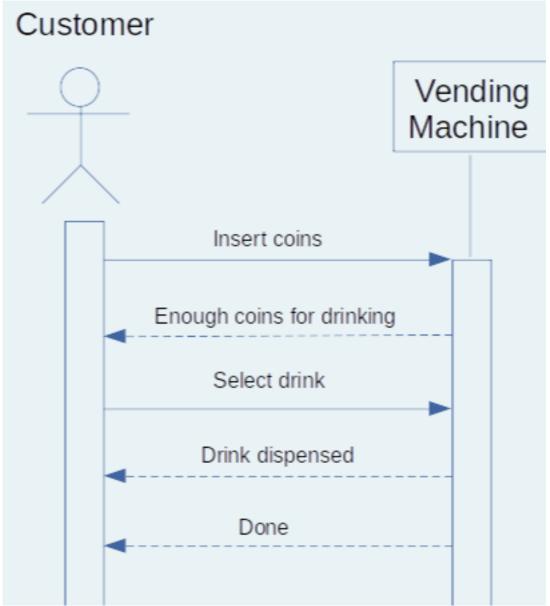
Risposta: 3

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, 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

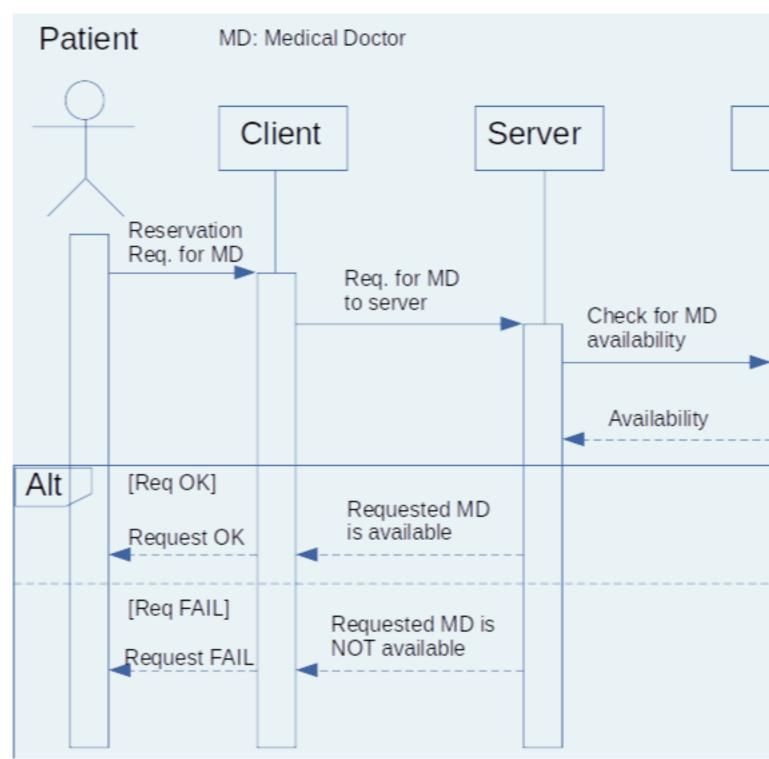






iVBORw0KGgoAAAANSUhEUgAAAZcAAAACCAYAAACUupFJAAAAKEIEQVRIiWN8+fnrf4ZRMApGwSgYBaOAioBpoB0wCkbBKBgFo2D4AQCkdAPUYeTFMQAAAABJRU5ErkJggg==

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

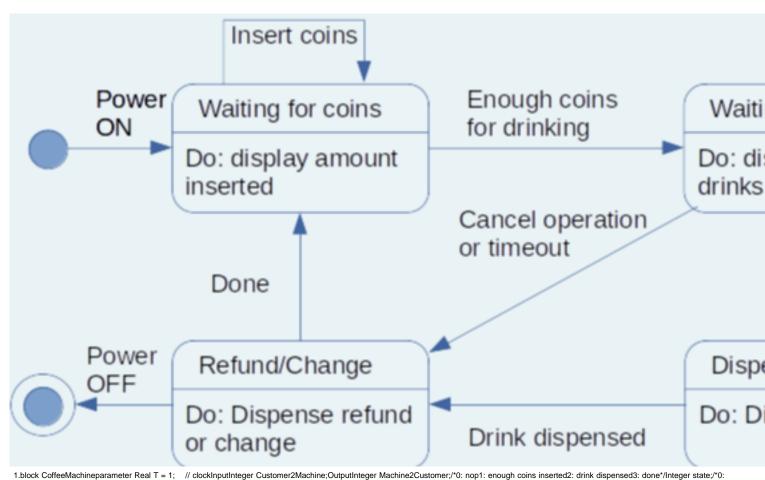


^{1.}II paziente richiede al server una visita con uno specifico medico e, dopo una verifica sul database, riceve conferma dal server della disponibilità o meno del medico richiesto.

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?

^{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.



waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0; Machine2Customer := 0; elsewhen sample(0, T) thenif (Customer2Machine == 1) then // customer has inserted enough coins Machine2Customer := 1: elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected state := 1: then // drink selected state := 2: // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund 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); 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 (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 3; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); CoffeeMachine:

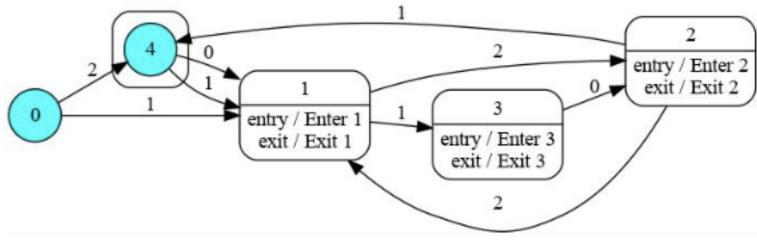
3.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer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

Risposta: 3

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 2Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



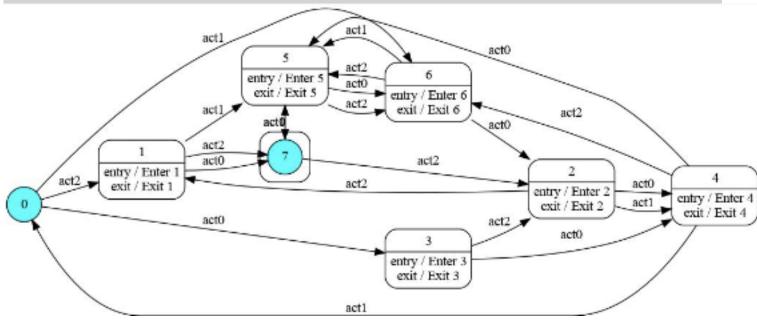
1.80%

2.90%

3.60%

Risposta: 2

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 ?



- 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

Quale pattern architetturale meglio descrive $\,$ l'architettura in figura $\,$?

Web Browser

Forms and Query Manager

Database

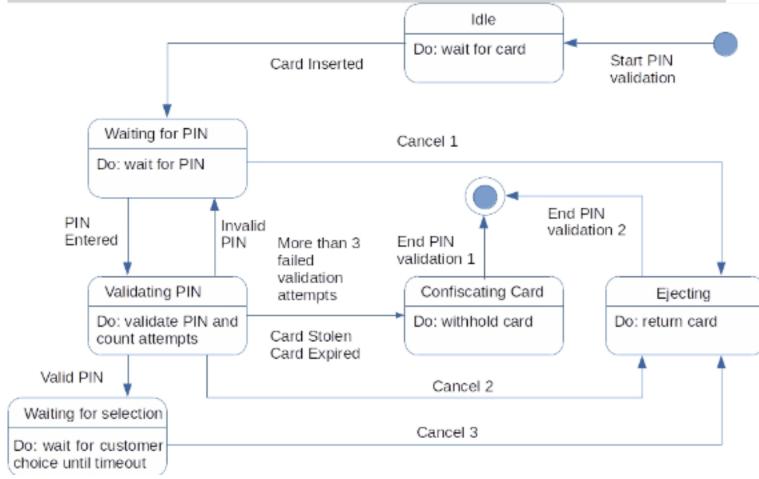
- 1.Model View Controller.
- 2.Layred architecture.
- 3.Pipe and filter architecture.

Risposta: 2

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



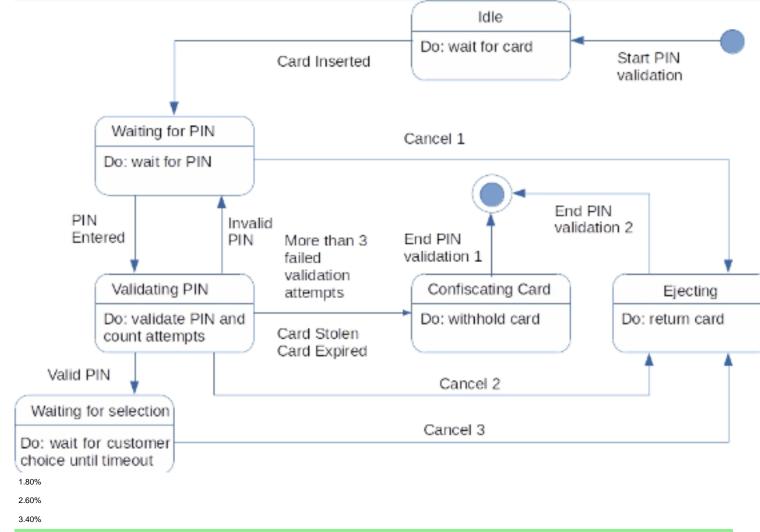
1.80%

2.90%

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.

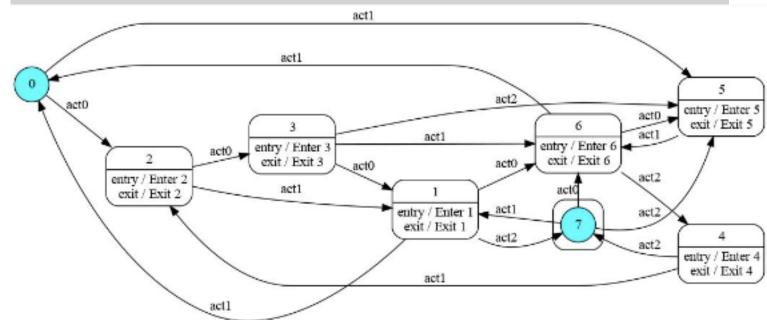
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

- 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 2Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra



Risposta: 2

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 ?

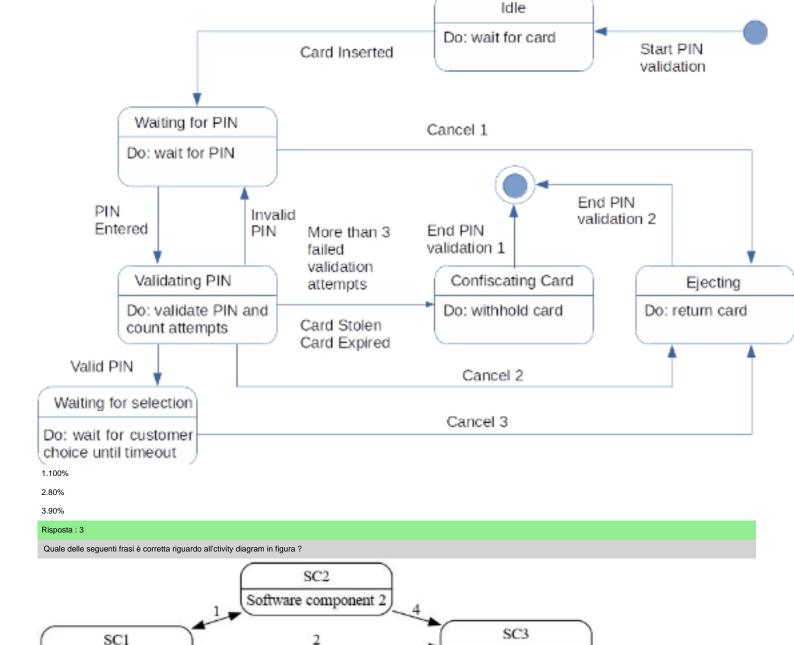


- 1.Una volta inserite monete per due bevande è possibile ottenerle senza reinserire le monete.
- 2.Una volta selezionata la bevanda non è possibile cancellare l'operazione.
- 3.La macchina non dà resto.

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



3

Software component 3

SC4 Software component 4

- 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 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 sequenza.

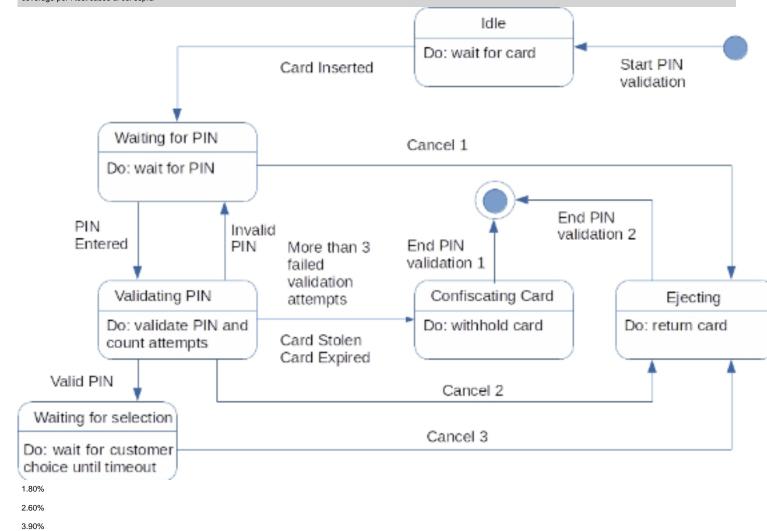
Risposta: 1

Software component 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.

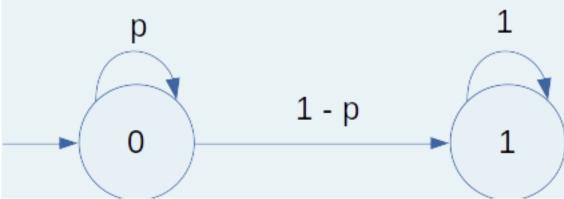
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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



Risposta : 3

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.



1.1/(p*(1 - p))

2.1/(1 - p)

3.(1 - p)/p

Risposta: 2

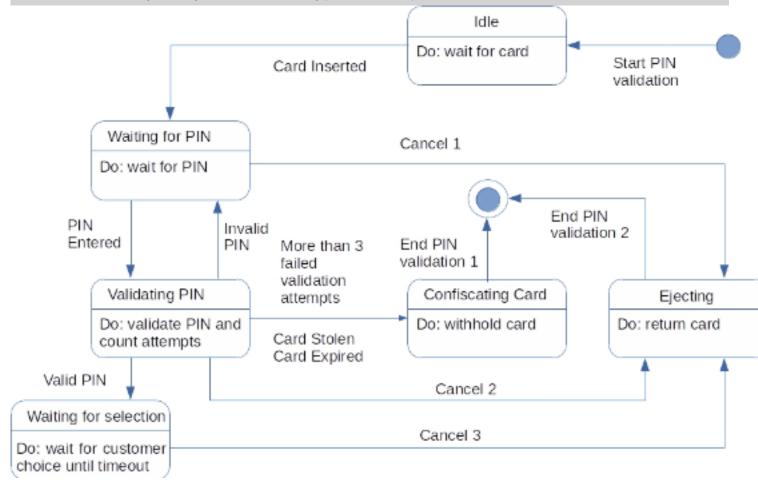
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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?



1.Transition coverage: 75%

2.Transition coverage: 25%

3.Transition coverage: 50%

Risposta: 2

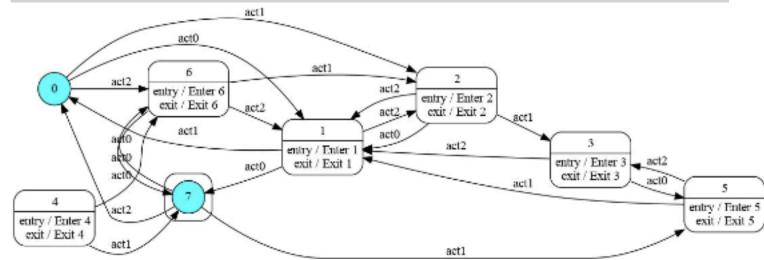
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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



1.State coverage: 100%

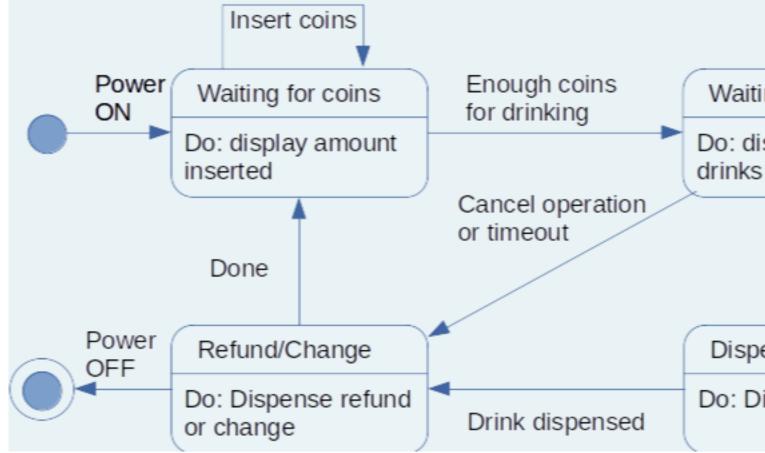
2.State coverage: 75%

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

Uno scenario è una sequenza di stati.

Qual'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 ?



1.0.07

2.0.03 3.0.27

Risposta: 2

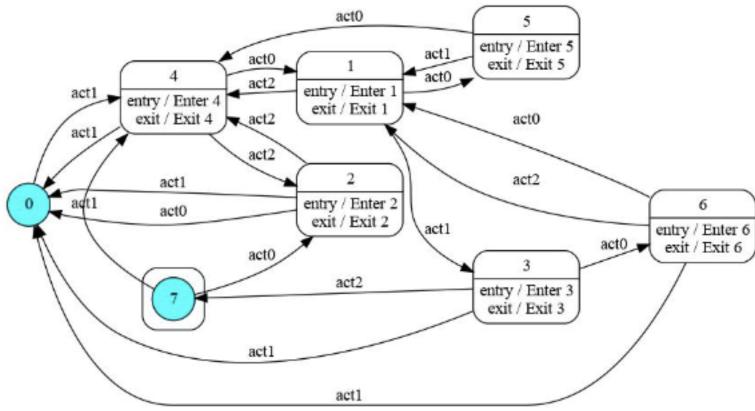
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura ed il seguente insieme di test cases

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

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?



- 1.Transition coverage: 50%
- 2.Transition coverage: 75%
- 3.Transition coverage: 100%

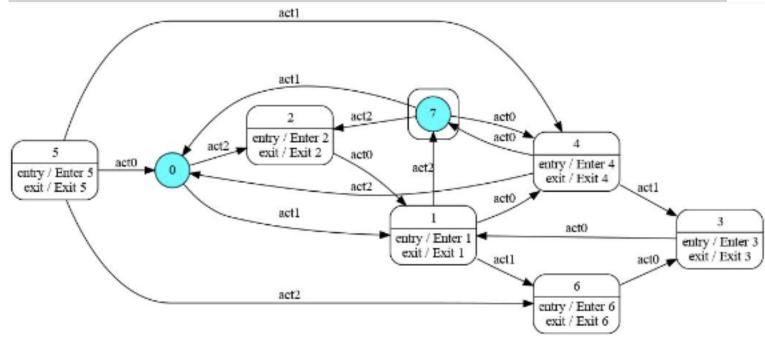
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

Test case 1: act2 act1 act1 act2 act1 act0

Test case 2: act2 act0 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



- 1.State coverage: 87%
- 2.State coverage: 100%
- 3.State coverage: 50%

Risposta: 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.

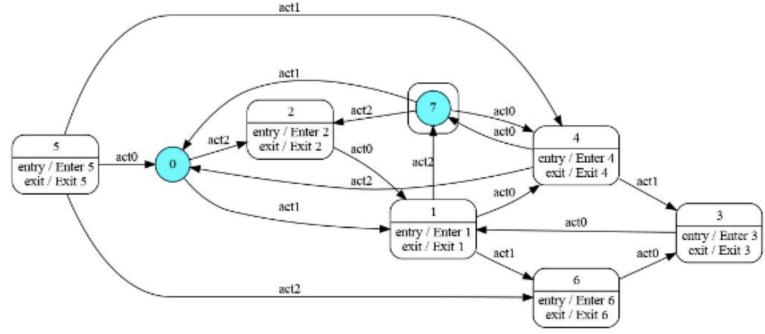
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: 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?



- 1.Transition coverage: 40%
- 2.Transition coverage: 70%
- 3.Transition coverage: 100%

Risposta: 1

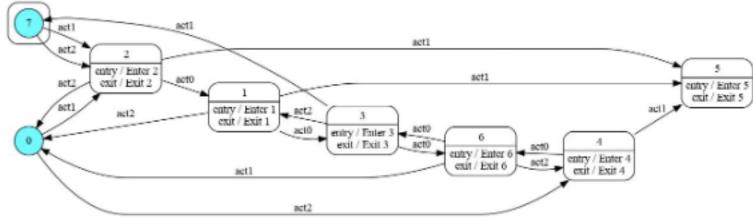
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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 act0 act2 act0 act2 act0 act0 act0 act2 act2 act2 act2 act2

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 act2 act0 act1 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



- 1.State coverage: 87%
- 2.State coverage: 100%
- 3.State coverage: 50%

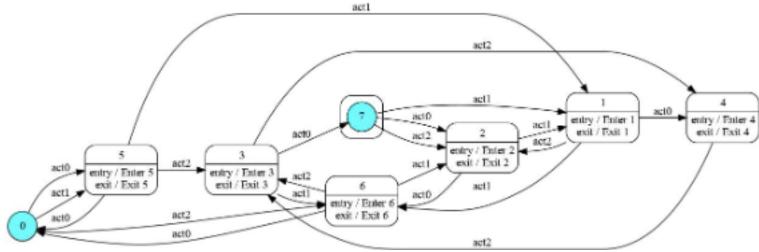
Risposta: 1

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: act1 act2 act0

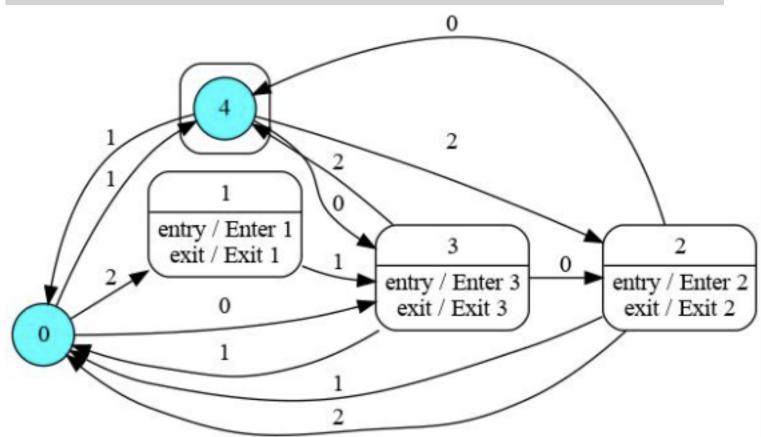
Test case 2: act2 act2 act2 act2 act2 act0



- 1.Transition coverage: 35%
- 2.Transition coverage: 50%
- 3.Transition coverage: 90%

Risposta: 2

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



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) == 2) then x := 0;elseif (pre(x) == 3) and (pre(u) == 2) then x := 4;elseif (pre(x) == 3) and (pre(u) == 2) then x := 4;elseif (pre(x) == 3) and (pre(u) == 2) then x := 4;elseif (pre(x) == 3) 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) == 1) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) =

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

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) (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) = 3) and (pre(u) = 3) a == 4) and (pre(u) == 2) then x := 3;else x := pre(x); // defaultend if;end when;end FSA;

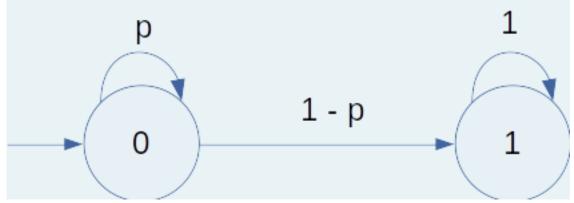
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il tempo necessario per completare la fase x è time(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta 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 $X = x(0), x(1), x(2), \dots$ è Time(X) = time(x(0)) + time(x(1)) + time(x(2)) + \dots

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



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

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

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

Risposta: 3

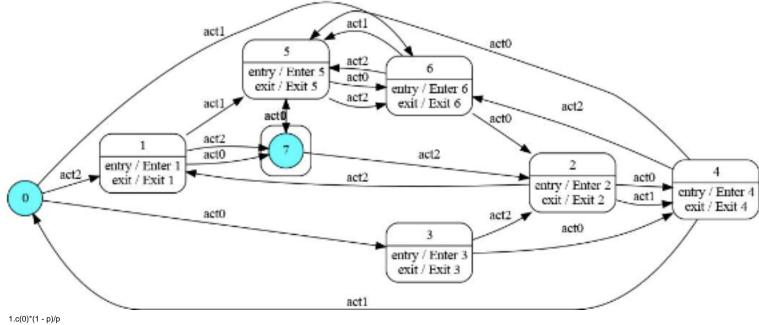
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il costo dello stato (fase) x è c(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta il completamento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



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

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

Risposta: 2

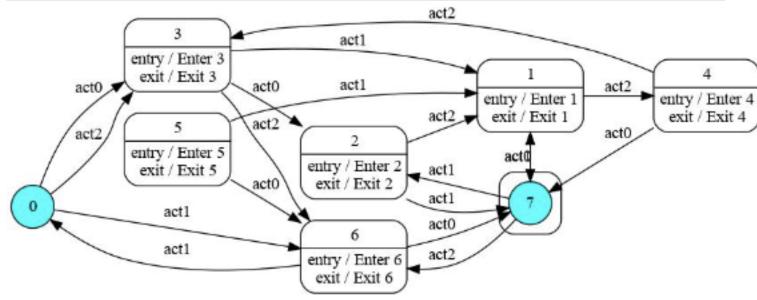
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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 act0 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



1.State coverage: 75%

2.State coverage: 100%

3.State coverage: 60%

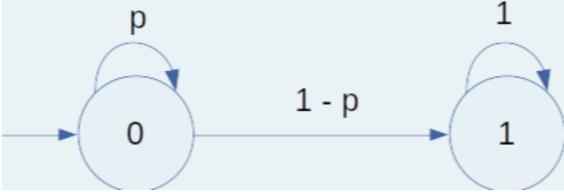
Risposta: 1

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 figuraRappresenta 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) ?



1.0.28

2.0.12

3.0.42

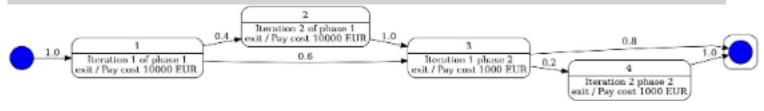
Risposta: 1

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.

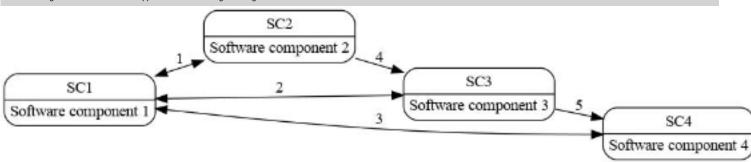
Ad esempio lo state diagram in figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità 0.4 di dover essere ripetuta (a causa di errori) ed F2 ha una probabilità 0.2 di dover essere ripetuta (a causa di errori).

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) ?



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

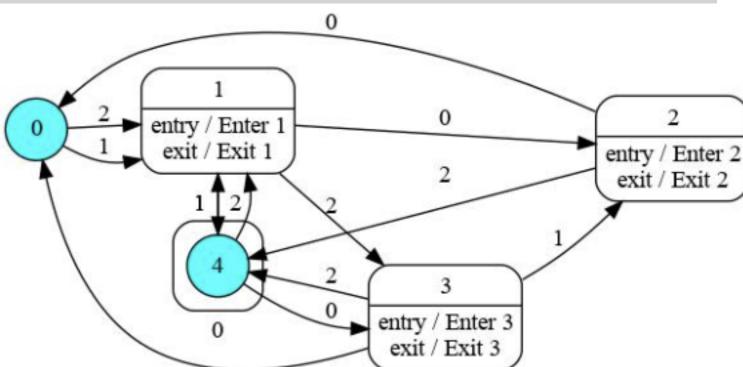


1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer,*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() then x := 0;elsewhen sample(0,T) thenif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

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() 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) == 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) == 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 := 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() 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 := 3; elseif (pre(x) == 1) 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) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 3) and (pre(u) == 2) then x := 3; 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) == 3) and (pre(u) == 2) then x := 3; elseif (pre(x) == 3) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 3; elseif (pre(x) == 3) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 3) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (

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



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) == 1) then x := 3; elseif (pre(x) == 1) and (pre(u) == 1) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) and (pre(u) == 2) and (pre(u) == 1) then x := 1; elseif (pre(x) == 2) and (pre(u) == 2) and (pre(u) == 2) and (pre(u) == 2) and (pre(u) == 3) and (pre(u) == 2) and (pre(u) == 3) and (pre(u) == 3)

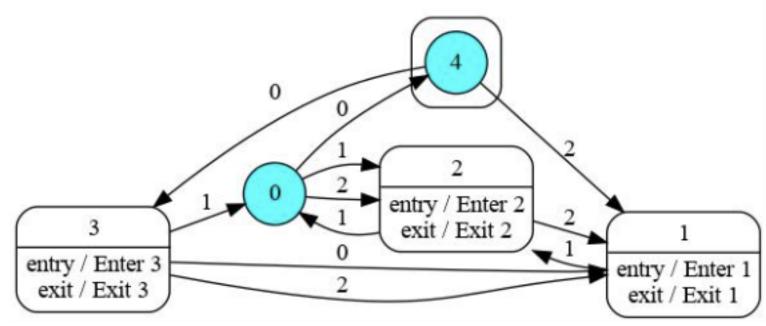
(pre(x) = 2) and (pre(u) = 2) then x := 0; elseif (pre(x) = 3) and (pre(u) = 1) then x := 2; elseif (pre(x) = 4) and (pre(u) = 0) then x := 1; elseif (pre(x) = 4) and (pre(u) = 1) then x := 1; else x := 1; elseif (pre(x) = 4) and (pre(u) = 1) then x := 1; else x := 1; elseif (pre(x) = 4) and (pre(u)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

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 := 3;elseif (pre(x) == 1) and (pre(u) == 2) then x := 0;elseif (pre(x) == 1) and (pre(u) == 2) then x := 2;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 3) and (pre(u) == 0) then x := 3;elseif (pre(x) == 3) and (pre(u) == 0) then x := 4;elseif (pre(x) == 3) and (pre(u) == 2) then x := 3;elseif (pre(x) == 4) and (pre(u) == 4) then x := 3;elseif (pre(x) == 4) 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;

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 := 1; elseif (pre(x) == 0) and (pre(u) == 2) then x := 3; elseif (pre(x) == 2) and (pre(u) == 0) then x := 2; elseif (pre(x) == 1) and (pre(u) == 0) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; 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; elseif (pre(x) == 4) and (pre(u) == 2) then x := 1; elseif (pre(x) == 4) and (pre(u) == 4) a

Risposta : 3

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



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() then x := 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 := 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) == 3) and (pre(u) == 1) then x := 0;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 2) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 1) then x := 3;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 3) and (pre

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) == 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 := 1;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 2) and (pre(u) == 0) then x := 3;elseif (pre(x) == 3) and (pre(u) == 0) then x := 3;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) == 3) 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 := 2; elseif (pre(x) == 1) and (pre(u) == 1) then x := 2; elseif (pre(x) == 2) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 3) and

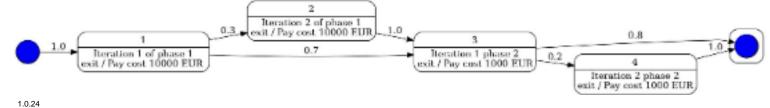
Risposta : 3

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 figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità 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?



.....

2.0.56

3.0.14

Risposta: 2

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.

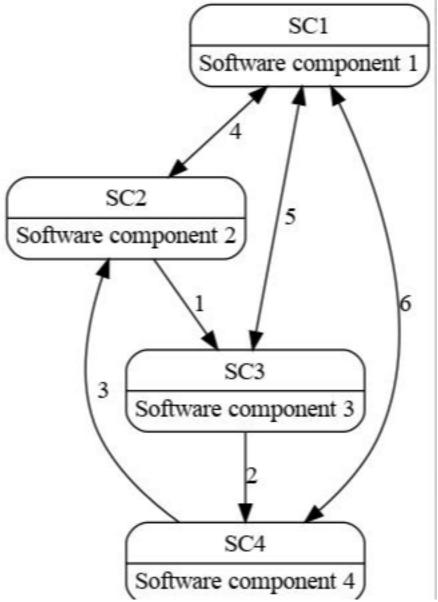
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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?



- 1.Transition coverage: 40%
- 2.Transition coverage: 70%
- 3.Transition coverage: 100%

Risposta: 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) raggiunti almeno una volta.

Si consideri lo state diagram in figura

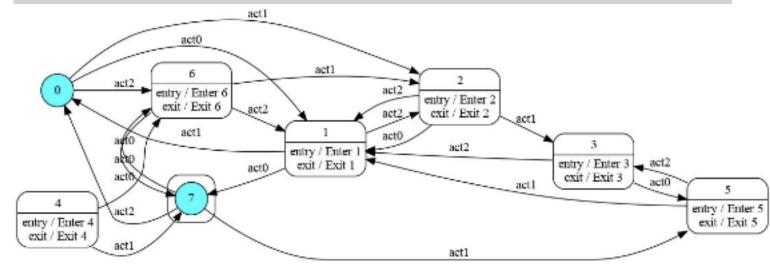
Si consideri il seguente insieme di test cases:

Test case 1: act1 act2 act2 act2 act2 act0 act2

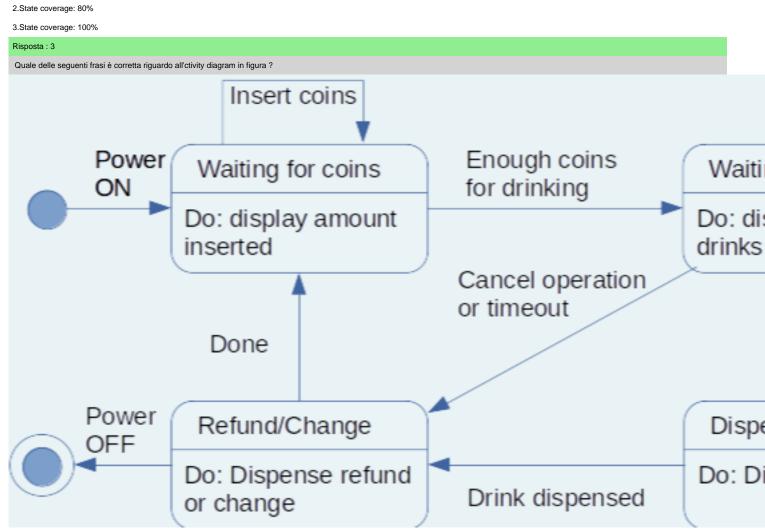
Test case 2: act2 act0 act0 act2 act0

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

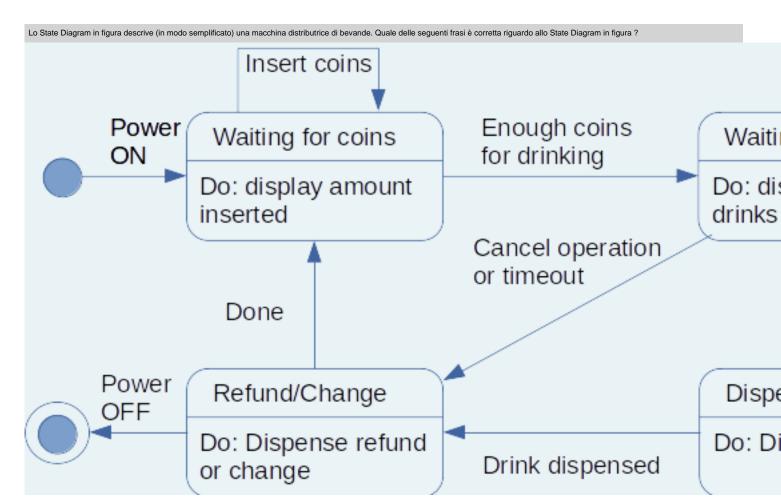


1.State coverage: 90%



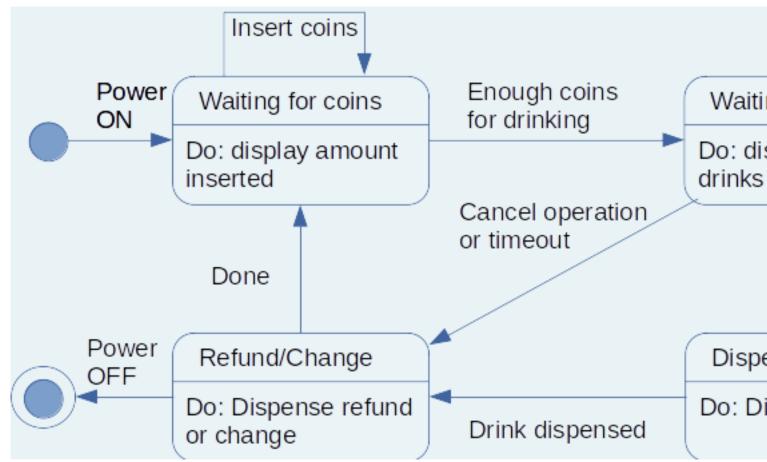
- 1.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in parallelo.
- 2. Una volta selezionato il piatto di mare da preparare, la stessa persona prepara prima il pesce e poi il contorno.
- 3. Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in sequenza.

Risposta : 1



- 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.

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 ?



- 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.

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.

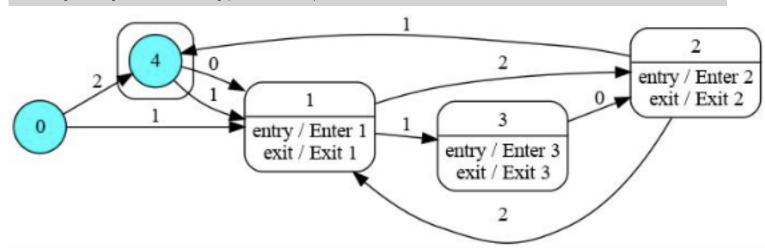
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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



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

Risposta: 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) raggiunti almeno una volta.

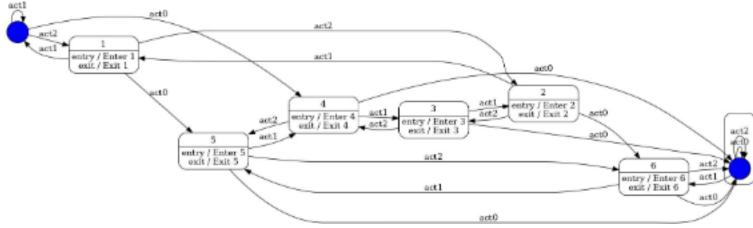
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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



1.State coverage: 100%

2.State coverage: 90%

3.State coverage: 70%

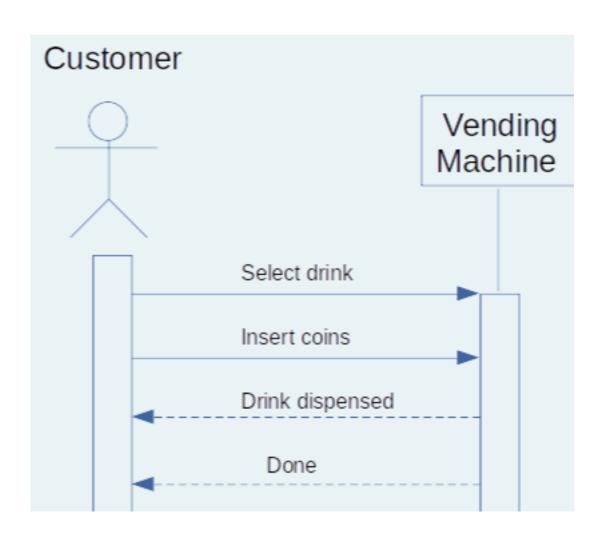
Risposta: 2

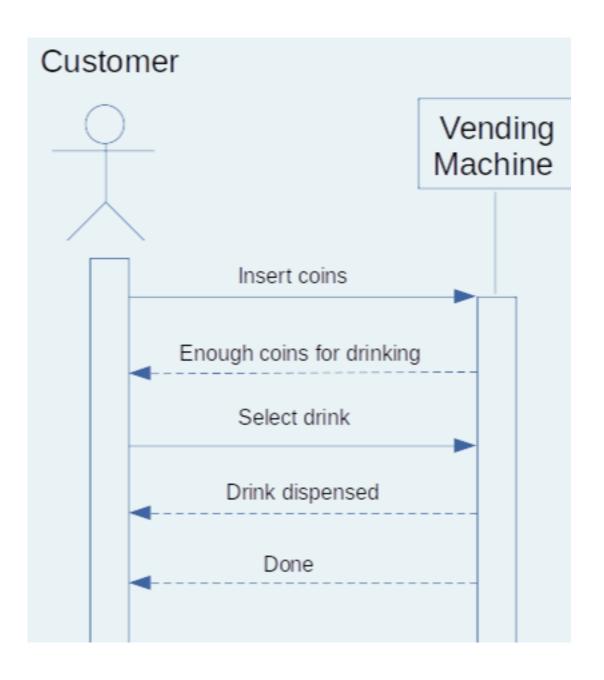
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?

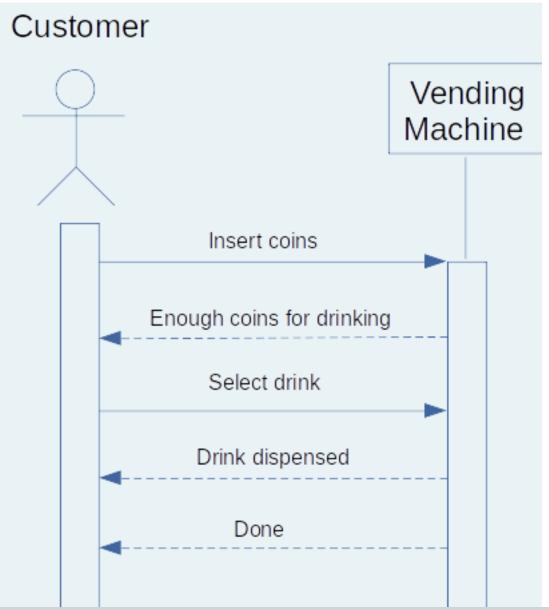
Web Browser

Forms and Query Manager

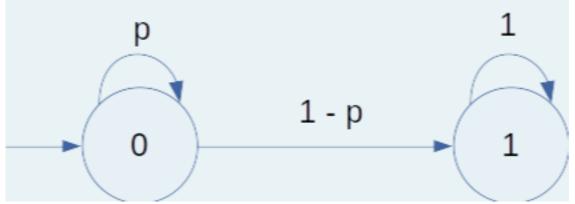
Database







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?



1.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: 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 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;

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

then // customer has inserted enough coins Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected (Customer2Machine == 1) state := 1: 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 Customer/Machine;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; $Machine 2 Customer := 1; \ else if (pre(state) == 1) \ and \ (Customer 2 Machine == 2) \ // \ drink \ selected$ then // drink selected state := 2; // dispensing drink $\label{eq:machine2Customer} \mbox{Machine2Customer} := 0; \ \mbox{elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction} \qquad \mbox{then // refund}$ 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 Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); state := 0; end if;end when;end

CoffeeMachine; Risposta : 2

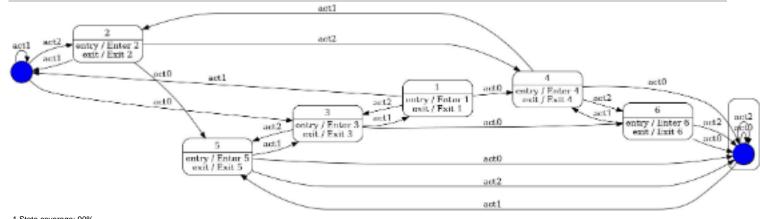
La state coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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



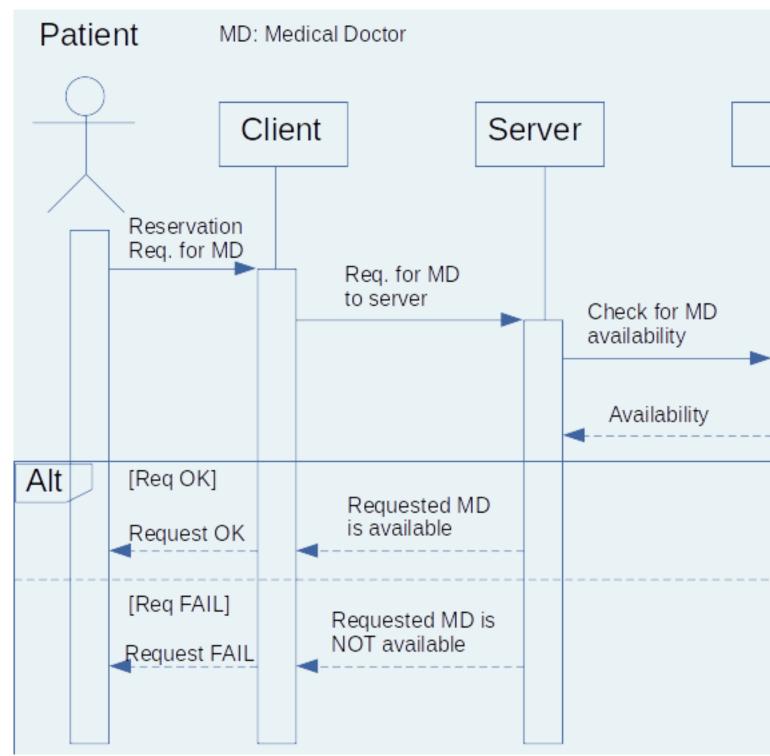
1.State coverage: 90%

2.State coverage: 70%

3.State coverage: 100%

Risposta: 1

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



- 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.

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.

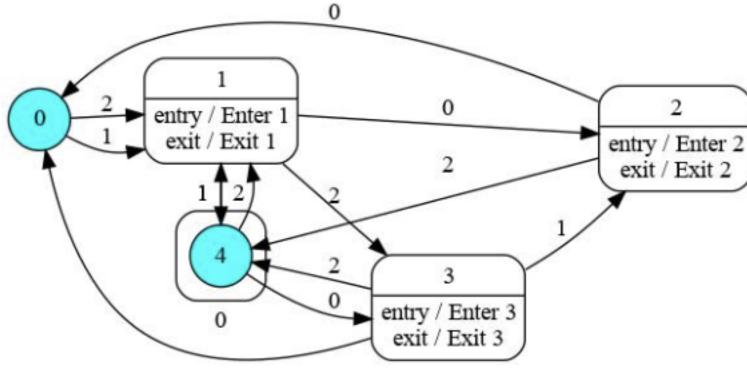
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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?



- 1.Transition coverage: 50%
- 2.Transition coverage: 100%
- 3.Transition coverage: 80%

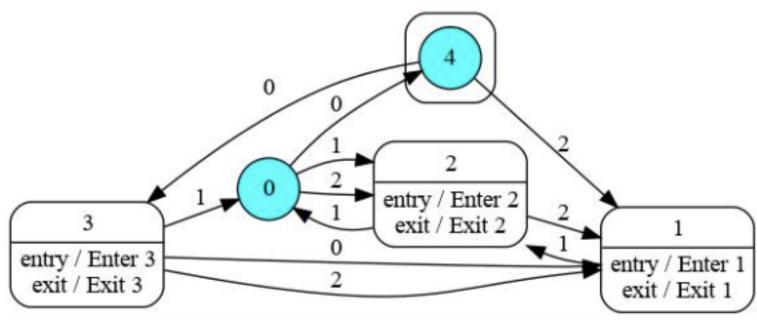
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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 act0Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?



- 1.Transition coverage: 60%
- 2.Transition coverage: 70%
- 3.Transition coverage: 100%

Risposta: 2

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.

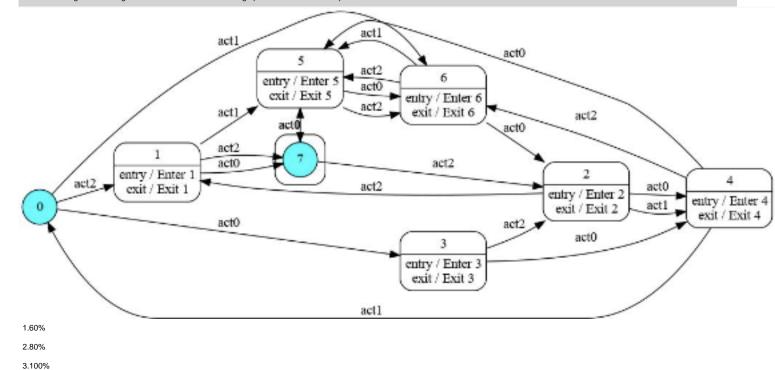
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

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

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

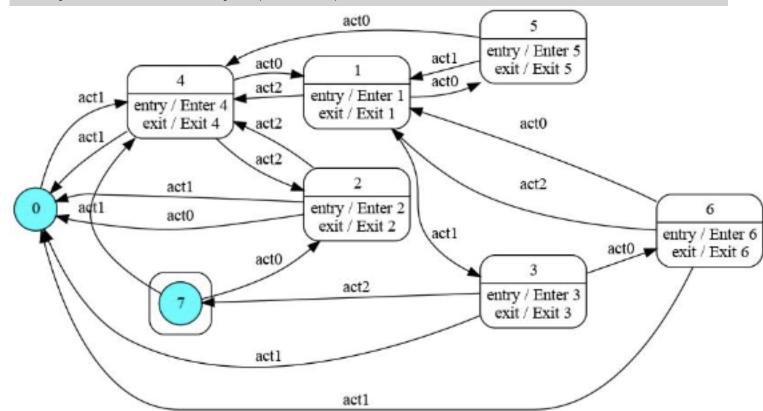
Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?



Risposta: 2

Si consideri l'automa segunete:

Quale dei seguenti modelli Modelica fornisce un modello ragionevole per l'automa di cui sopra.



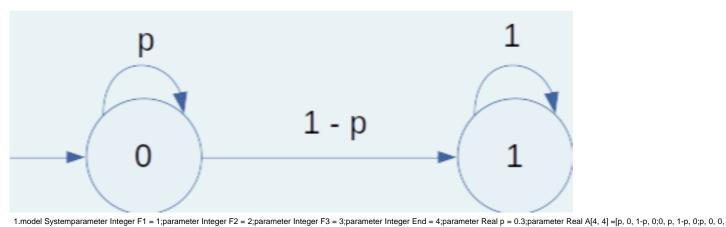
 $1. function next input Integer x; output Integer y; algorithm \quad y := x; end next; class SystemInteger x; initial equation x = 0; equation when sample (0, 1) then \quad x = next (pre(x)); end when; end System; the properties of the$

 $2. function\ next input\ Integer\ x; output\ Integer\ y; algorithm \quad y:=1-x; end\ next; class\ SystemInteger\ x; initial\ equation x=0; equation when\ sample(0,\ 1)\ then \quad x=next(pre(x)); end\ when; end\ System; algorithm of the property of the proper$

3.function nextinput Integer x;output Integer y;algorithm y := 1 + x;end next;class SystemInteger x;initial equationx = 0;equationwhen sample(0, 1) then x = next(pre(x));end when;end System;

Si consideri la seguente Markov Chain:

Quale dei seguenti modelli Modelica fornisce un modello ragionevole per la Markov Chain di cui sopra?

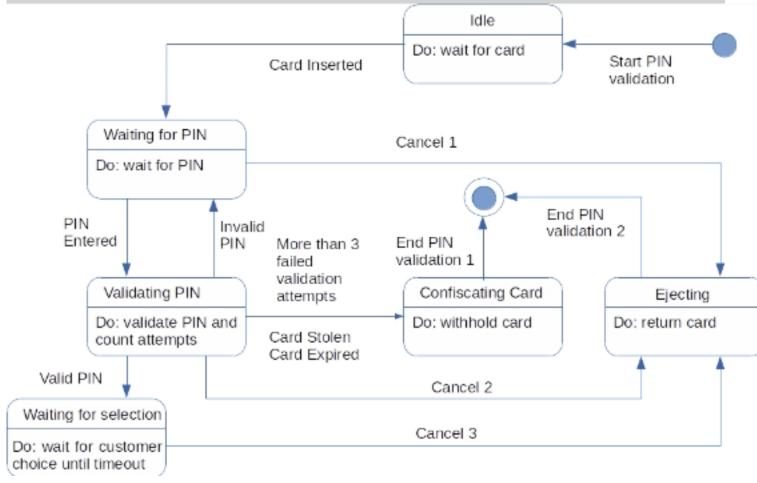


1];Integer Real r1024:Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen initial() thenstate1024 F1:r1024 Modelica, Math. Random, Generators, Xorshift 1024 star, initial State (614657. 30020):x 0:elsewhen sample(0,1) then(r1024.state1024) Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] A[x, F3]) then x := F3; else x := End; end if; end when; end System; 2.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 1-p, 0, 0;p, 0, 1-p, 0;p, 0, 0, 0, 0] $Real \qquad r 1024; Integer \qquad state 1024 [Modelica.Math.Random.Generators.Xorshift 1024 star.nState]; algorithm when \qquad initial () \qquad then \qquad r 1024; Integer \qquad r$ 1-p;0, 0, 0, 1];Integer x; state1024 := Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020); x := F1; r1024 := 0;elsewhen sample(0,1) then(r1024,state1024) := Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] A[x, F3]) then x := F3; else x := End; end if; end when; end System; 3.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 0, 1-p, 0;p, 1-p, 0, 0;p, 0, 0, 0, 0]

1-p;0, 0, 0, 1];Integer x; Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen initial() thenstate1024 := Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020);x := F1;r1024 := 0;elsewhen sample(0,1) then(r1024,state1024) := Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; else x := End;end if;end when;end System;

Si consideri la seguente Markov Chain:

Quale dei seguenti modelli Modelica fornisce un modello ragionevole per la Markov Chain di cui sopra?



2.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] =[0, 1, 0, 0;p, 1-p, 0, 0;0, 0, p, 1-p;0, 0, 0, 0, 1];Integer x; Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen initial() thenstate1024 := Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020);x := F1;r1024 := 0;elsewhen sample(0,1) then(r1024,state1024) := Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; else x := End;end if;end when;end System;

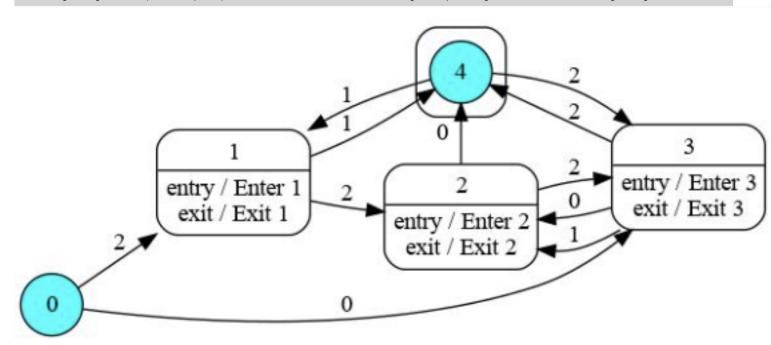
3.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] =[0, 1, 0, 0;p, 0, 0, 1-p;0, 0, p, 1-p;0, 0, p, 1-p;0, 0, 0, 1];Integer x; Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen initial() thenstate1024 := Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020);x := F1;r1024 := 0;elsewhen sample(0,1) then(r1024,state1024) := Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; else x := End:end if:end when;end System;

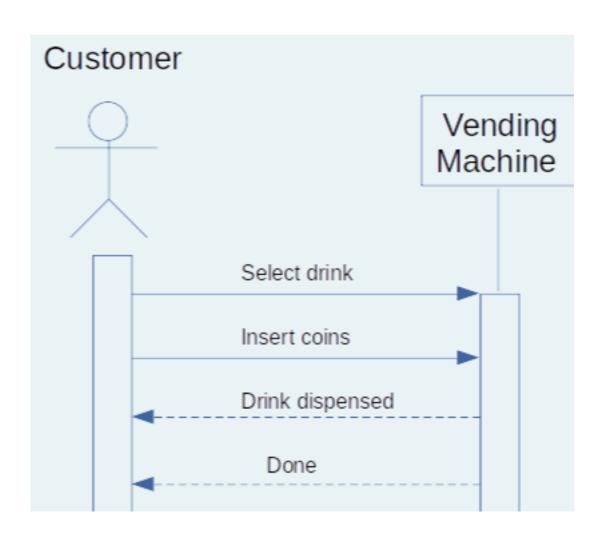
Quale delle seguenti frasi è corretta riguardo all'ctivity diagram in figura?

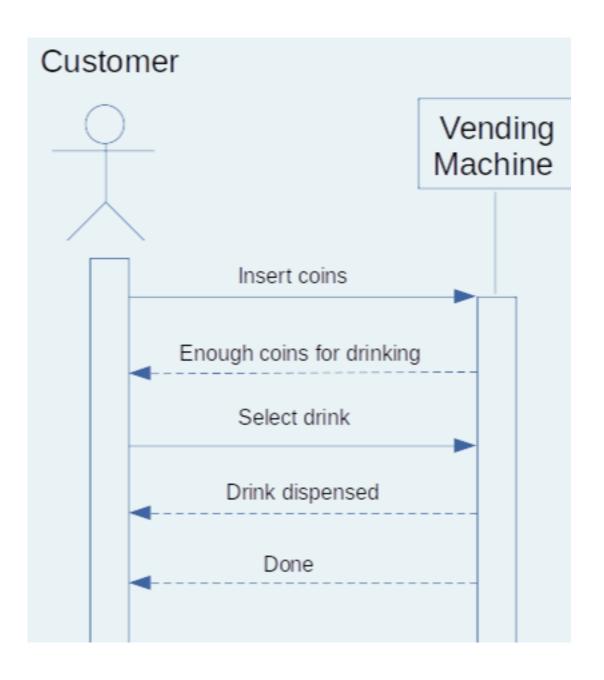
- 1.Una volta selezionato il piatto di mare da preparare, la stessa persona prepara prima il pesce e poi il contorno.
- 2. Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in parallelo.
- 3.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in sequenza.

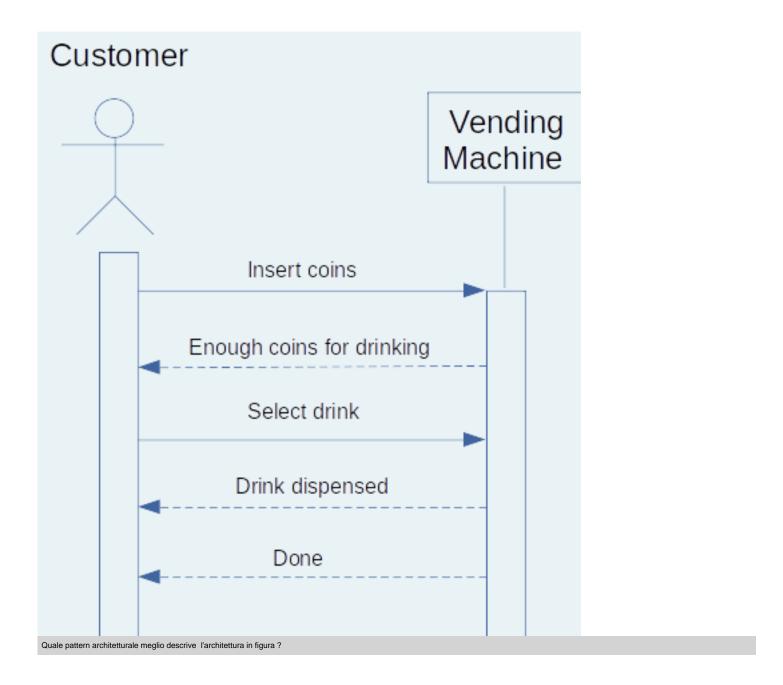
Risposta: 2

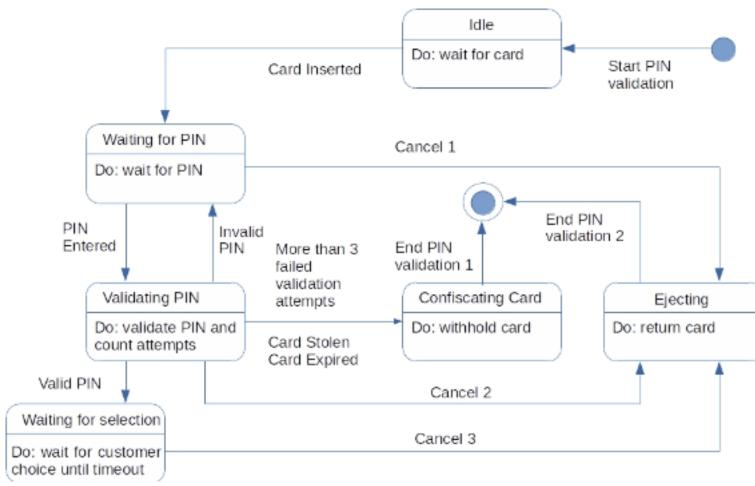
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?





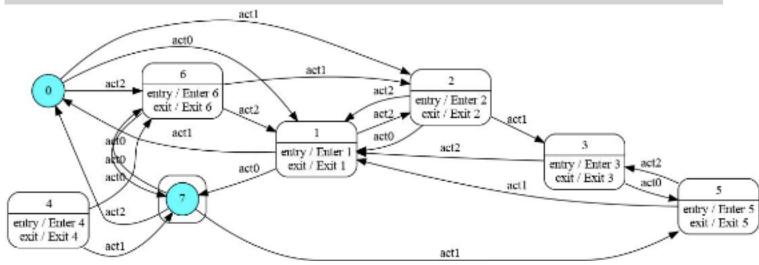






- 1.Pipe and filter architecture.
- 2.Model View Controller.
- 3.Layred architecture.

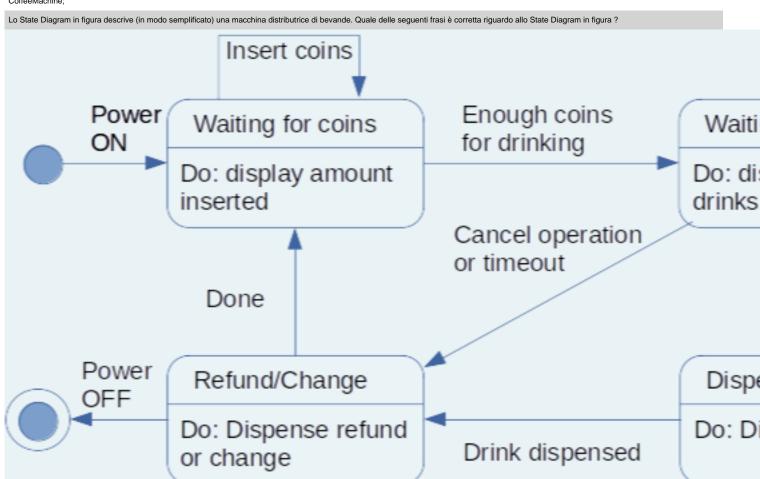
Lo state diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti modelli Modelica è plausibile per lo state diagram in figura?



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) $Machine 2 Customer := 1; \;\; elseif \; (pre(state) == 1) \;\; and \;\; (Customer 2 Machine == 2) \;\; // \; drink \; selected$ then // customer has inserted enough coins state := 1: then // drink selected Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction state := 2; // dispensing drink then // refund then // drink dispensed 3: // refund/change Machine2Customer := 0: elseif (pre(state) == 2) // drink dispensed Machine2Customer := 2;elseif (pre(state) == 3) // 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

 $Machine 2 Customer := 1; \;\; elseif \; (pre(state) == 1) \;\; and \;\; (Customer 2 Machine == 2) \;\; // \; drink \; selected$ (Customer2Machine == 1) then // customer has inserted enough coins state := 1: then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund 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); 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 then // customer has inserted enough coins Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction 3: // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed Machine2Customer := 2:elseif (pre(state) == 3) // refund/change Machine2Customer := pre(Machine2Customer); CoffeeMachine;



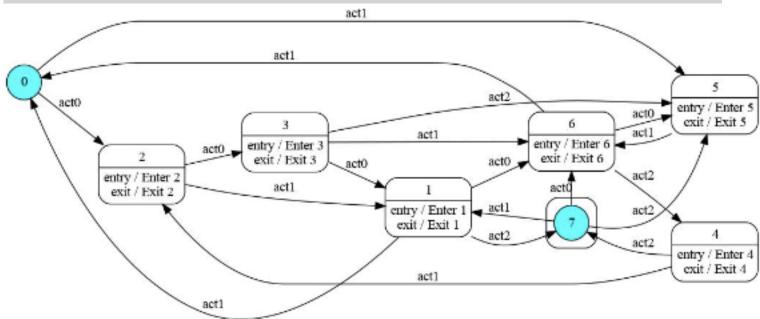
- .Una volta inserite le monete bisogna necessariamente consumare almeno una bevanda.
- 2. Anche se ci sono abbastanza monete potrebbe non essere possibile ottenere la bevanda selezionata.
- 3.Se ci sono abbastanza monete è sempre possibile ottenere la bevanda selezionata.

Risposta: 3

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

- 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.

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 ?

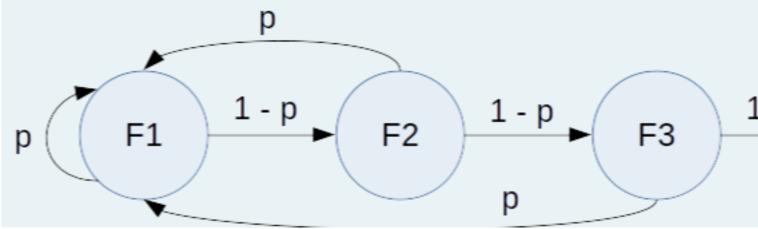


- 1.Una volta selezionata la bevanda non è possibile cancellare l'operazione.
- 2.La macchina non dà resto.
- 3.Una volta inserite monete per due bevande è possibile ottenerle senza reinserire le monete.

Risposta: 1

Si consideri la seguente Markov Chain:

Quale dei seguenti modelli Modelica fornisce un modello ragionevole per la Markov Chain di cui sopra?



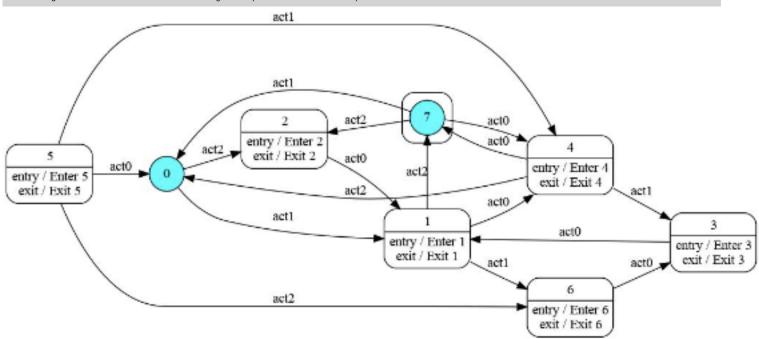
1.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 0, 1-p, 0;0, p, 1-p, 0;p, 0, 0, 1-p:0. 0, 0, 1];Integer Real r1024;Integer state 1024 [Modelica. Math. Random. Generators. Xorshift 1024 star.n State]; algorithm when the properties of the propinitial() thenstate1024 X; Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020);x F1;r1024 := 0;elsewhen sample(0,1) then(r1024,state1024) Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A A[x, F3]) then x := F3; else x := End; end if; end when; end System;

2.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 0, 1-p, 0;p, 1-p, 0, 0;p, 0, 0, 0, 0] Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen initial() 1-p;0, 0, 0, 1];Integer x; thenstate1024 Modelica, Math. Random, Generators, Xorshift 1024 star, initial State (614657). 30020):x F1:r1024 0:elsewhen sample(0.1) then(r1024.state1024) := := Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2]) A[x, F3]) then x := F3; else x := End;end if;end when;end System;

3.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 1-p, 0, 0;p, 0, 1-p, 0;p, 0, 0, 0, 0, 0, 0]; Integer x; Real r1024; Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState]; algorithmwhen initial() then state1024 := Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020); x := F1; r1024 := 0; elsewhen sample(0,1) then(r1024, state1024) := Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024)); if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; else x := End; end if; end when; end System;

Si consideri la seguente Markov Chain:

Quale dei seguenti modelli Modelica fornisce un modello ragionevole per la Markov Chain di cui sopra?



1.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real p = 0.3;param

Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020);x F1;r1024 0;elsewhen sample(0,1) then(r1024,state1024) Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2 A[x, F3]) then x := F3; else x := End;end if;end when;end System; 3.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real p = 0.3;parameter Real p = 0.3;parameter Real p = 0.4;parameter Real p = 0.4 1];Integer Real r1024;Integer state 1024 [Modelica. Math. Random. Generators. Xorshift 1024 star.n State]; algorithm when the properties of the propthenstate1024 Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020);x F1;r1024 0;elsewhen sample(0,1) then(r1024,state1024) Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2 A[x, F3]) then x := F3; else x := End; end if; end when; end System;

state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen

initial()

thenstate1024

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:

Real

r1024;Integer

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

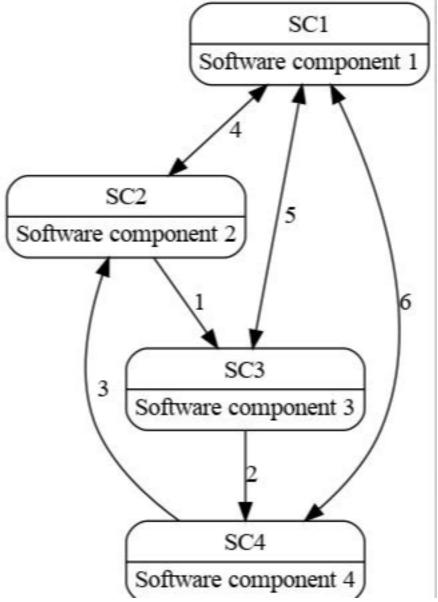
Test case 3: act1 act1 act2 act0 act1 act0

1-p;0,

0, 0,

1];Integer

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



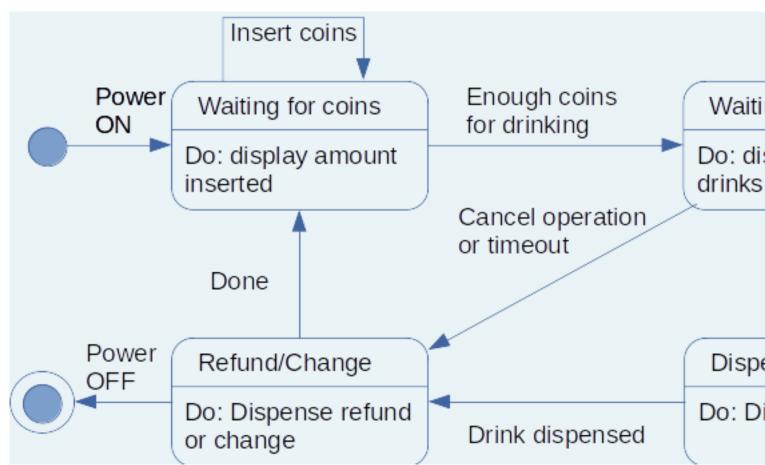
1.State coverage: 60%

2.State coverage: 75%

3.State coverage: 100%

Risposta : 2

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



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) == 1) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 3) and (pre(u) == 1) then x := 2; elseif (pre(x) == 3) and (pre(u) == 1) then x := 2; elseif (pre(x) == 3) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 4) an

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() 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) == 1) and (pre(u) == 1) then x := 2; 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) == 2) then x := 4; 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) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 3) then x := 2; elseif (pre(x) == 3) and (pre(u) == 3) then x := 4; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 3) and (pre(u) == 3) then x := 4; elseif (pre(x) == 4) and (pre(u) == 3) then x := 2; elseif (pre(x) == 3) and (pre(u) == 3) then x := 4; elseif (pre(x) == 4) and (pre(u) == 3) then x := 2; elseif (pre(x) == 4) and (pre(u) == 3) then x := 2; elseif (pre(x) == 4) and (pre(u) == 3) then x := 4; elseif (pre(x) == 4) and (pre(u) == 4) then x := 4; elseif (pre(x) == 4) and (pre(u) == 4) then x := 4; elseif (pre(x) == 4) and (pre(u) == 4) then x := 4; elseif (pre(x) == 4) then x := 4;

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() then x := 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) == 1) and (pre(u) == 2) then x := 2;elseif (pre(x) == 1) and (pre(u) == 2) then x := 2;elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;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) == 4) and (pre(u) == 4) and (pre(u) == 1) then x := 1;elseif (pre(x) == 4) and (pre(u) == 5) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) then x := 1;elseif (pre(x) == 4) then x := 1

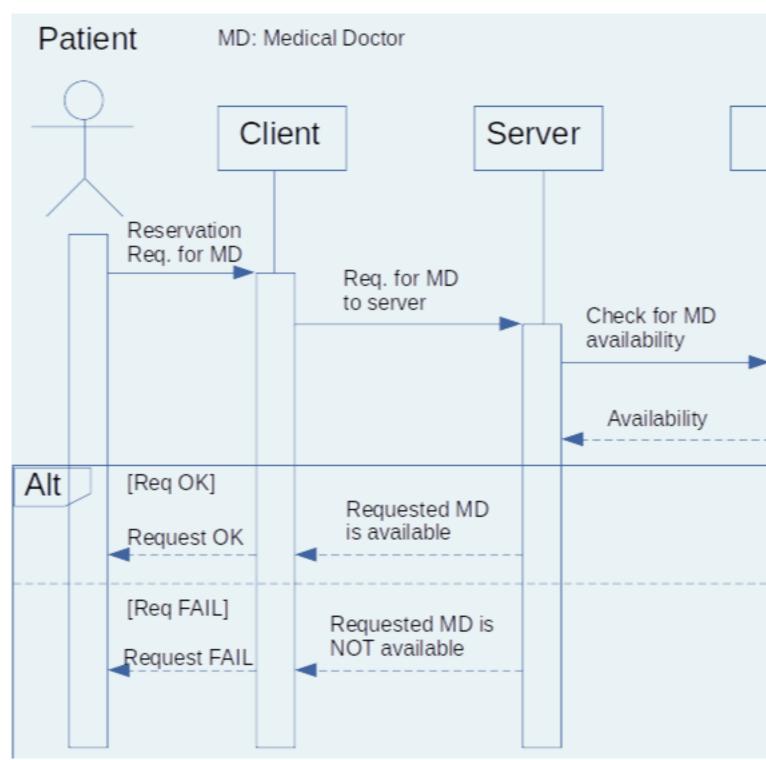
Risposta: 3

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

Uno scenario è una sequenza di stati.

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



1.0.56

2.0.24 3.0.14

Risposta: 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.

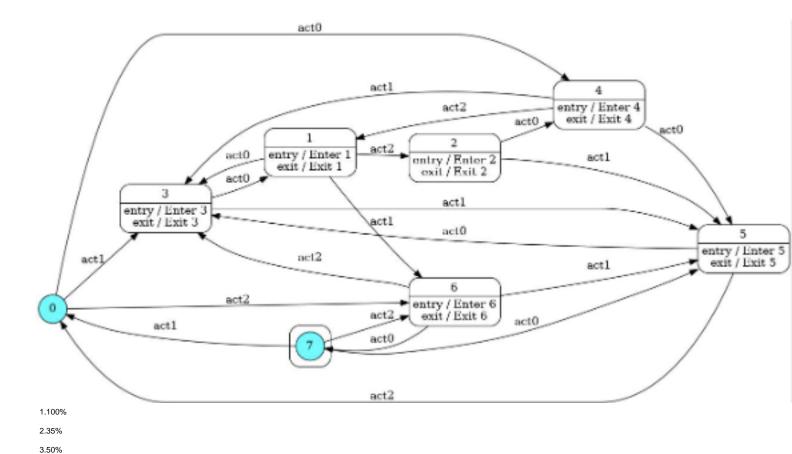
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act2 act0

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

Test case 3: act2 act0

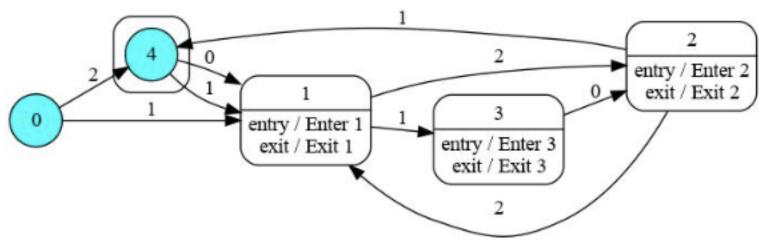
Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?



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 2: act0



1.50%

2.90%

3.100%

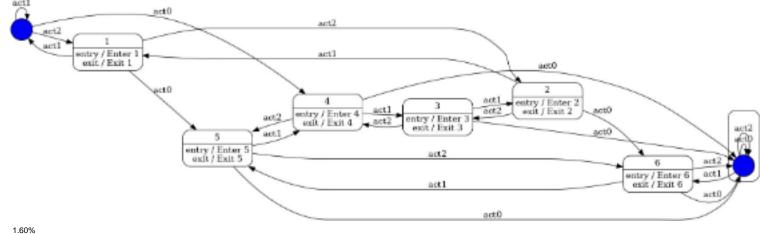
Risposta: 2

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 act2 act1 act2 act1 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?



2.80%

3.100%

Risposta: 1

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.

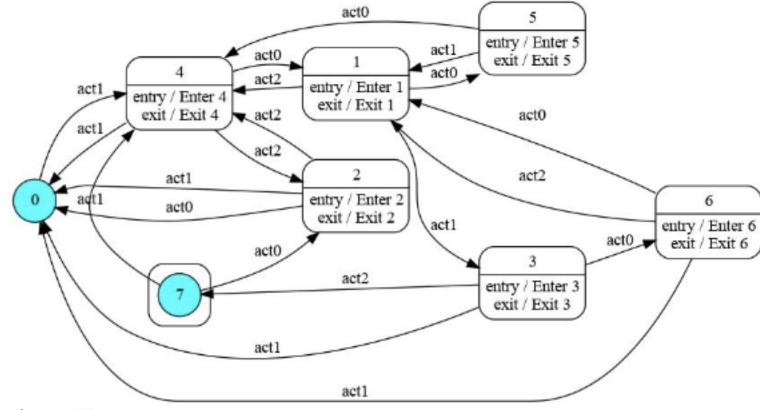
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

Test case 1; act1 act2 act2 act1 act2 act1 act1 act1 act0 act1 act2 act0 act1 act2 act1 act2 act1 act2 act0 act2 act2 act0 act2 act0 act1 act2 act2 act0 act1

Test case 2; act1 act2 act0 act0 act2 act2 act2 act2 act2 act1 act2 act0 act0 act2 act1 act2 act2 act0 act0 act0 act0 act0 act0 act0

Test case 3: act1 act1

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



- 1.State coverage: 60%
- 2.State coverage: 85%
- 3.State coverage: 100%

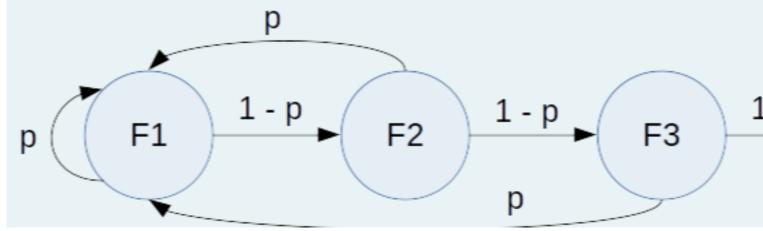
Risposta: 2

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 figuraRappresenta 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)?



1.0.08

2.0.12

3.0.32

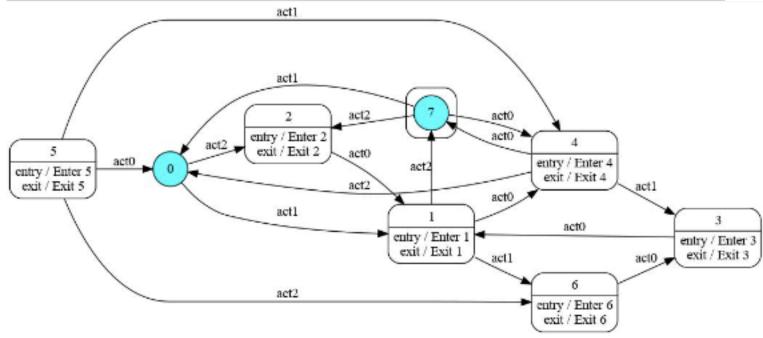
Risposta : 2

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

Uno scenario è una sequenza di stati.

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



1.0.12

2.0.28

3.0.42

Risposta: 2

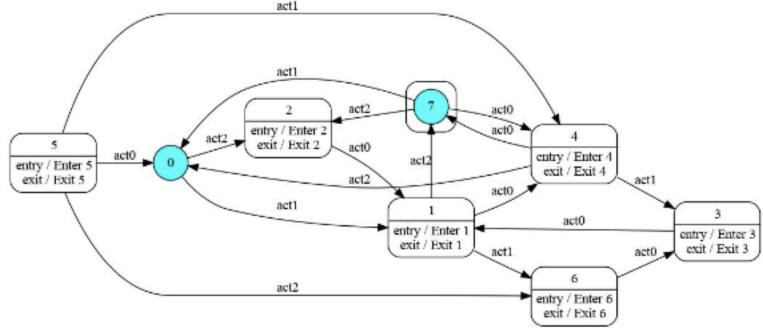
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il costo dello stato (fase) x è c(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta il completamento 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 $X = x(0), x(1), x(2), \\ e C(X) = c(x(0)) + c(x(1)) + c(x(2)) + ...$

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



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

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

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

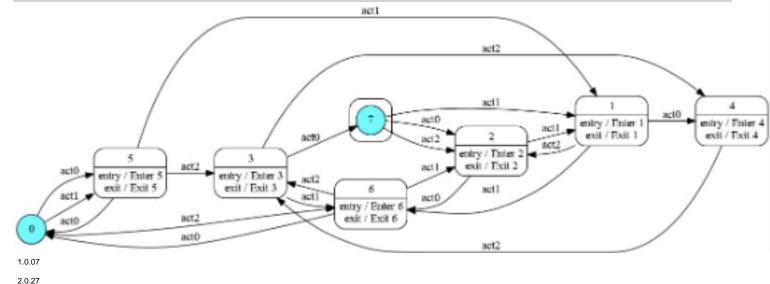
Risposta: 2

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

Uno scenario è una sequenza di stati.

Qual'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 ?



Risposta : 3

3.0.03

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.

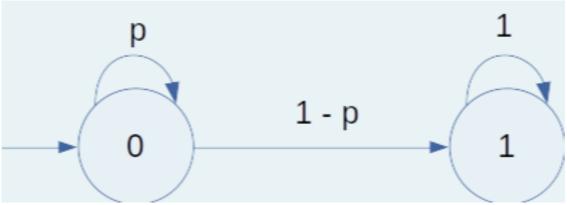
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act0

Test case 2: act1 act0 act2 act2 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?



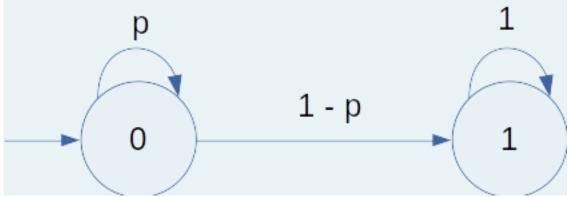
1.60%

2.90%

3.75%

Risposta: 1

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



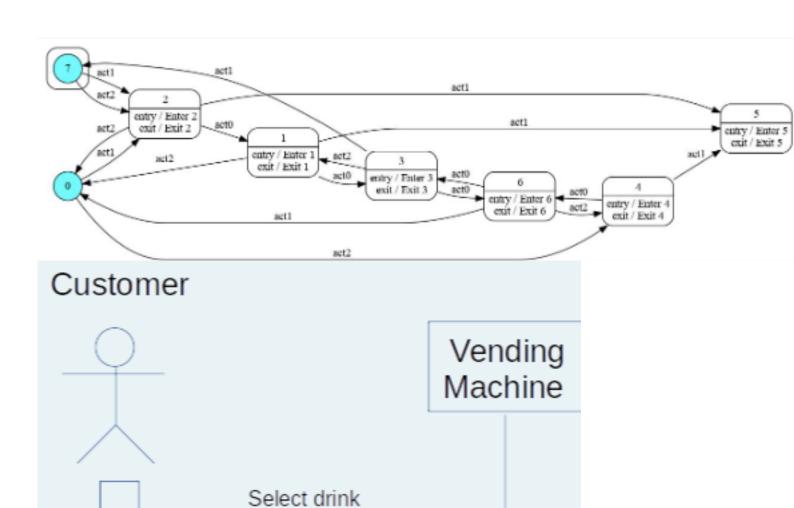
1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer; connector OutputInteger = output Integer; // InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1; algorithmwhen initial() then x := 0; elsewhen sample(0,T)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

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() then x := 0; elsewhen sample(0,T) then if (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 := 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) == 4) and (pre(u) == 0) then x := 3; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (

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

Risposta: 2

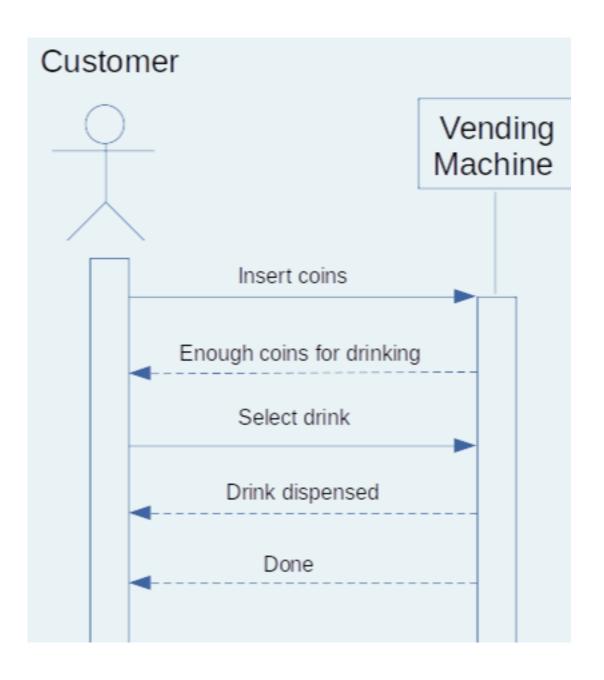
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?

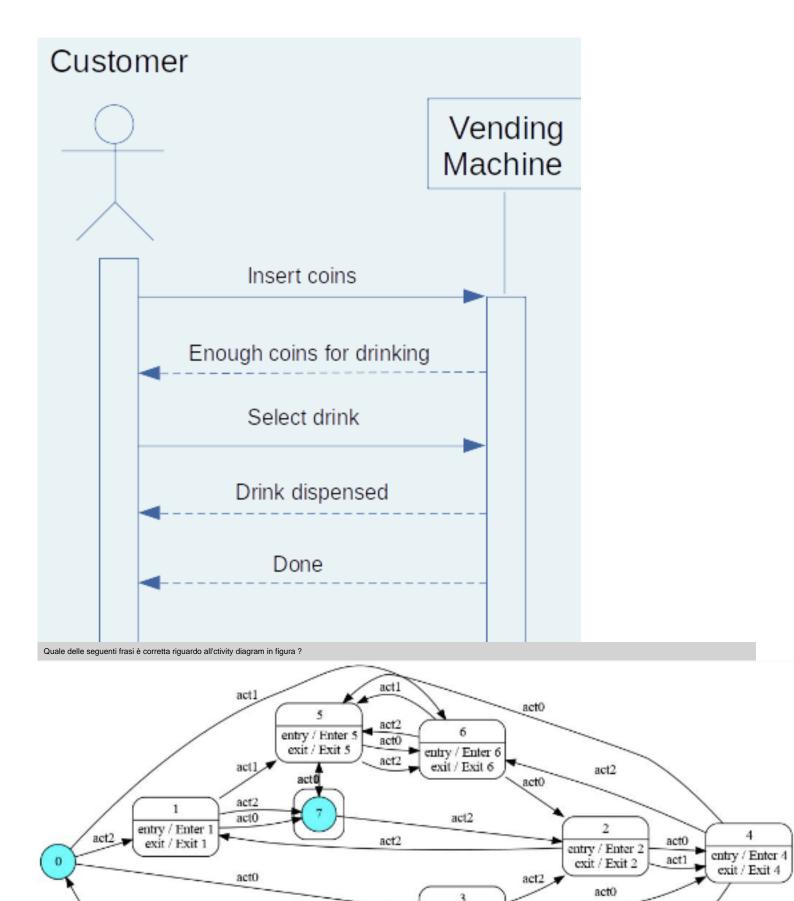


Insert coins

Done

Drink dispensed





entry / Enter 3 exit / Exit 3

- 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.



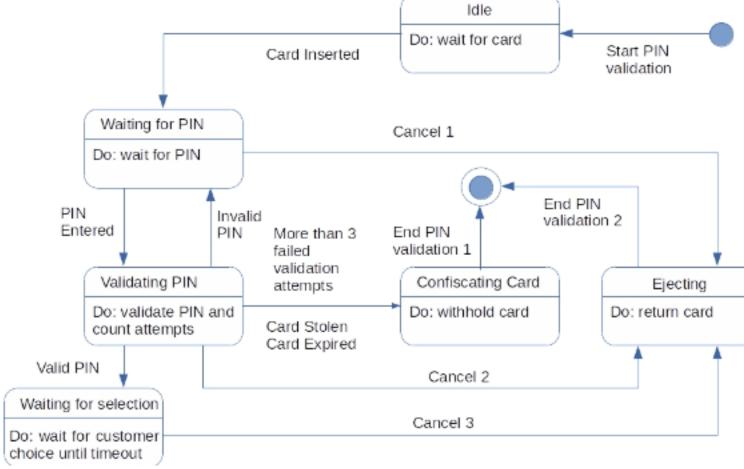
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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 act1 act1 act2 act1 act2 act0 act0 act0 act0

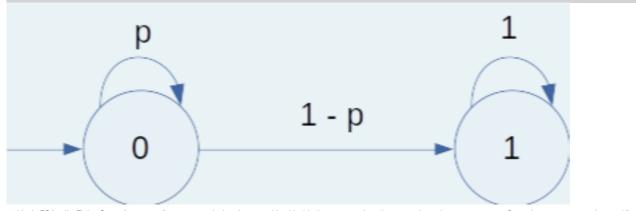
Test case 3: act2 act0 act1 act2 act2 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



- 1.State coverage: 60%
- 2.State coverage: 75%
- 3.State coverage: 90%



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



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() 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 := 2; elseif (pre(x) == 1) and (pre(u) == 2) then x := 0; elseif (pre(x) == 1) and (pre(u) == 2) then x := 0; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 3) an

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

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() 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 := 4; elseif (pre(x) == 2) and (pre(u) == 0) then x := 1; elseif (pre(x) == 0) then x := 1; elseif (pre(x) == 3) and (pre(u) == 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) == 2) then x := 1; elseif (pre(x) == 4) and (

Risposta: 3

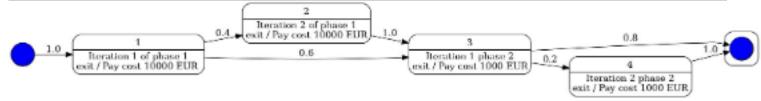
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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

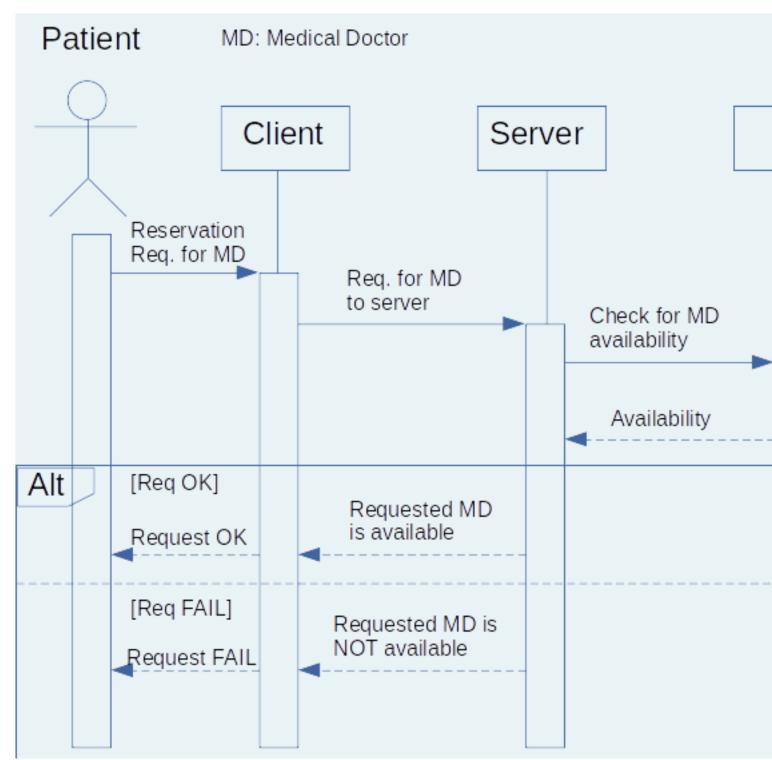
Test case 3: act2 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



- 1.State coverage: 90%
- 2.State coverage: 60%
- 3.State coverage: 75%

Risposta: 3

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



^{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.

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

^{2.}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.

^{3.} Periodicamente il client comunica ai pazienti le disponibilità dei medici.

exit / Exit 1

act1

act2

act2

entry / Enter 4 exit / Exit 4

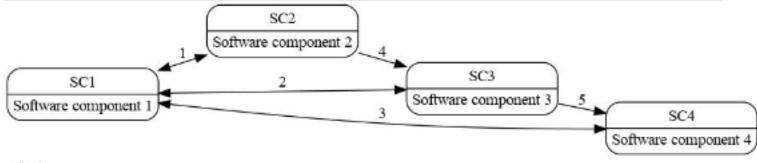
1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer; connector OutputInteger = output Integer; // InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1; algorithmwhen initial() thenx := 0; elsewhen sample(0,T)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

Risposta: 2

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.



1.1/(1 - p)

2.1/(p*(1 - p))

3.(1 - p)/p

Risposta: 1

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.

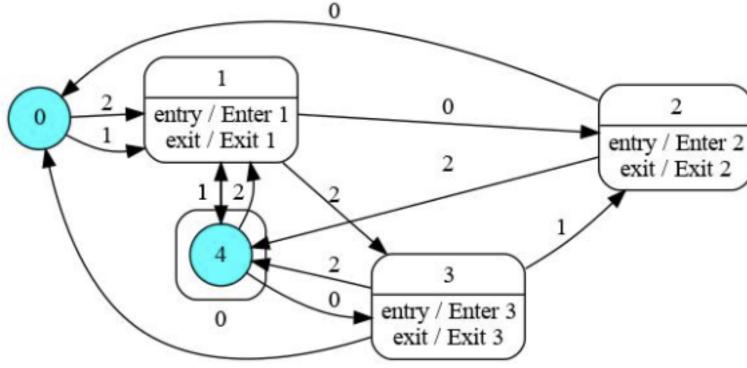
Il tempo di una istanza del processo software descritto sopra è la somma dei tempi degli stati (fasi) attraversati (tenendo presente che si parte sempre dallo stato 0.

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)) + ...

act1

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



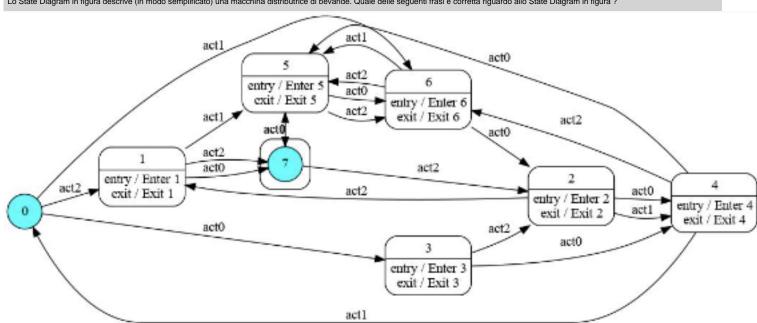
1.time(0)/(p*(1 - p))

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

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

Risposta: 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 ?



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

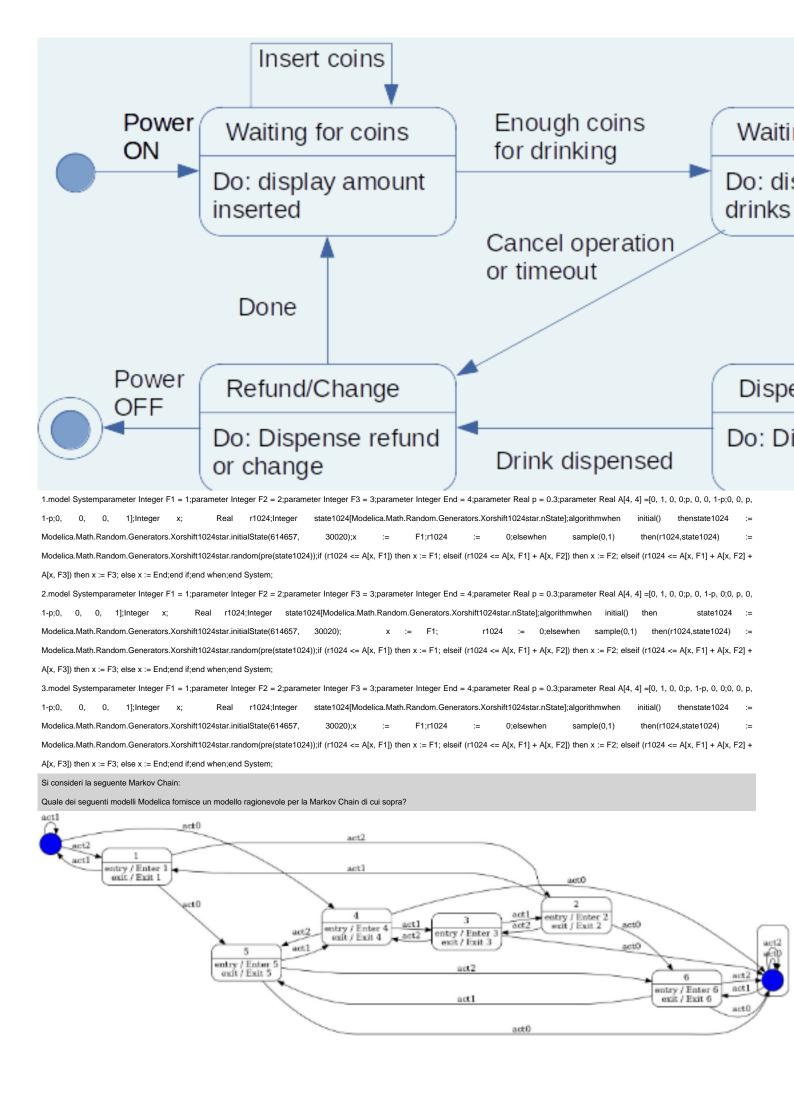
2. Anche se ci sono abbastanza monete potrebbe non essere possibile ottenere la bevanda selezionata.

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

Risposta: 3

Si consideri la seguente Markov Chain:

Quale dei seguenti modelli Modelica fornisce un modello ragionevole per la Markov Chain di cui sopra?



1.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 1-p, 0, 0;p, 0, 1-p, 0;p, 0, 0, 0, 0, 0, 0] $Real \quad r1024; Integer \quad state 1024 [Modelica.Math.Random.Generators.Xorshift 1024 star.nState]; algorithm when \quad initial () \quad then \quad r1024 [Modelica.Math.Random.Generators.Xorshift 1024 star.nState]; algorithm when \quad r1024 [Modelica.Math.Random.Generators.Xorshift 1024 star.nStar$ 1-p;0, 0, 0, 1];Integer x; state1024 := Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020); x := F1; r1024 := 0;elsewhen sample(0,1) then(r1024,state1024) := Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2]) A[x, F3]) then x := F3; else x := End; end if; end when; end System; 2.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 0, 1-p, 0;p, 1-p, 0, 0;p, 0, 0, 0, 0] 1];Integer x; Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen initial() thenstate1024 Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020);x F1;r1024 0;elsewhen sample(0,1) then(r1024,state1024) Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; elseif (r1024 <= A[x, F1] + A[x, F2] + A[A[x, F3]) then x := F3; else x := End; end if; end when; end System; 3.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 0, 1-p, 0;0, p, 1-p, 0;p, 0, 0, 0, 0] 0, 0, 1];Integer Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen initial() thenstate1024 1-p;0, x; Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020);x F1;r1024 := 0;elsewhen sample(0,1) then(r1024,state1024) Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2])

Si consideri l'automa segunete:

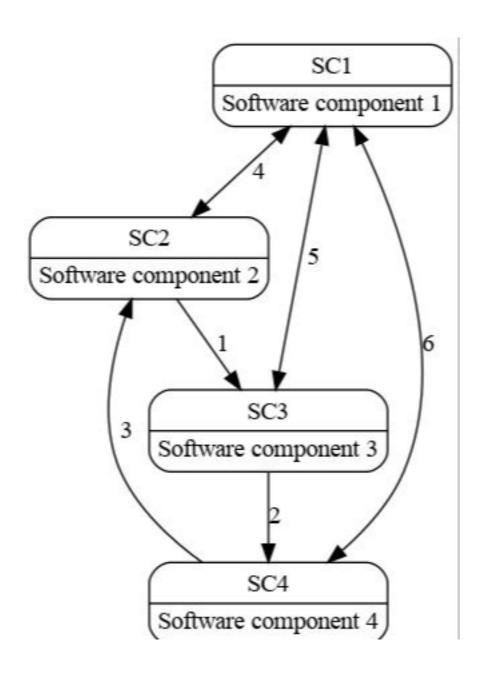
Quale dei seguenti modelli Modelica fornisce un modello ragionevole per l'automa di cui sopra.

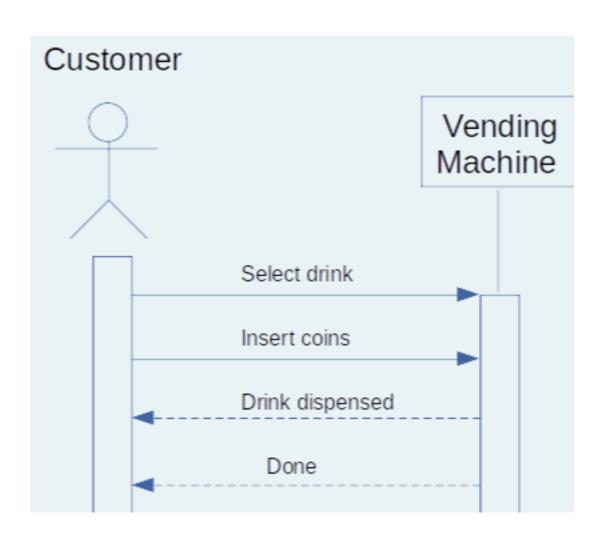
A[x, F3]) then x := F3; else x := End; end if; end when; end System;

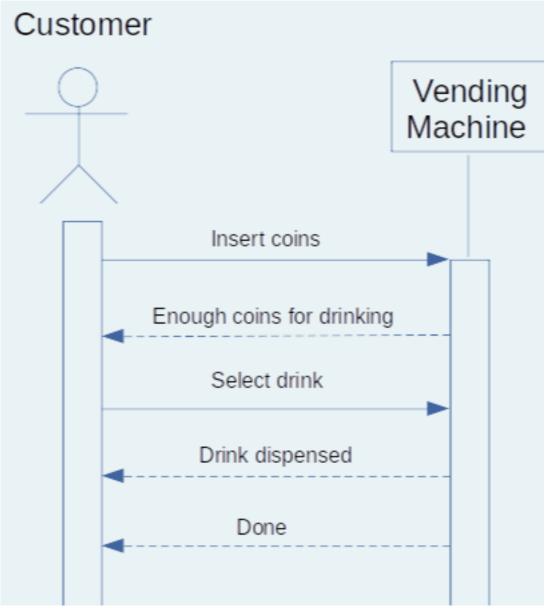


1.function nextinput Integer x;output Integer y;algorithm y := x;end next;class SystemInteger x;initial equationx = 0;equationwhen sample(0, 1) then x = next(pre(x));end when;end System; 2.function nextinput Integer x;output Integer y;algorithm y := 1 + x;end next;class SystemInteger x;initial equationx = 0;equationwhen sample(0, 1) then x = next(pre(x));end when;end System; 3.function nextinput Integer x;output Integer y;algorithm y := 1 + x;end next;class SystemInteger x;initial equationx = 0;equationwhen sample(0, 1) then x = next(pre(x));end when;end System; x = next(pre(x));end when;end System;

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 ?







iVBORw0KGgoAAAANSUhEUgAAAZcAAAACCAYAAACUupFJAAAAKEIEQVRIiWN8+fnrf4ZRMApGwSgYBaOAioBpoB0wCkbBKBgFo2D4AQCkdAPUYeTFMQAAAABJRU5ErkJggg==

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.

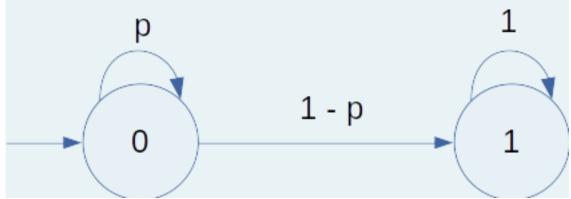
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

Test case 2: act0 act1 act2 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?



1.80%

2.60%

3.100%

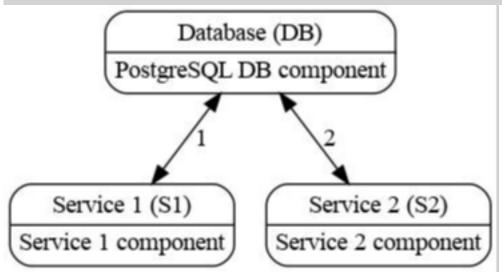
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act2 act2 act2 act0

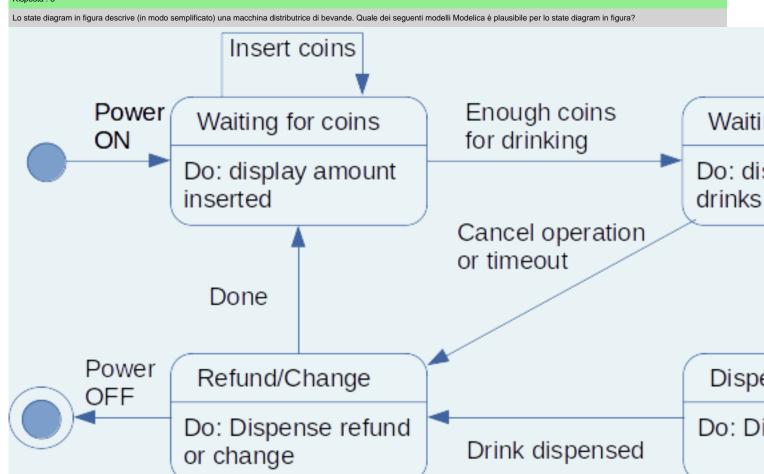
Test case 2: act0 act1 act2 act0 act2
Test case 3: act2 act2 act2 act2

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



- 1.Transition coverage: 100%
- 2.Transition coverage: 70%
- 3.Transition coverage: 40%





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

Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // drink selected state := 2; // dispensing drink 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 Customer/2Machine;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 0; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed Machine2Customer := 2;elseif (pre(state) == 3) // state := 3; Machine2Customer := 3; // done else state := pre(state); refund/change then // refund state := 0: Machine2Customer := pre(Machine2Customer); 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 then // drink dispensed Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed state := 3; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund Machine2Customer := 3: // done else state := pre(state): Machine2Customer := pre(Machine2Customer): end if:end when:end state := 0: CoffeeMachine:

Risposta: 3

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.

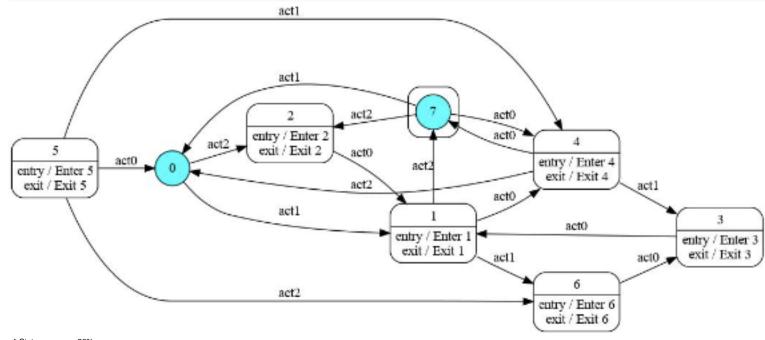
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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



1.State coverage: 90%

2.State coverage: 70%

3.State coverage: 100%

Risposta: 1

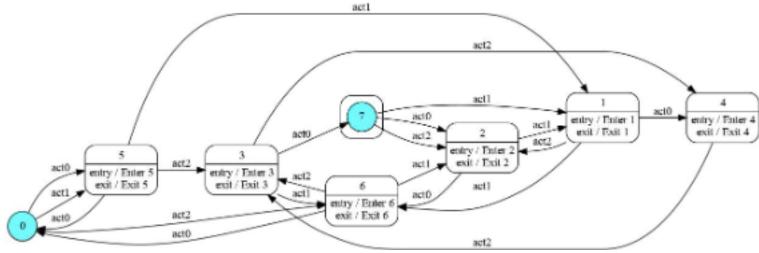
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta. Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act0 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?



- 1.Transition coverage: 100%
- 2.Transition coverage: 80%
- 3.Transition coverage: 50%

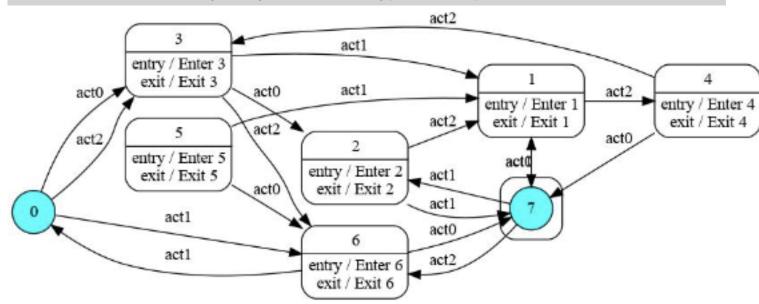
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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 act0Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?



- 1.Transition coverage: 100%
- 2.Transition coverage: 70%
- 3.Transition coverage: 60%

Risposta: 2

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

Web Browser

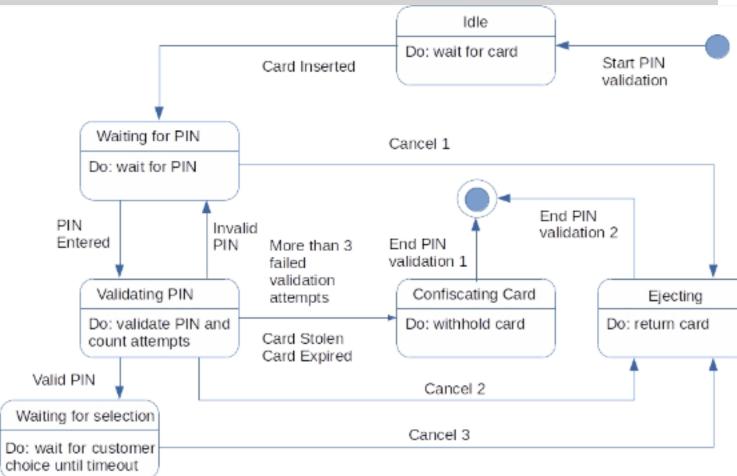
Forms and Query Manager

Database

- 1.State coverage: 90%
- 2.State coverage: 80%
- 3.State coverage: 100%

Risposta: 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?



- 1. Una volta inserite monete per due bevande è possibile ottenerle senza reinserire le monete.
- 2. Una volta selezionata la bevanda non è possibile cancellare l'operazione.
- 3.La macchina non dà resto.

Risposta: 2

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.

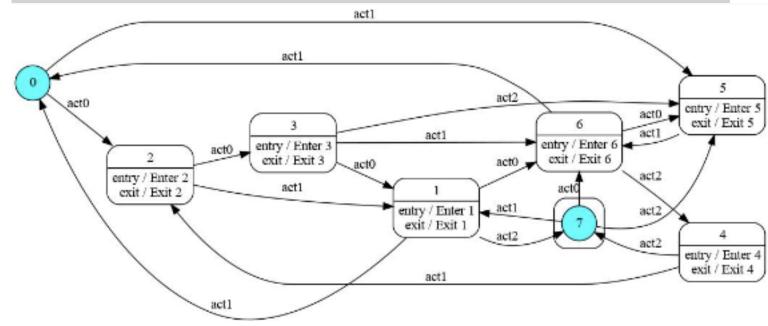
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

Test case 1: act1 act1 act0 act1 act1 act2 act0

Test case 2: act2 act0 act0

Test case 3: act1 act2 act0 act0

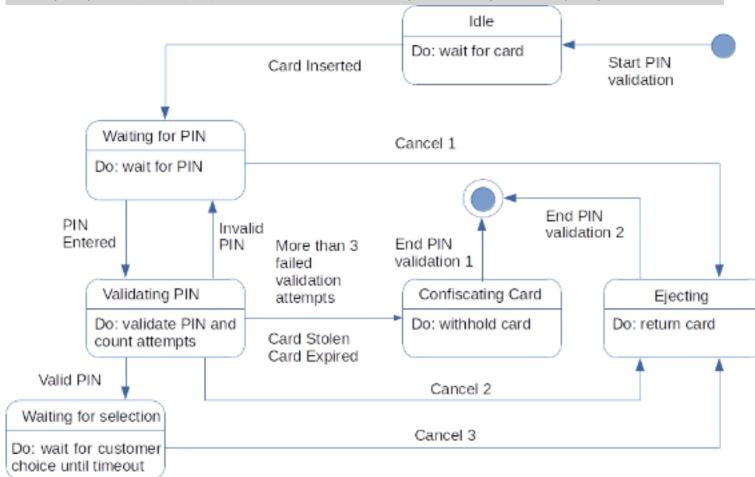
Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



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

Risposta: 1

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 ?

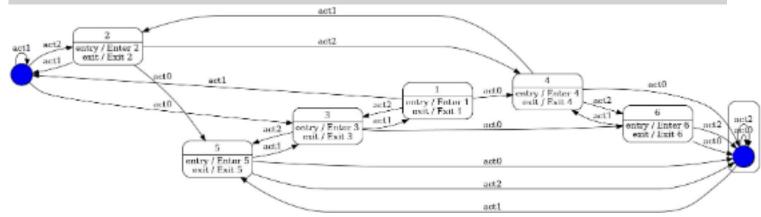


- 1. Anche se ci sono abbastanza monete potrebbe non essere possibile ottenere la bevanda selezionata.
- 2. Una volta inserite le monete bisogna necessariamente consumare almeno una bevanda.

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

Risposta: 3

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



- 1.II 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

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.

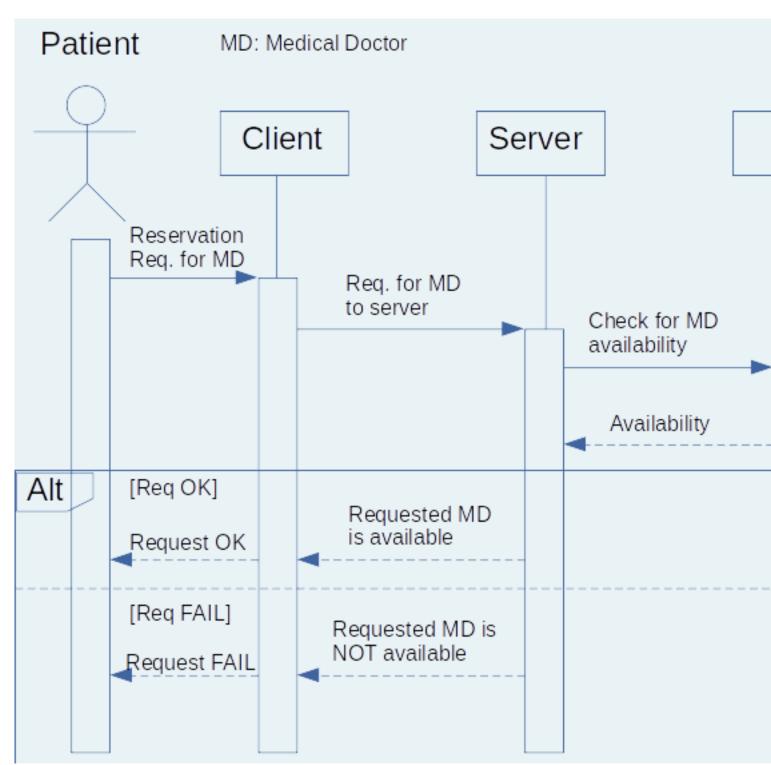
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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



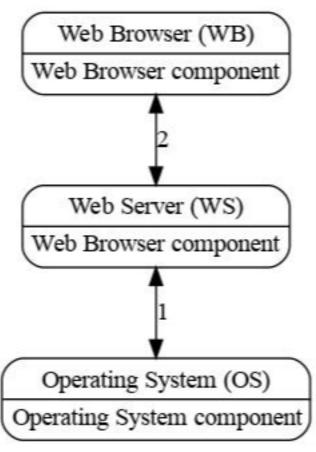
1.State coverage: 75%

2.State coverage: 100%

3.State coverage: 80%

Risposta: 1

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



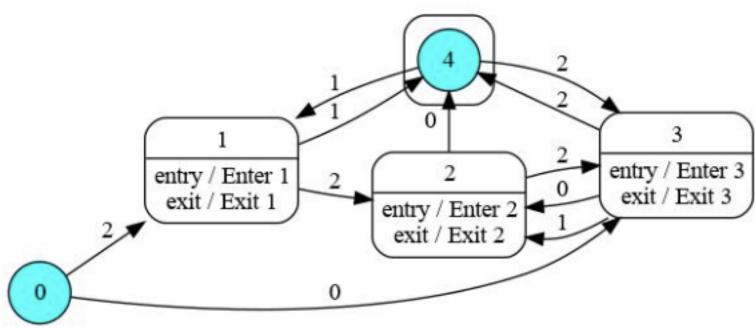
- 1.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in sequenza.
- 2.Una volta selezionato il piatto di mare da preparare, la stessa persona prepara prima il pesce e poi il contorno.
- 3.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in parallelo.

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

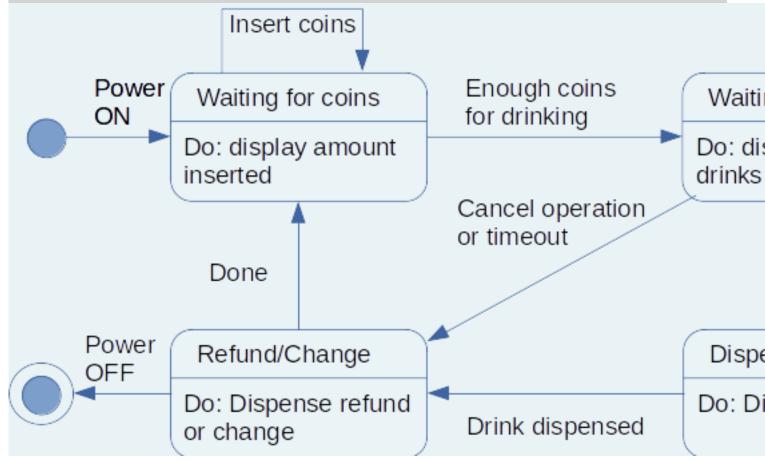
Uno scenario è una sequenza di stati.

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



1.0.28

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.



1.1/(1 - p)

2.(1 - p)/p 3.1/(p*(1 - p))

Risposta: 1

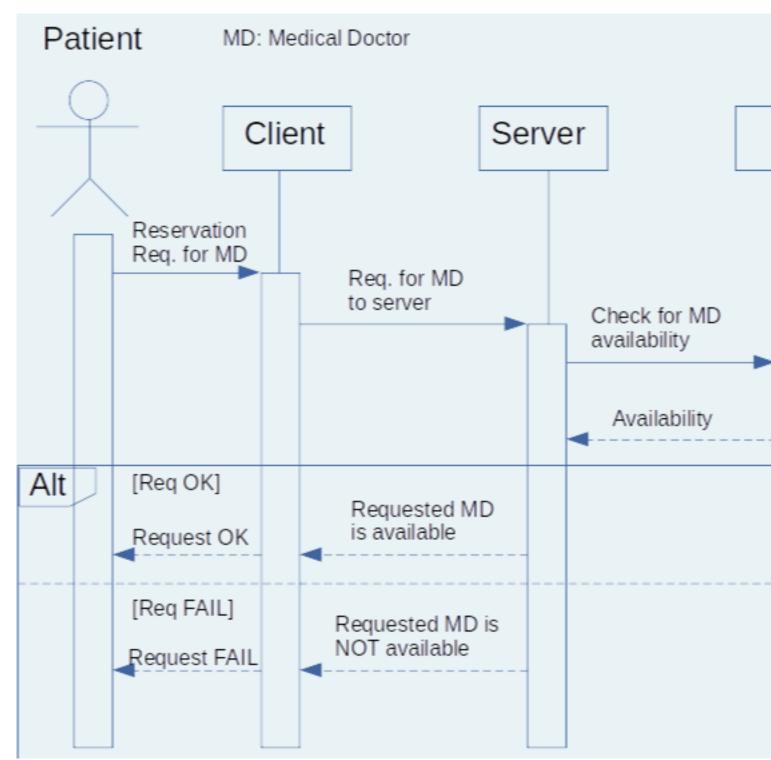
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

Test case 2: act1 act2

Test case 3: act2 act0Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?



1.70%

2.90% 3.40%

Risposta: 3

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.

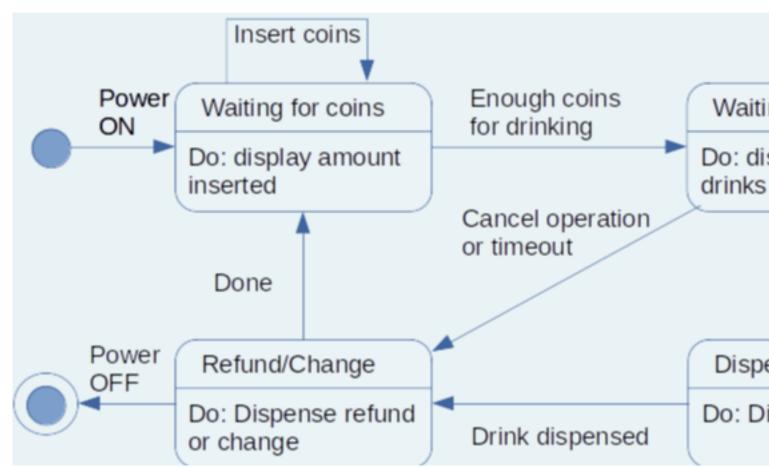
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

Test case 1: act0 act1 act2 act2 act1 act1 act0 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



1.State coverage: 60%

2.State coverage: 100%

3.State coverage: 75%

Risposta: 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.

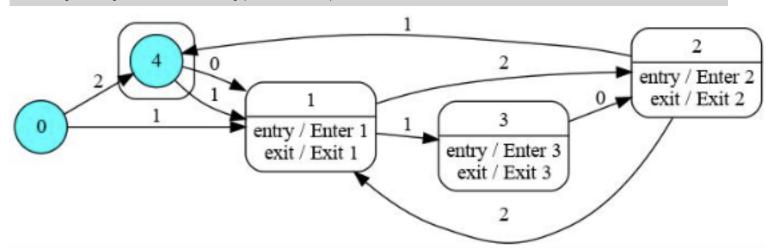
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act0 act2 act1 act2

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?



- 1.Transition coverage: 50%
- 2.Transition coverage: 100%
- 3.Transition coverage: 80%

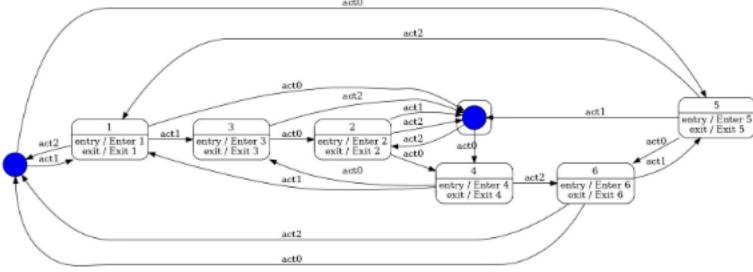
Risposta: 1

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 act1 act2 act1 act2 act1 act2 act1 act2 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



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

Risposta:

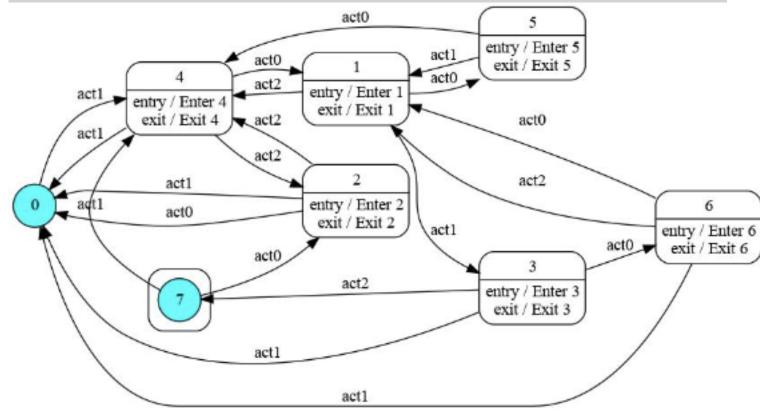
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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 act0Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?



- 1.Transition coverage: 100%
- 2.Transition coverage: 70%
- 3.Transition coverage: 60%

Risposta: 2

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.

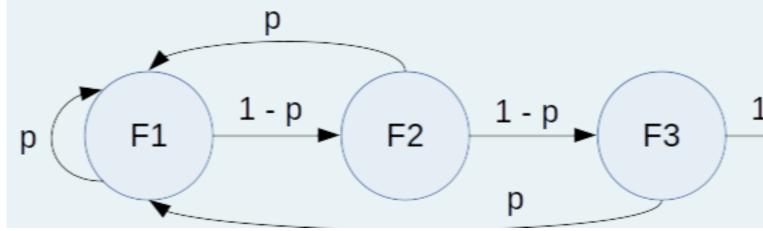
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act1 act1 act1

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

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

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



1.100%

2.60%

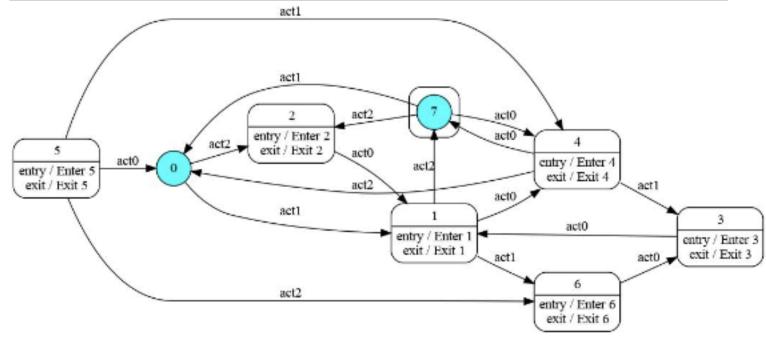
3.80%

Risposta: 3

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.

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 act0 act2 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



1.State coverage: 100%

2.State coverage: 80%

3.State coverage: 60%

Risposta: 1

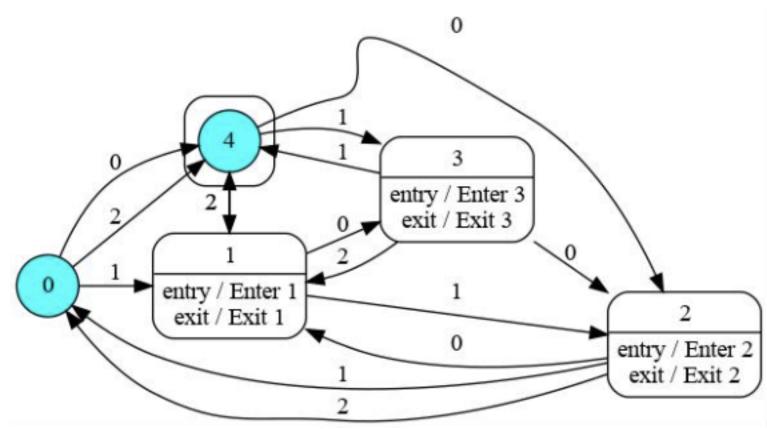
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act0 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?



1.60%

2.90%

3.45%

Risposta: 3

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.

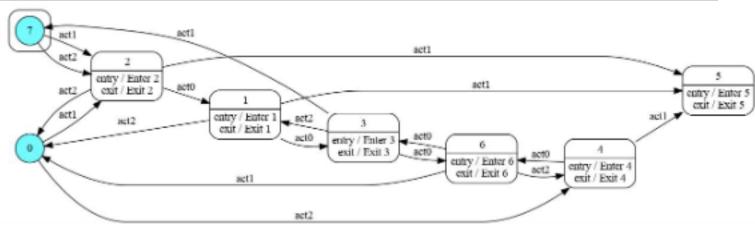
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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



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

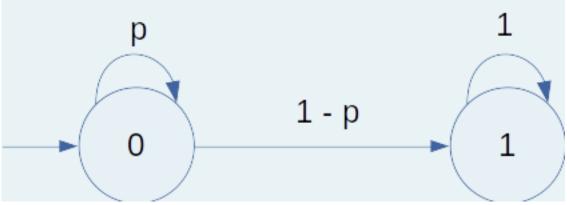
Risposta: 2

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 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?



1.50%

2.90%

3.70%

Risposta: 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) raggiunti almeno una volta.

Si consideri lo state diagram in figura

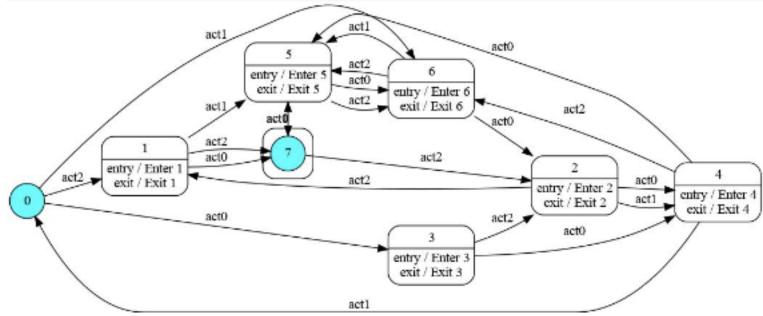
Si consideri il seguente insieme di test cases:

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



1.State coverage: 80%

2.State coverage: 100%

3.State coverage: 90%

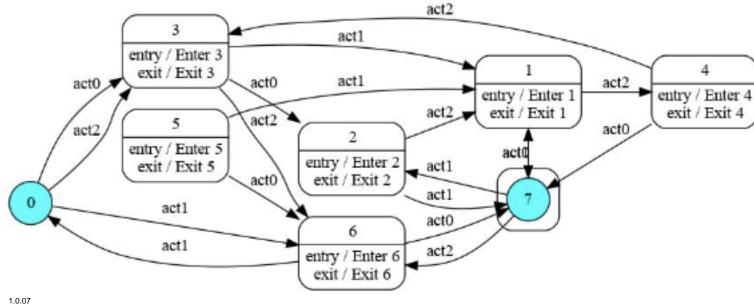
Risposta : 2

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

Uno scenario è una sequenza di stati.

Qual'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 ?



2.0.03

3.0.27

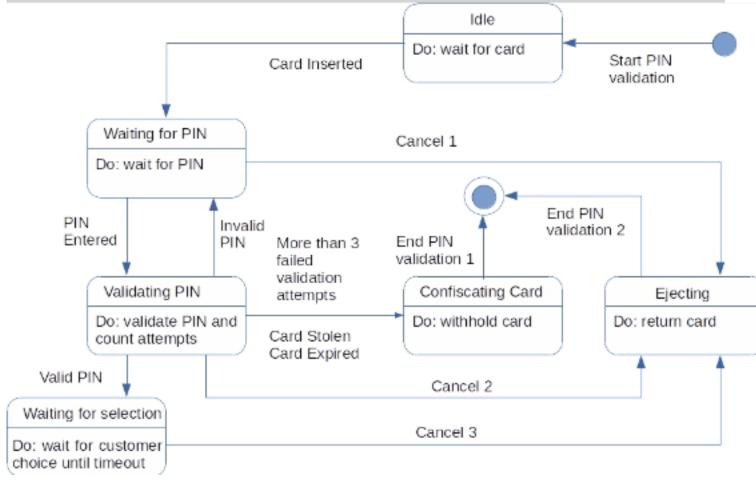
Risposta: 2

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 figuraRappresenta 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?



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.

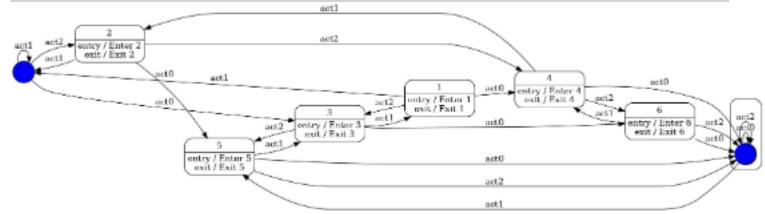
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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



1.State coverage: 90%

2.State coverage: 100%

3.State coverage: 70%

Risposta: 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.

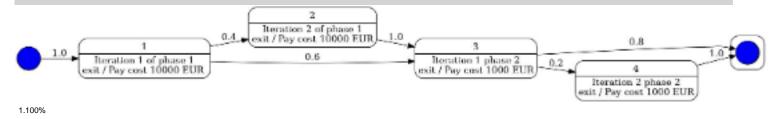
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

Test case 2: act0 act1 act2 act1 act2 act0 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?



2.60%

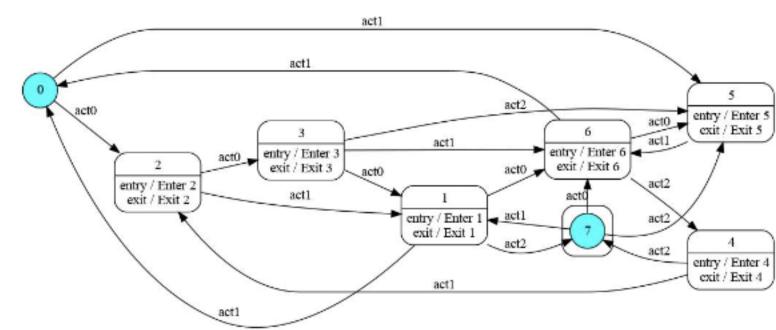
Risposta: 3

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 act1 act2 act1

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

Test case 3: act2 act0 act2 act1 act2 act1 act0 act2 act2 act0 act2 act0 act2 act1 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



1.State coverage: 90%

2.State coverage: 60%

3.State coverage: 70%

Risposta: 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.

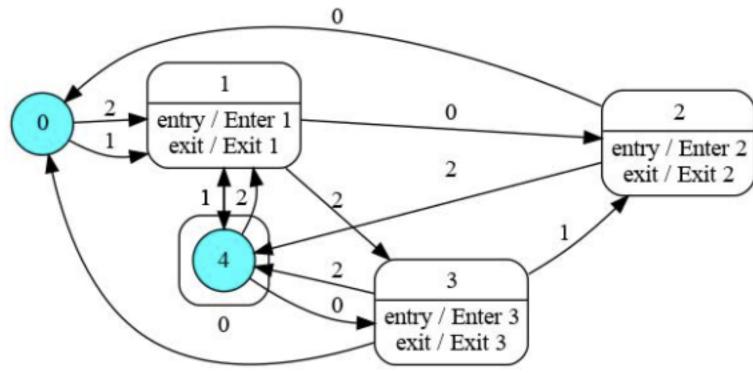
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act2 act2 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?



1.Transition coverage: 100%

2.Transition coverage: 40%

3.Transition coverage: 70%

Risposta : 2

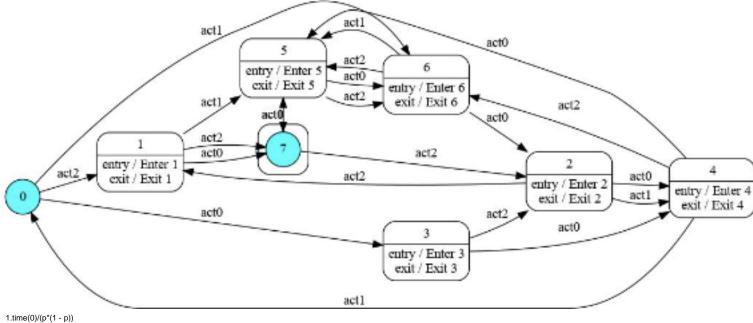
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il tempo necessario per completare la fase x è time(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta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



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

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

Risposta: 3

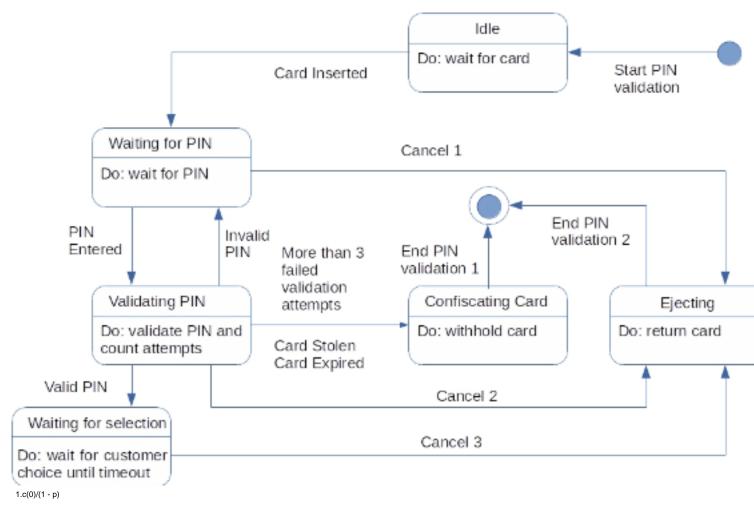
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il costo dello stato (fase) x è c(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta il completamento 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 $X = x(0), x(1), x(2), \\ e C(X) = c(x(0)) + c(x(1)) + c(x(2)) + ...$

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



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

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

Risposta: 1

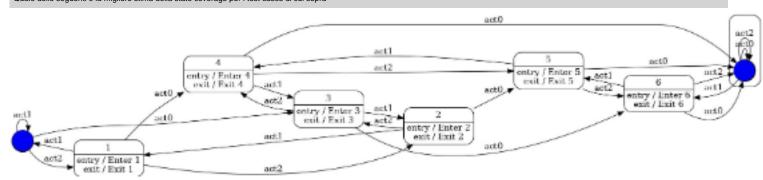
La state coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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



1.State coverage: 80%

2.State coverage: 100%

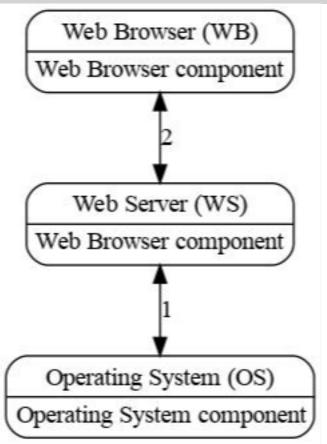
3.State coverage: 75%

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

Uno scenario è una sequenza di stati.



1.0.08

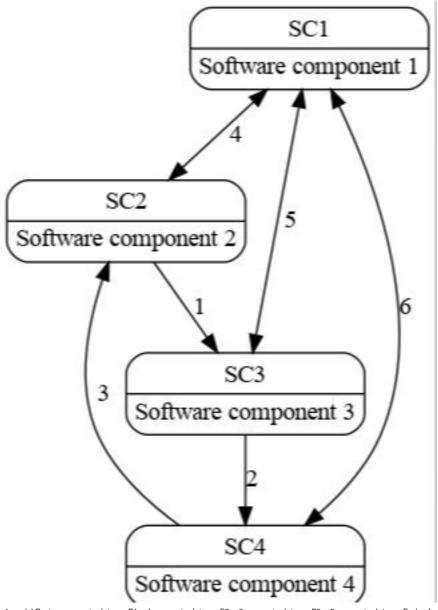
2.0.32

3.0.12

Risposta: 3

Si consideri la seguente Markov Chain:

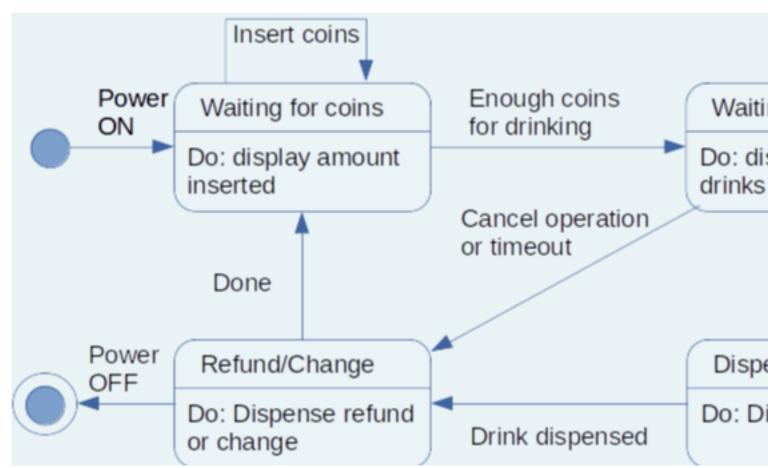
Quale dei seguenti modelli Modelica fornisce un modello ragionevole per la Markov Chain di cui sopra?



1.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 0, 1-p, 0;0, p, 1-p, 0;p, 0, 0, 0, 0] state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen initial() thenstate1024 1-p:0. 0. 0. r1024:Integer 11:Integer Real Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657. 30020):x F1:r1024 then(r1024.state1024) 0:elsewhen sample(0.1) Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A A[x, F3]) then x := F3; else x := End;end if;end when;end System; 2.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] =[p, 0, 1-p, 0;p, 1-p, 0, 0;p, 0, 0, 0, 0, 0] state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen initial() thenstate1024 1];Integer Real r1024;Integer Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020);x F1;r1024 0;elsewhen sample(0,1) then(r1024,state1024) Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2]) A[x, F3]) then x := F3; else x := End; end if; end when; end System; 3.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 1-p, 0, 0;p, 0, 1-p, 0;p, 0, 0, 0, 0, 0, 0] Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen initial() then x := F1; r1024 := 0;elsewhen sample(0,1) then(r1024,state1024) := Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020); Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] A[x, F3]) then x := F3; else x := End;end if;end when;end System;

Si consideri l'automa segunete:

Quale dei seguenti modelli Modelica fornisce un modello ragionevole per l'automa di cui sopra.



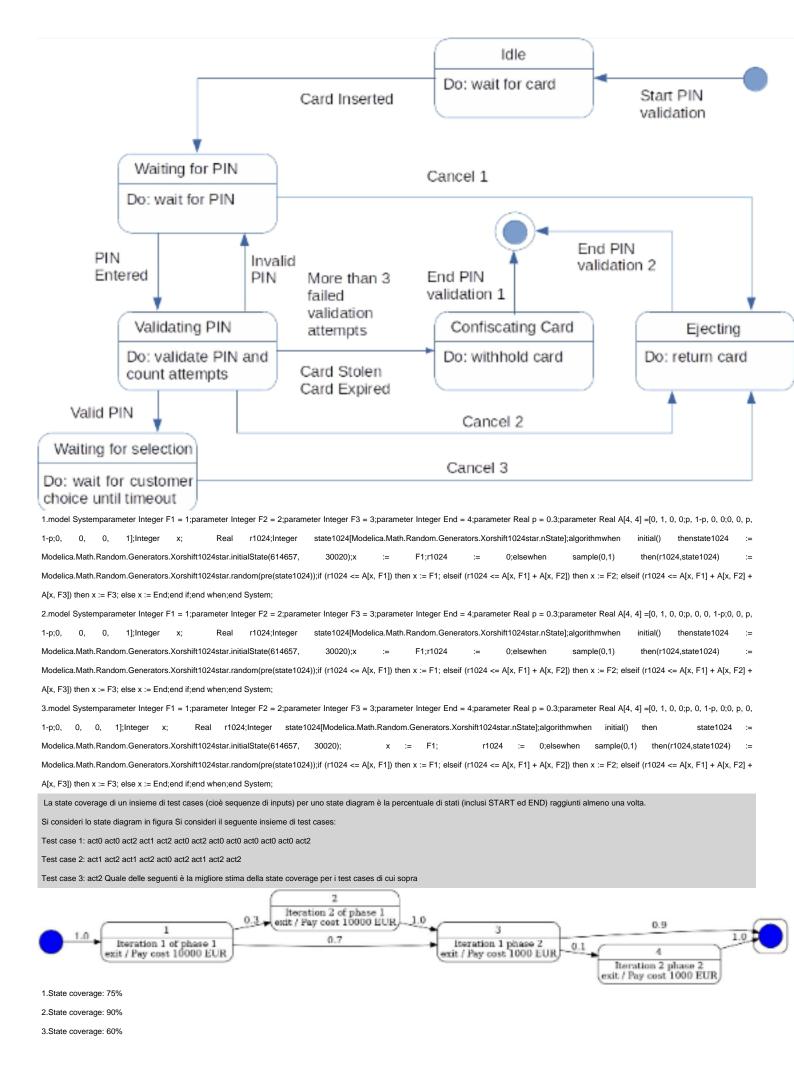
1.function nextinput Integer x;output Integer x;output Integer y;algorithm y := 1 + x;end next;class SystemInteger x;initial equationx = 0;equationwhen sample(0, 1) then x = next(pre(x));end when;end System;

2.function nextinput Integer x;output Integer y;algorithm y := x;end next;class SystemInteger x;initial equationx = 0;equationwhen sample(0, 1) then x = next(pre(x));end when;end System;

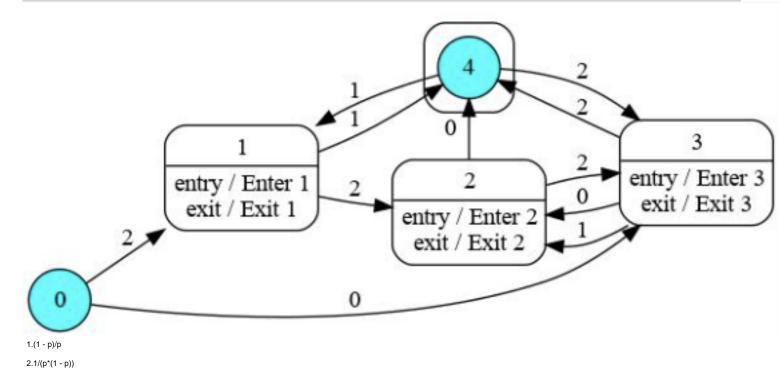
3.function nextinput Integer x;output Integer y;algorithm y := 1 - x;end next;class SystemInteger x;initial equationx = 0;equationwhen sample(0, 1) then x = next(pre(x));end when;end System;

Si consideri la seguente Markov Chain:

Quale dei seguenti modelli Modelica fornisce un modello ragionevole per la Markov Chain di cui sopra?



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.



3.1/(1 - p)

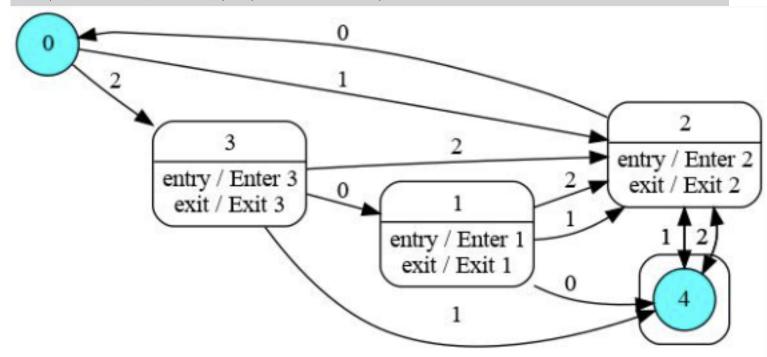
Risposta: 3

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 figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità 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.

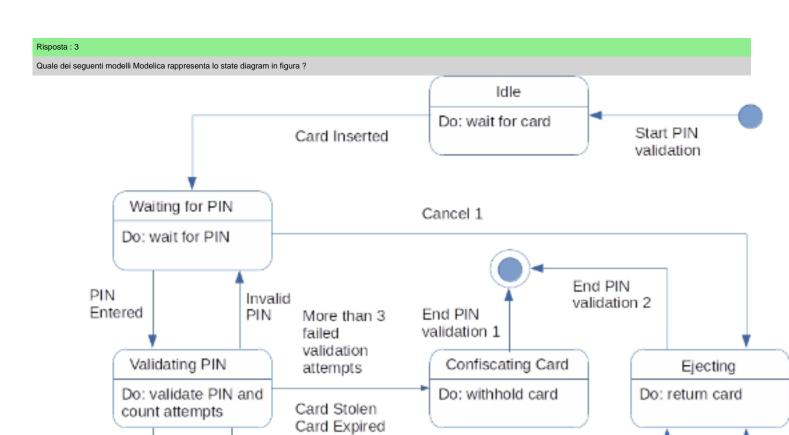
Qual'e' la probabilità dello scenario: 1, 3 ? In altri terminti, qual'è la probabilità che non sia necessario ripetere nessuna fase?



1.0.24

2.0.14

3.0.56



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) == 1) then x := 4;elseif (pre(x) == 1) 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) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;elseif (pre(x) == 4) and (pre(u) ==

Cancel 2

Cancel 3

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) == 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) == 2) 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) == 4) and (pre(u) == 0) then x := 3;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 4) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 4) and (pre(u) == 4) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) ==

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

Risposta: 2

Valid PIN

Waiting for selection

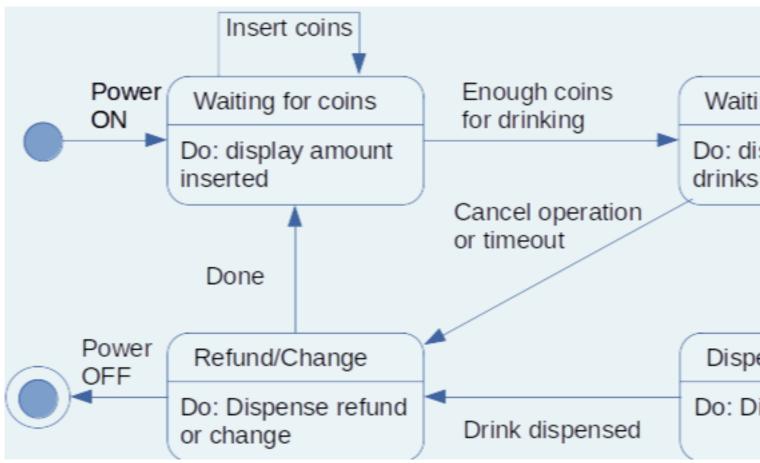
Do: wait for customer choice until timeout

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 act2

Test case 3: act2 act0 Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?



1.80%

2.60% 3.100%

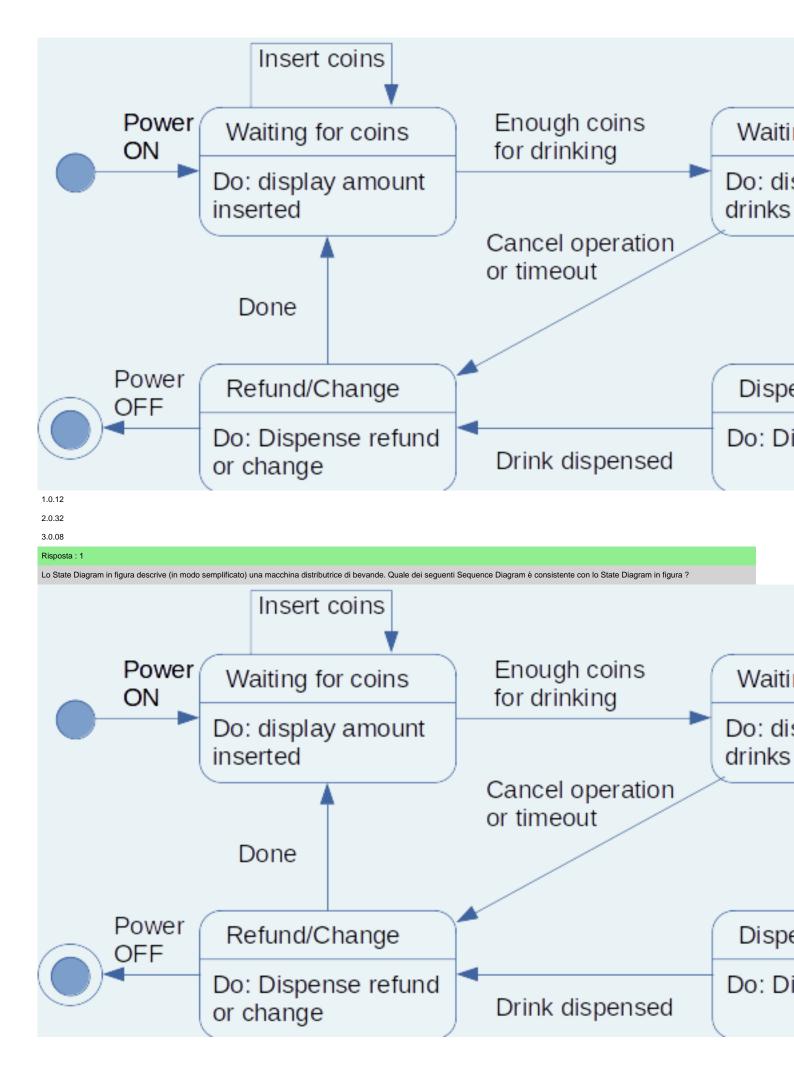
Risposta : 2

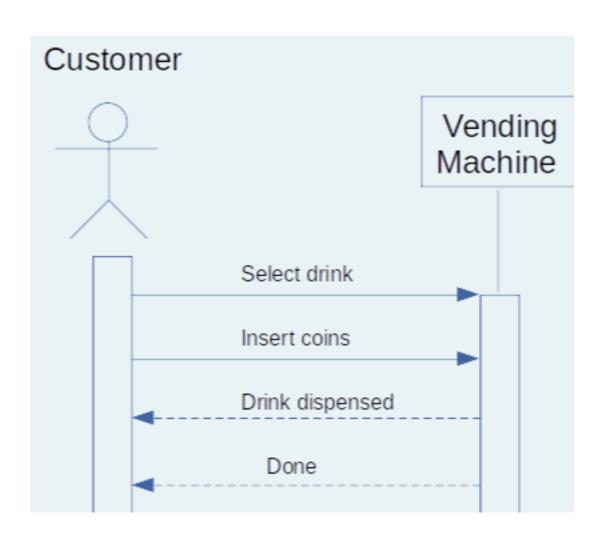
Un processo software può essere rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del processo software e gli archi le transizioni da una fase all'altra. Gli archi sono etichettati con le probabilità della transizione e gli stati sono etichettati con il costo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

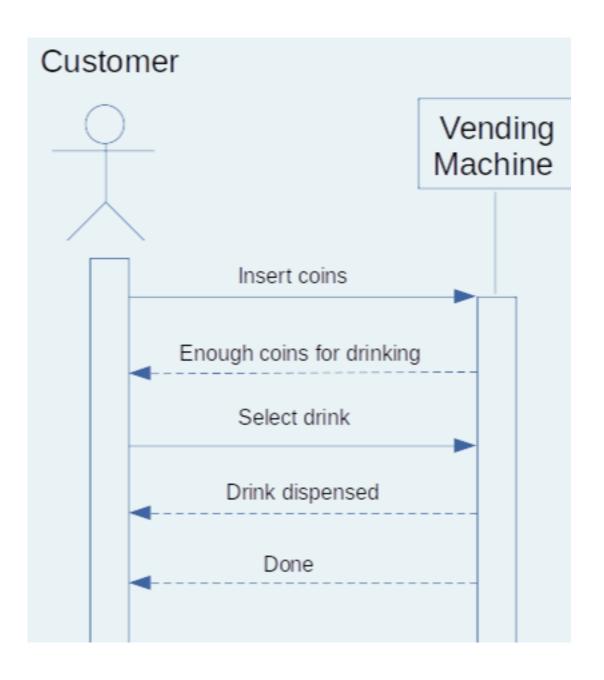
Ad esempio lo state diagram in figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità 0.2 di dover essere ripetuta (a causa di errori) ed F2 ha una probabilità 0.2 di dover essere ripetuta (a causa di errori).

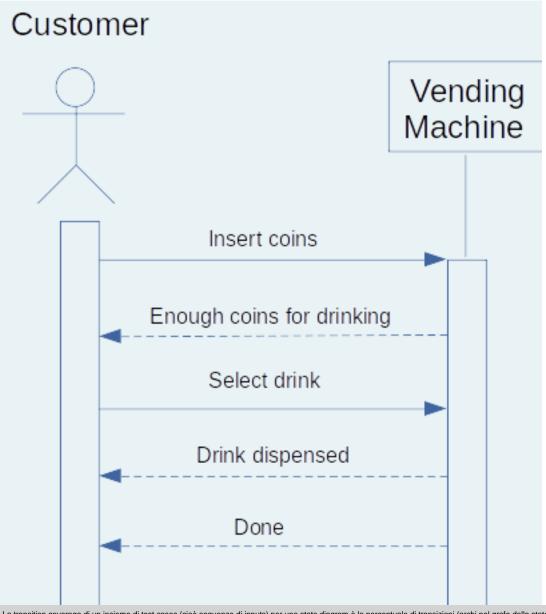
Uno scenario è una sequenza di stati.

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)?







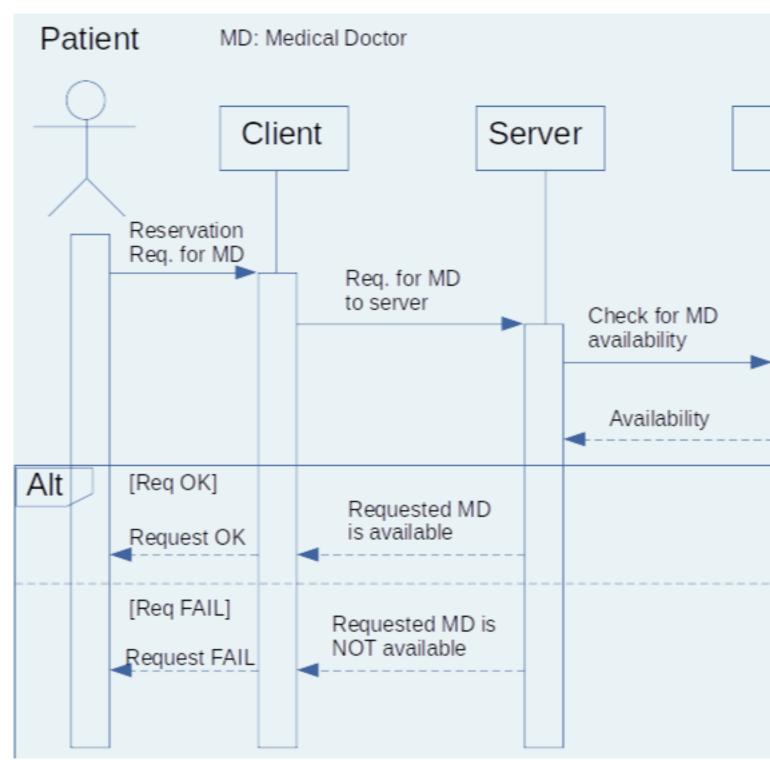


La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

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



1.100%

2.35% 3.50%

Risposta: 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.

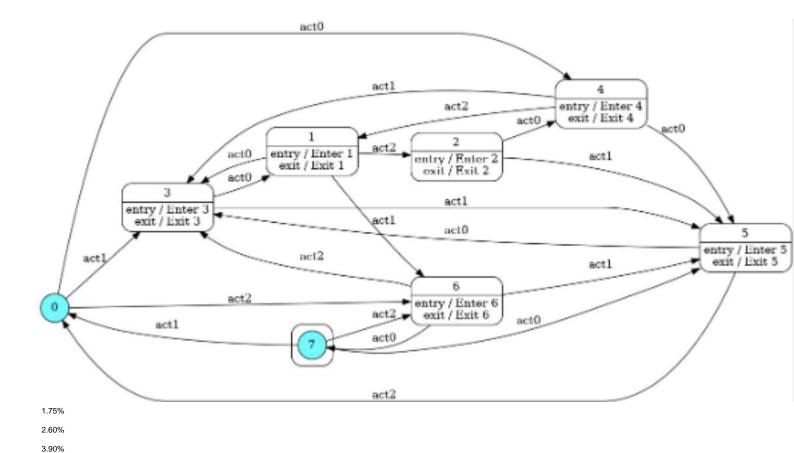
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act0

Test case 2: act1 act0 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?



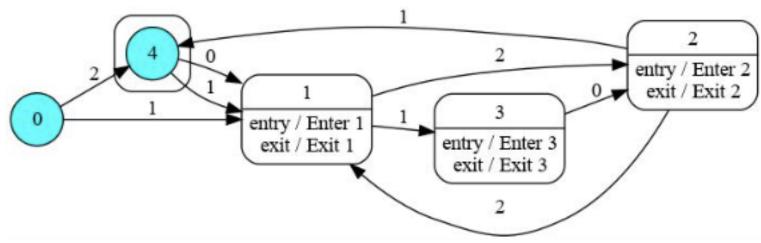
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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

Test case 3: act1 act1

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



1.State coverage: 60%

2.State coverage: 100%

3.State coverage: 85%

Risposta: 3

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 act1 act1 act2 act1 act2 act1 act2 act0 act0 act0 act0

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

- 1.State coverage: 90%
- 2.State coverage: 75%
- 3.State coverage: 60%

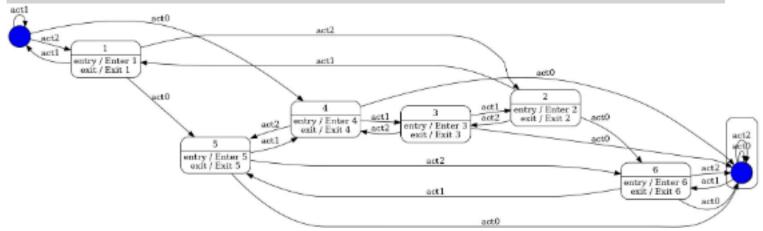
Un processo software può essere rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del processo software e gli archi le transizioni da una fase all'altra. Gli archi sono etichettati con le probabilità della transizione e gli stati sono etichettati con il costo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

Ad esempio lo state diagram in figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità dello 0.3 di dover essere ripetuta (a causa di errori) ed F2 ha una probabilità 0.1 di dover essere ripetuta (a causa di errori).

Uno scenario è una sequenza di stati.

Qual'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 ?

act2



1.0.03

2.0.07

3.0.27 Risposta : 1

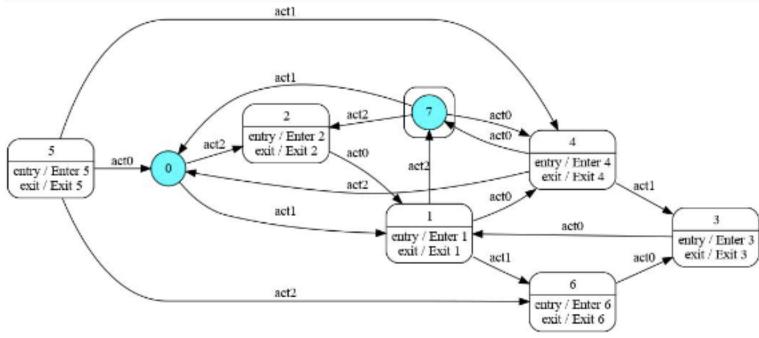
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il tempo necessario per completare la fase x è time(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta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

 $Quindi il costo Time(X) della sequenza di stati X = x(0), x(1), x(2), \ \dot{e} \ Time(X) = time(x(0)) + time(x(1)) + time(x(2)) + ...$

 $\label{eq:Adesempiose} Ad \ esempiose \ X=0, \ 1 \ abbiamo \ Time(X) = time(0) + time(1) = time(0) \ \ (poichè \ time(1) = 0).$

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



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

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

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

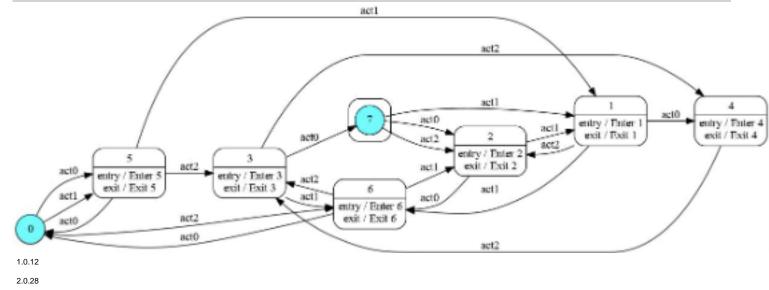
Risposta: 3

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 figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità 0.3 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.

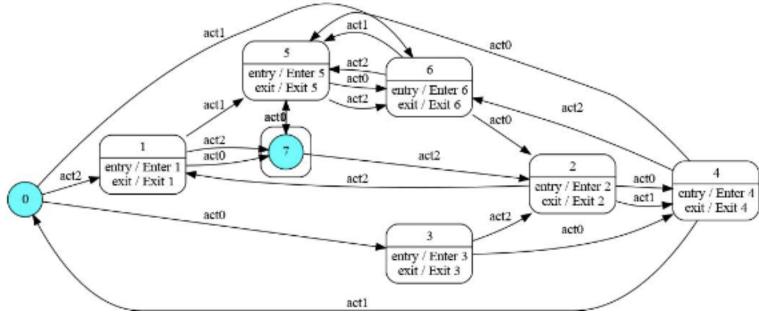
Qual'e' la probabilità dello scenario: 1, 2, 3? In altri terminti, qual'è la probabilità che non sia necessario ripetere la prima fase (ma non la seconda)?



Risposta : 2

3.0.42

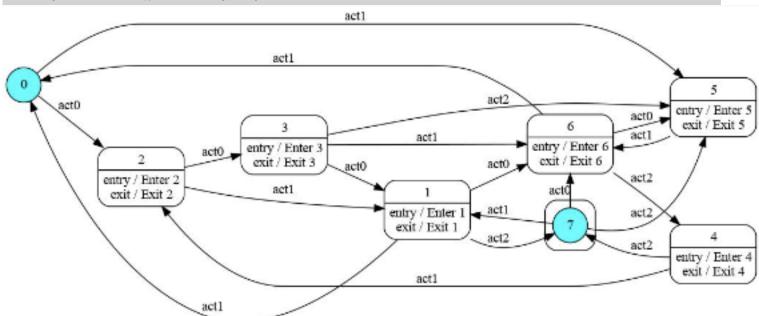
Quale delle seguenti frasi è corretta riguardo all'ctivity diagram in figura?



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

Risposta: 2

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

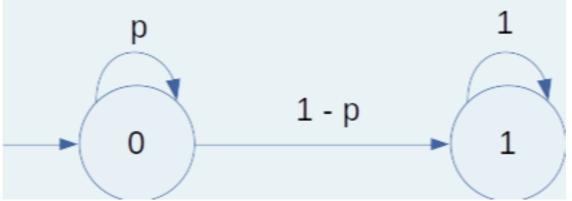


1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif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

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) == 2) then x := 3;elseif (pre(x) == 2) and (pre(u) == 2) 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 := 1;elseif (pre(x) == 3) and (pre(u) == 0) then x := 0;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) == 2) then x := 0;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() 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 := 3; elseif (pre(x) == 2) 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;

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



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 := 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) == 1) and (pre(u) == 1) and ((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 := 1; elseif (pre(x) = 3) and (pre(u) = 2) then x := 1; elseif (pre(x) = 3) and (pre(u) = 2) then x := 1; elseif (pre(x) = 3) and (pre(u) = 2) then x := 1; elseif (pre(x) = 3) and (pre(u) = 2) then x := 1; elseif (pre(x) = 3) and (pre(u) = 2) then x := 1; elseif (pre(x) = 3) and (pre(u) = 2) then x := 1; elseif (pre(x) = 3) and (pre(u) = 3) and (p==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((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) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 3) then x := 4; elseif (pre(x) == 3) and (pre(u) == 3) then x := 4; elseif (pre(x) == 4) then 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 := 4; elseif (pre(x) == 1) and (pre(u) == 2) then x := 4; elseif (pre(x) == 1) and (pre(u) == 2) then x := 4; elseif (pre(x) == 1) and (pre(u) == 2) then x := 4; elseif (pre(x) == 1) and (pre(u) == 2) then x := 4; elseif (pre(x) == 1) and (pre(u) == 2) then x := 4; elseif (pre(x) == 1) and (pre(u) == 2) then x := 4; elseif (pre(x) == 1) and (pre(u) == 2) then x := 4; elseif (pre(x) == 1) and (pre(u) == 2) then x := 4; elseif (pre(x) == 1) 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) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) (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 := 6; elseif (pre(x) = 4) and (pre(u) = 6) then (pre(u) = 1) then (== 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

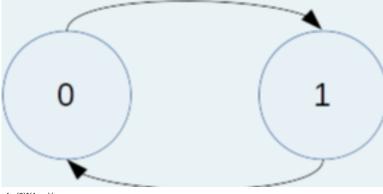
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il costo dello stato (fase) x è c(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta il completamento 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 $X = x(0), x(1), x(2), \\ e C(X) = c(x(0)) + c(x(1)) + c(x(2)) + ...$

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



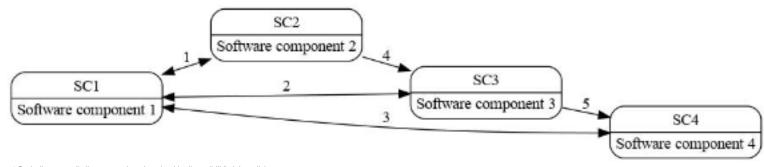
1.c(0)*(1 - p)/p

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

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

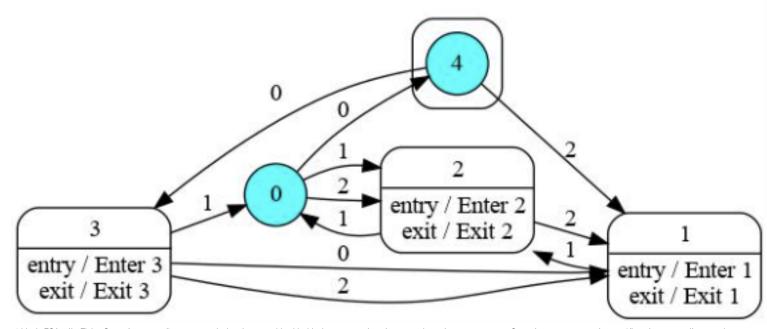
Risposta: 2

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



- 1.Periodicamente il client comunica ai pazienti le disponibilità dei medici.
- 2.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.
- 3.Il paziente richiede al server una visita con uno specífico medico e, dopo una verifica sul database, riceve conferma dal server della disponibilità o meno del medico richiesto.

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



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() 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) == 1) and (pre(u) == 2) then x := 2; elseif (pre(x) == 1) and (pre(u) == 2) then x := 2; elseif (pre(x) == 1) and (pre(u) == 2) then x := 2; elseif (pre(x) == 2) and (pre(u) == 2) then x := 3; elseif (pre(x) == 2) and (pre(u) == 1) then x := 1; elseif (pre(x) == 3) and (pre(u) == 0) then x := 1; elseif (pre(x) == 4) an

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) == 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) == 0) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 3) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 4) and (pre(

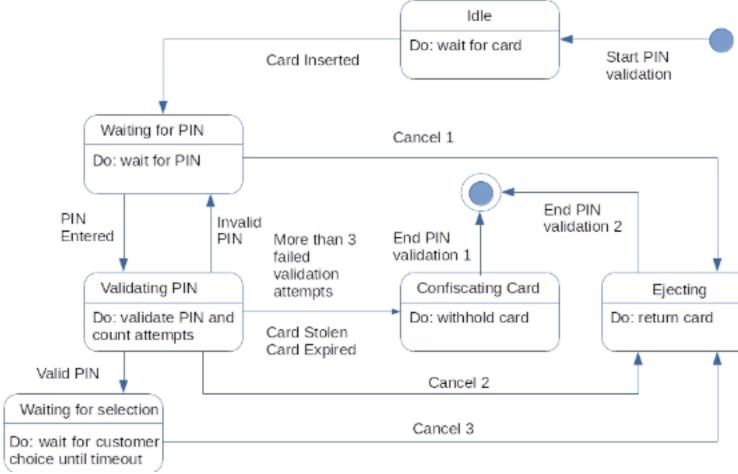
3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer; connector OutputInteger = output Integer; InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1; algorithmwhen initial() then x := 0; elsewhen sample(0,T)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

Risposta: 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 2: act0



1.90%

2.100%

3.50%

Risposta:

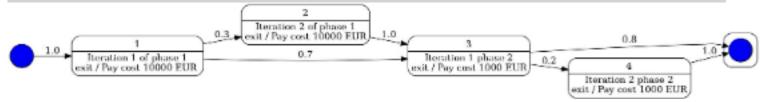
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

Test case 2: act1 act2 act0 act2 act1 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



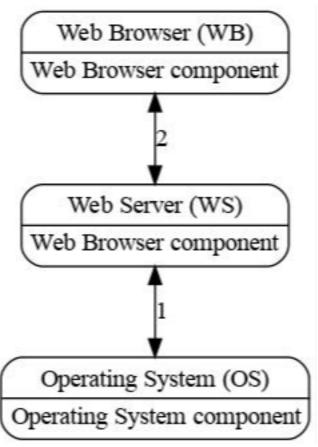
1.State coverage: 60%

2.State coverage: 100%

3.State coverage: 75%

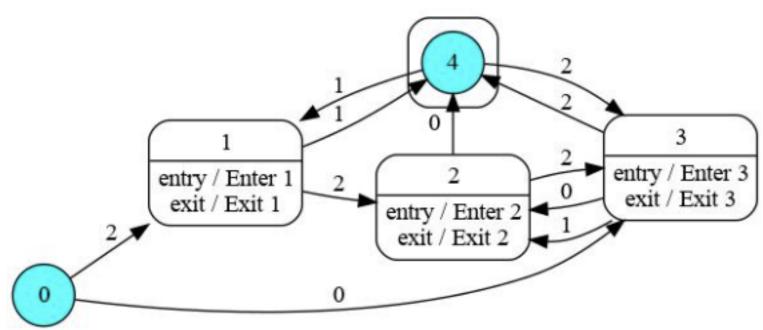
Risposta: 3

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 ?



- 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.

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

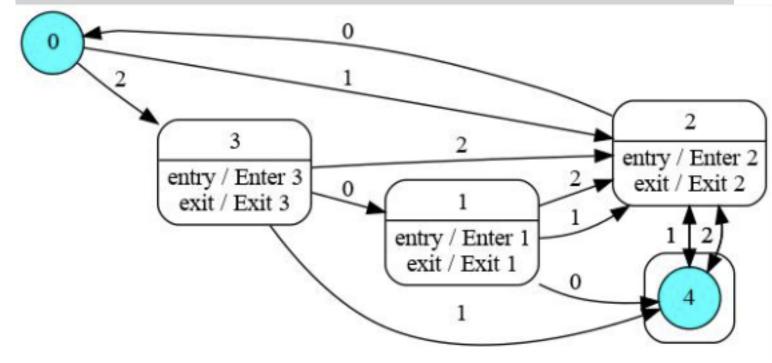


- 1.Una volta selezionato il piatto di mare da preparare, la stessa persona prepara prima il pesce e poi il contorno.
- 2. Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in sequenza.
- 3.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in parallelo.

Risposta: 3

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



1.80%

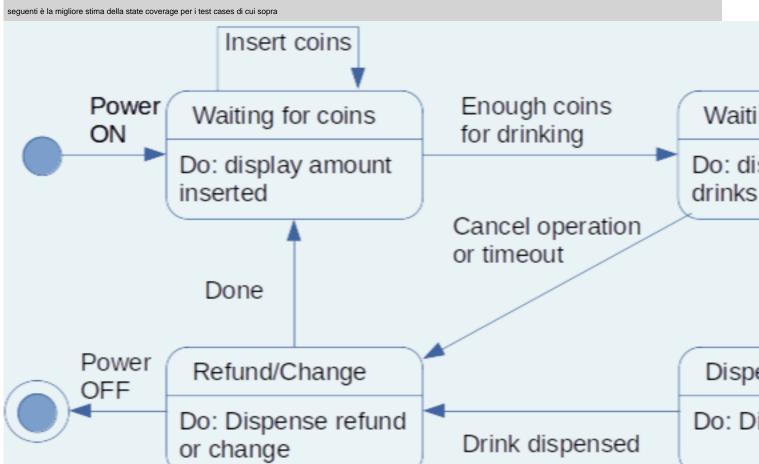
2.90%

3.60%

Risposta: 2

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.

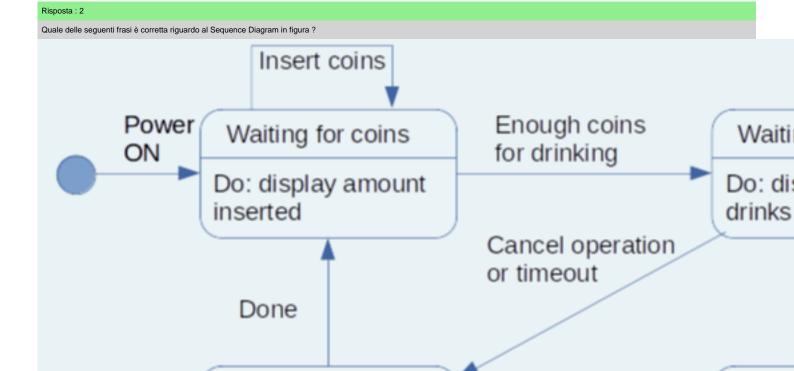
- Si consideri lo state diagram in figura ed il seguente insieme di test cases:1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2
- 2) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 2, End PIN Validation 2
- 3) Start PIN validation, card 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



```
3.100%
```

```
Risposta: 3
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 \le 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]) \mathrel{<=} 2) ? 0 : (abs(x[i]) \mathrel{-} 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?
                                            Insert coins
                                                                                           Enough coins
                Power
                                     Waiting for coins
                                                                                                                                             Waiti
                                                                                           for drinking
                ON
                                  Do: display amount
                                                                                                                                          Do: di:
                                  inserted
                                                                                                                                          drinks
                                                                                          Cancel operation
                                                                                          or timeout
                                       Done
              Power
                                     Refund/Change
                                                                                                                                             Dispe
               OFF
                                  Do: Dispense refund
                                                                                                                                          Do: Di
                                                                                            Drink dispensed
                                  or change
```

3.80%



Dispe

Do: Di

Drink dispensed

1.II 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

Refund/Change

or change

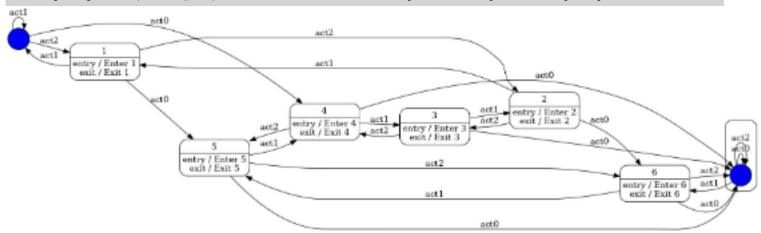
Do: Dispense refund

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

Power

Risposta: 1

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 ?



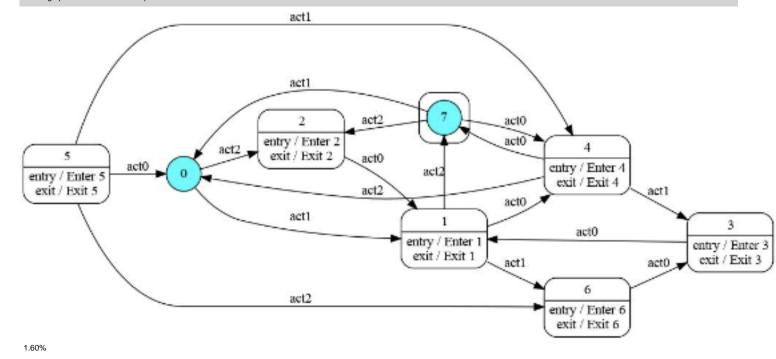
- 1.La macchina non dà resto.
- 2.Una volta inserite monete per due bevande è possibile ottenerle senza reinserire le monete.
- 3.Una volta selezionata la bevanda non è possibile cancellare l'operazione.

Risposta: 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, 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



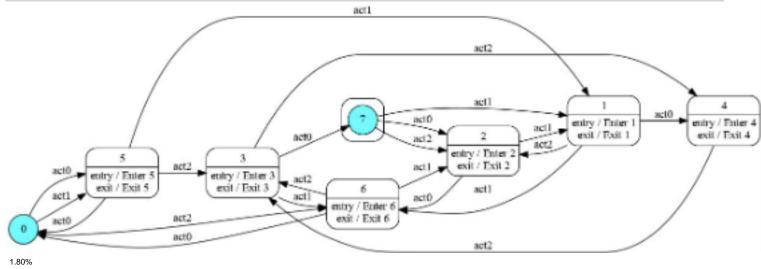
3.80% Risposta : 1

2.40%

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 2Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra

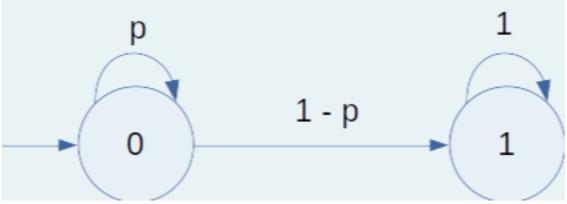


2.40% 3.60% Risposta : 3

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, 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



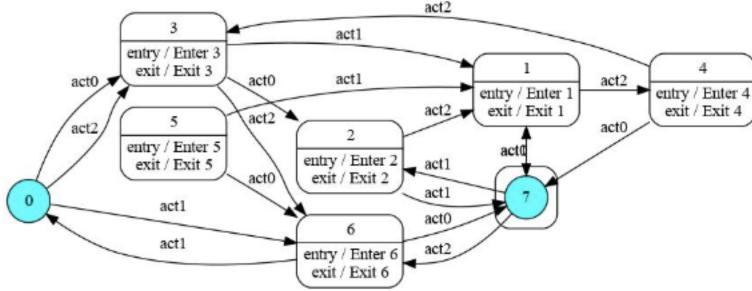
1.90%

2.100%

3.80%

Risposta: 1

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 ?



- 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

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?

Web Browser

Forms and Query Manager

Database

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 := 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:

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 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 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:

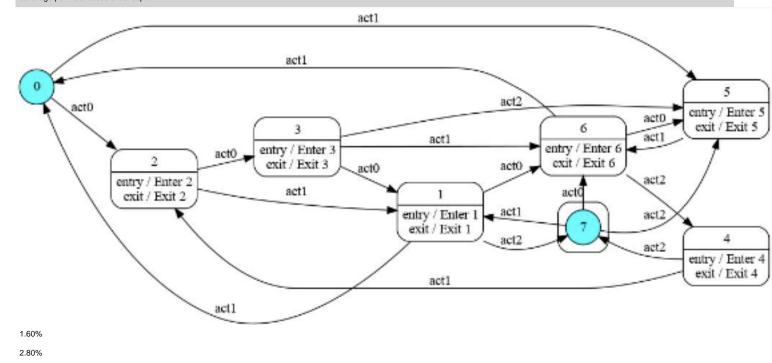
3.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0:Machine2Customer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

Risposta: 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:

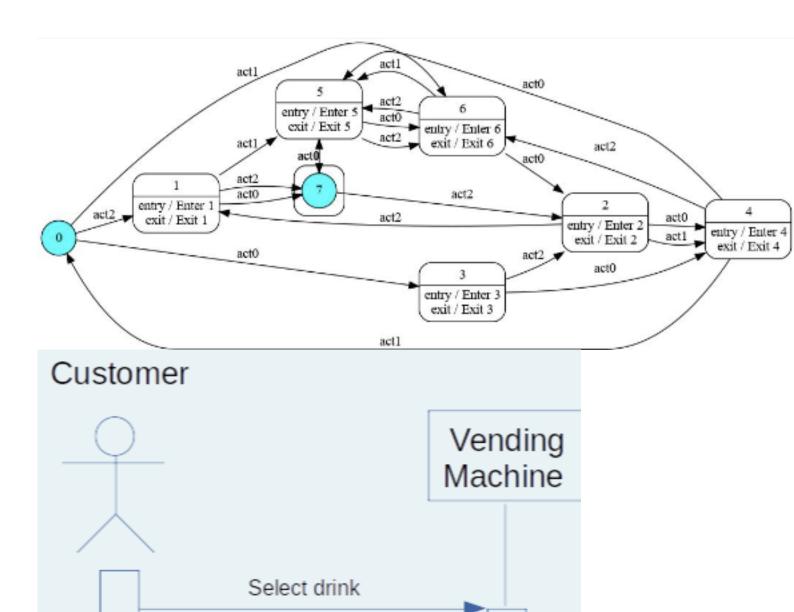
- 1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2;
- 2) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2;
- 3) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered,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



Risposta : 3

3.90%

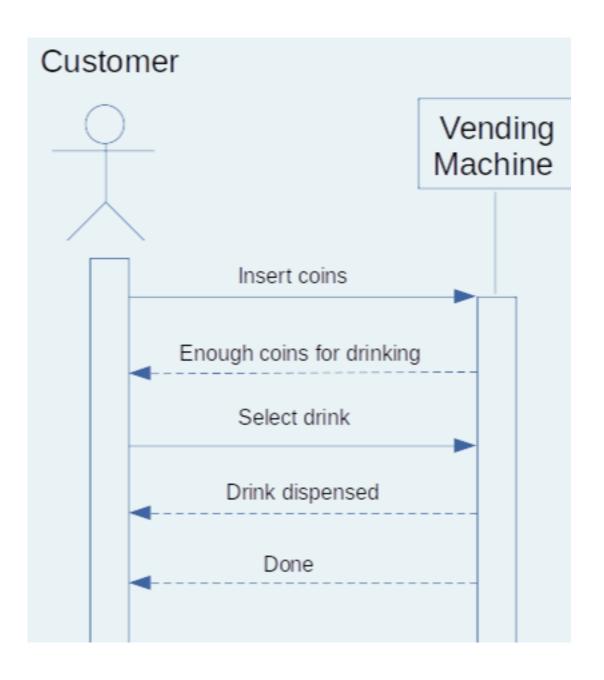
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura?

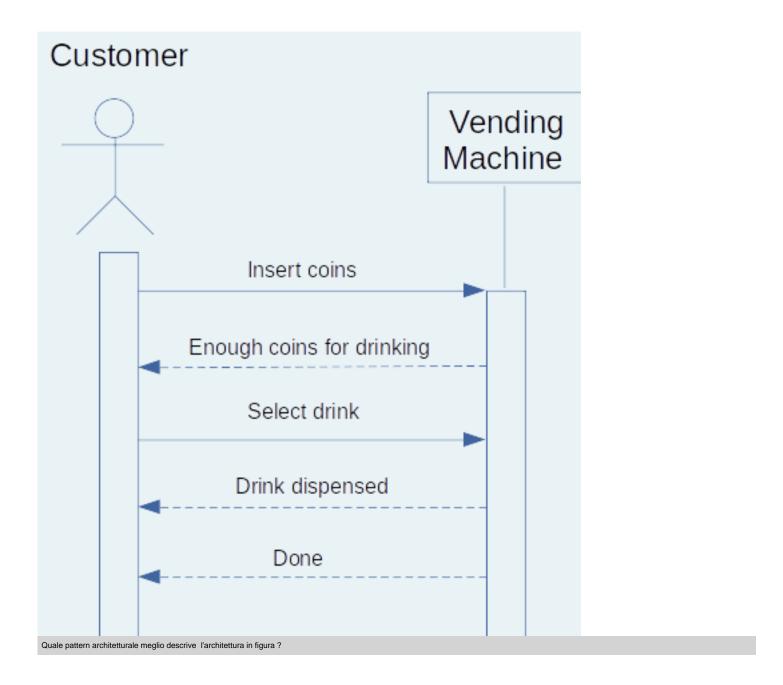


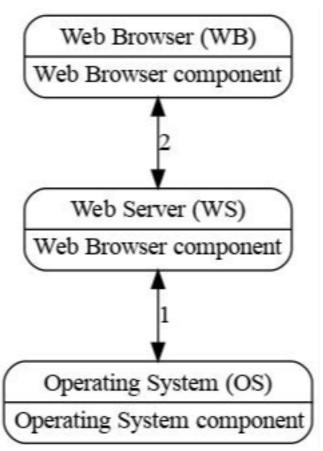
Insert coins

Done

Drink dispensed



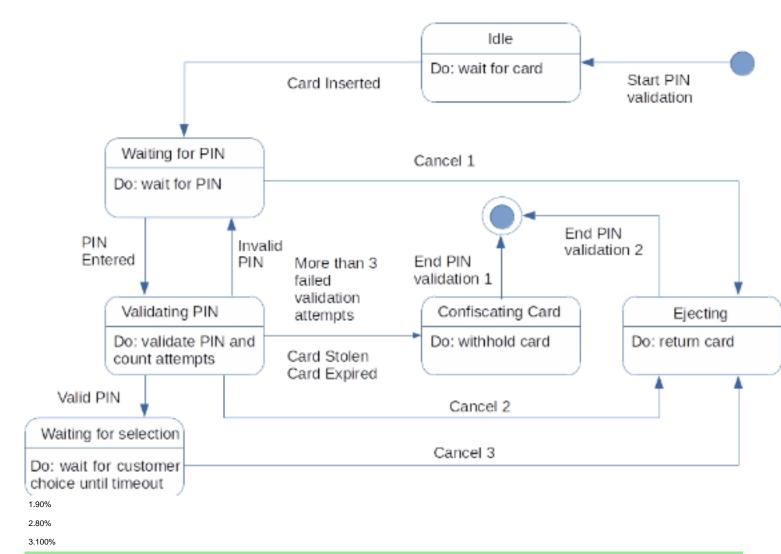




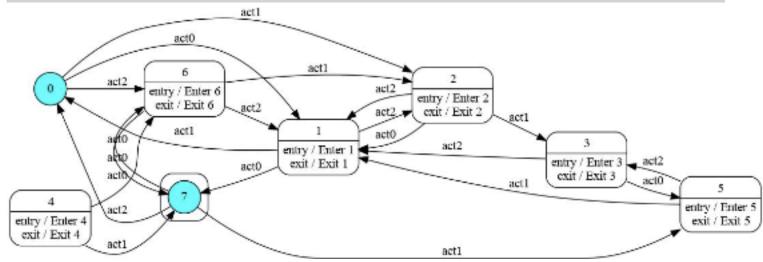
- 1.Layred architecture.
- 2.Model View Controller.
- 3.Pipe and filter architecture.

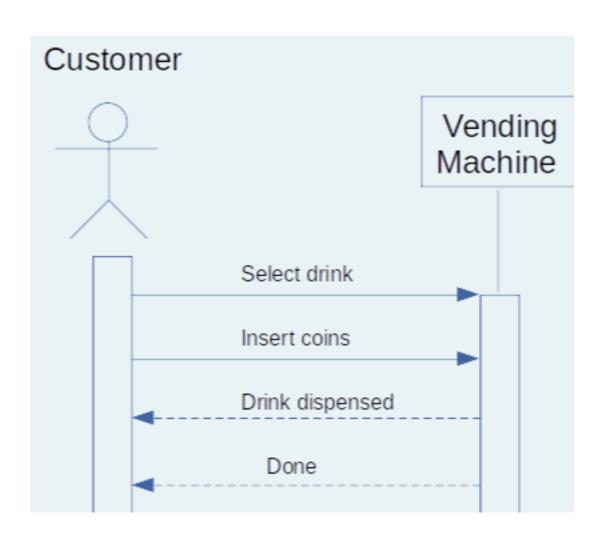
La state coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) rggiunti almeno una volta.

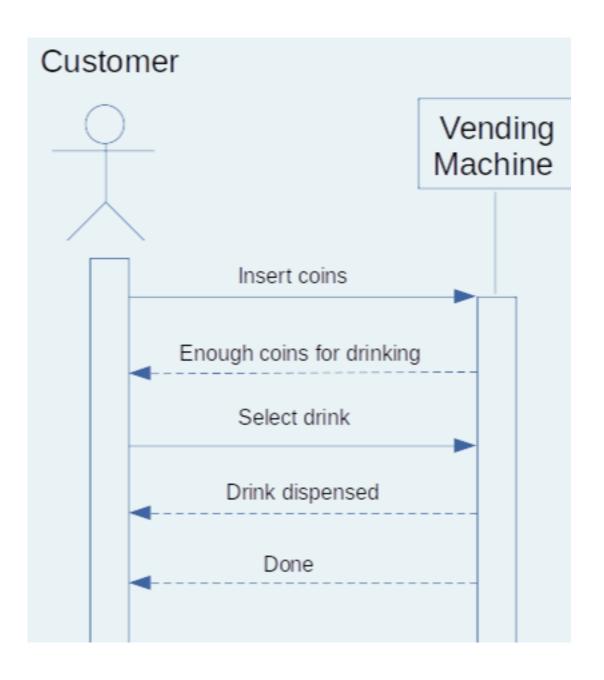
- Si consideri lo state diagram in figura ed il seguente insieme di test cases:1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2
- 2) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 2, End PIN Validation 2
- 3) Start PIN validation, card 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

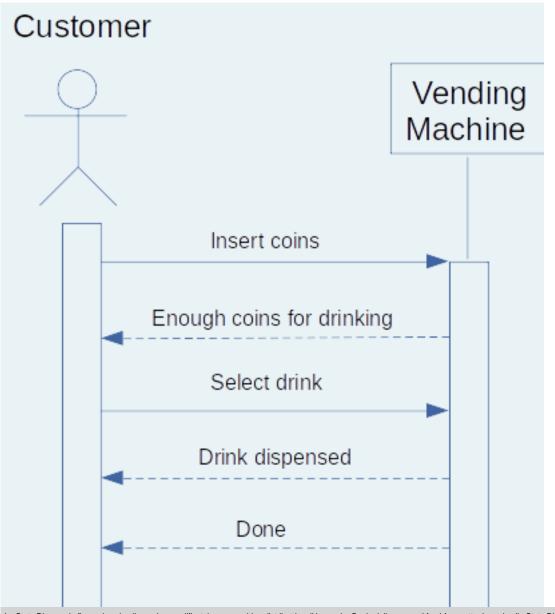


Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?

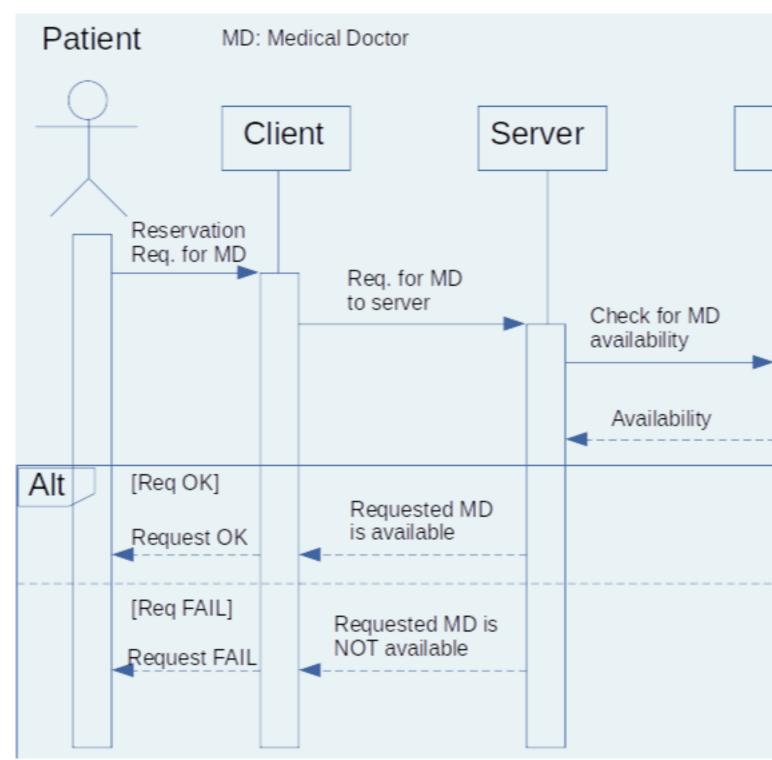






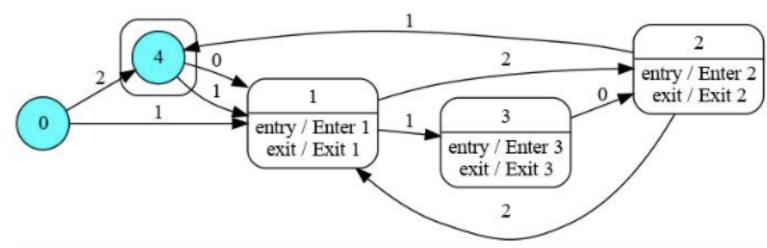


Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura?



- 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.

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?



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 3; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 0; Machine2Customer := 2;elseif (pre(state) == 3) // Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); refund/change then // refund state := 0; end if;end when;end CoffeeMachine;

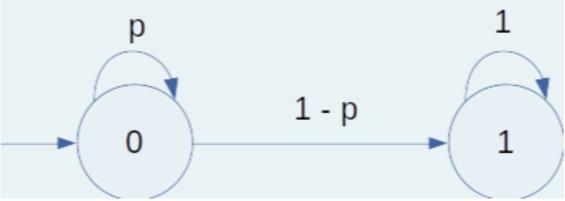
2.block CoffeeMachineparameter Real T = 1: // clockInputInteger Customer2Machine;OutputInteger Machine2Customer/*0: nop1; enough coins inserted2; drink dispensed3; done*/Integer state:/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer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

Risposta: 2

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



3.80%

Risposta: 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, 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

Web Browser

Forms and Query Manager

Database

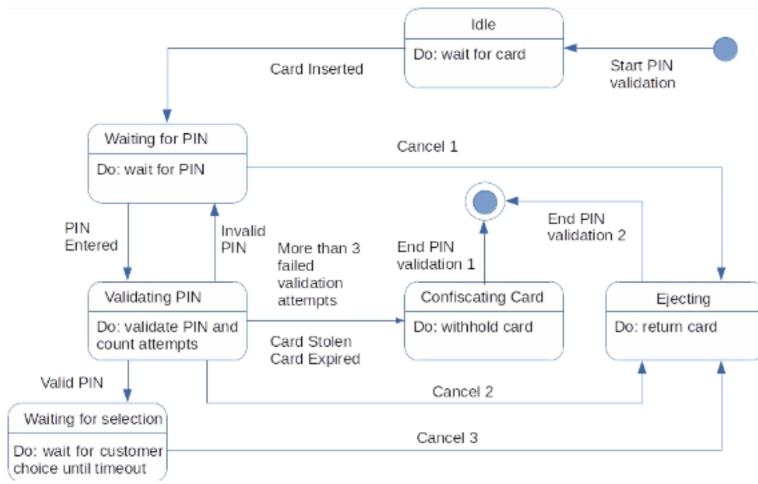
1.60%

2.90%

3.80%

Risposta: 2

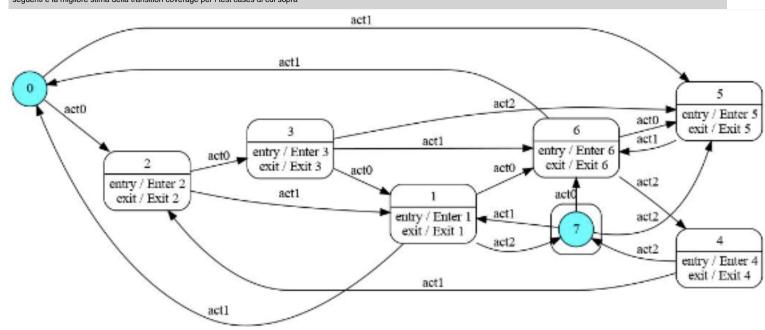
Quale delle seguenti frasi è corretta riguardo al Sequence Diagram in figura ?



- 1.II 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.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.
- 3.Periodicamente il client comunica ai pazienti le disponibilità dei medici.

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

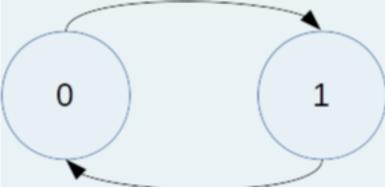


3.90%

Risposta: 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, Cancel 2, End PIN Validation 2Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra



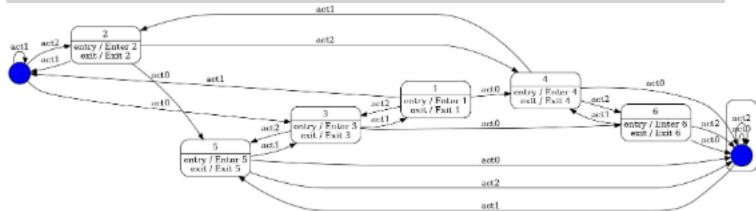
1.60%

2.80%

3.40%

Risposta: 1

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

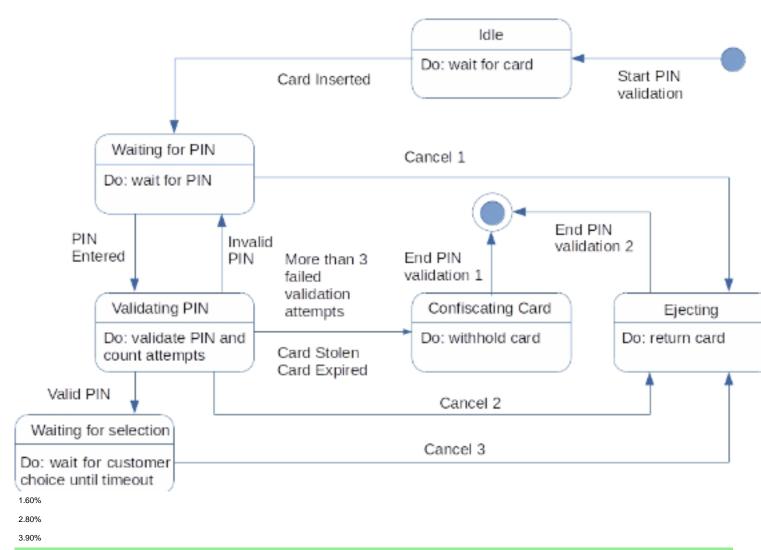


- 1. Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in parallelo.
- 2.Una volta selezionato il piatto di mare da preparare, la stessa persona prepara prima il pesce e poi il contorno.
- 3.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in sequenza.

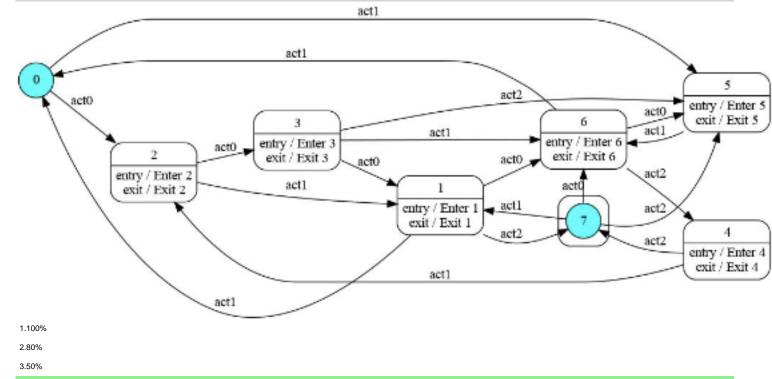
Risposta: 1

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.

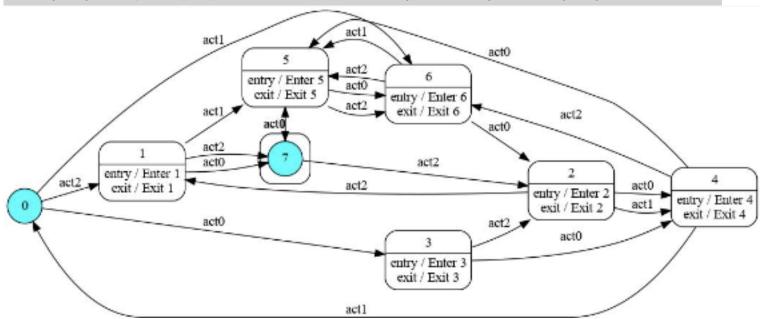
- Si consideri lo state diagram in figura ed il seguente insieme di test cases:
- 1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2
- 2) Start PIN validation, card inserted, PIN Entered, Cancel 2, End PIN Validation 2Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



```
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 \le 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].
```



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 ?



- 1.Una volta inserite monete per due bevande è possibile ottenerle senza reinserire le monete.
- 2.Una volta selezionata la bevanda non è possibile cancellare l'operazione.
- 3.La macchina non dà resto.

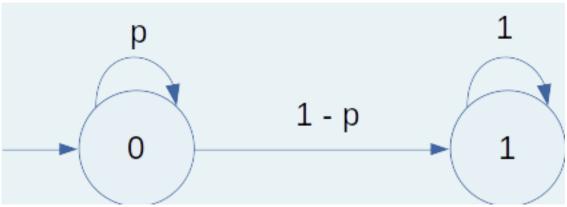
Risposta: 2

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

Uno scenario è una sequenza di stati.

Qual'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)?



1.0.32

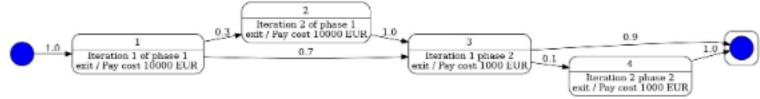
2.0.12

3.0.08

Risposta: 2

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.

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



1.State coverage: 60%

2.State coverage: 80%

3.State coverage: 100%

Risposta: 3

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.

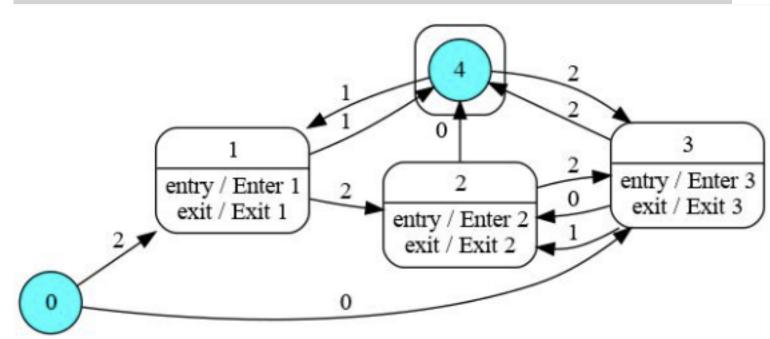
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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



1.State coverage: 80%

3.State coverage: 75%

Risposta: 3

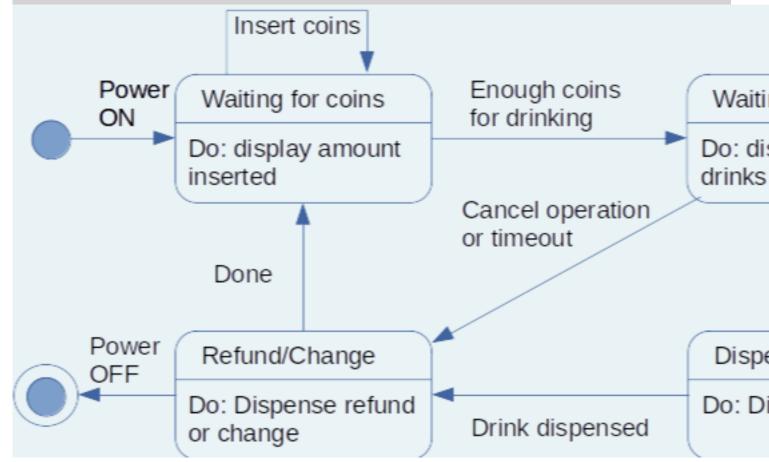
La state coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

Test case 1: act2 act1 act2 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



1.State coverage: 90%

2.State coverage: 70%

3.State coverage: 100%

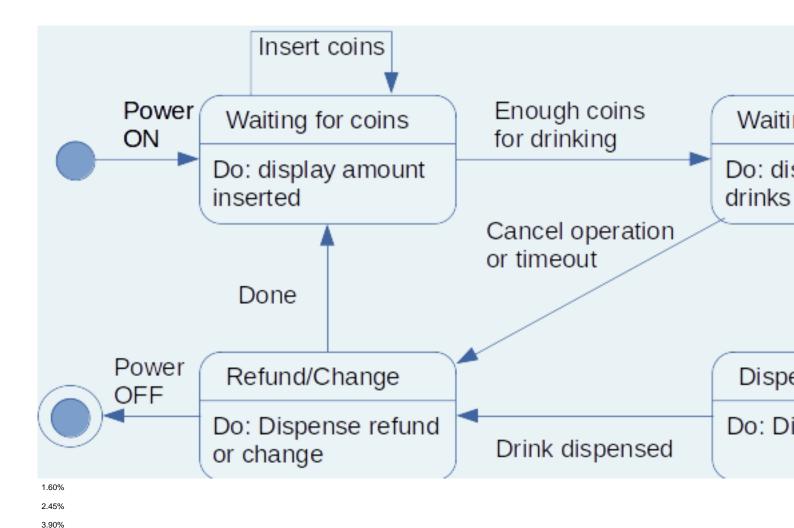
Risposta: 1

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 2: act2

Test case 3: act2 Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?



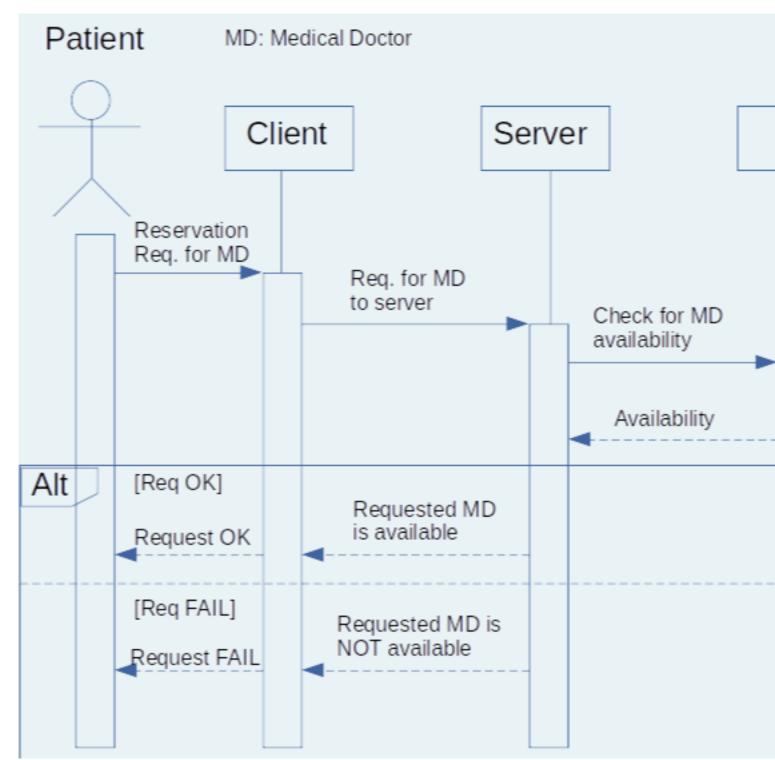
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il tempo necessario per completare la fase x è time(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta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

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

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



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

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

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

Risposta: 1

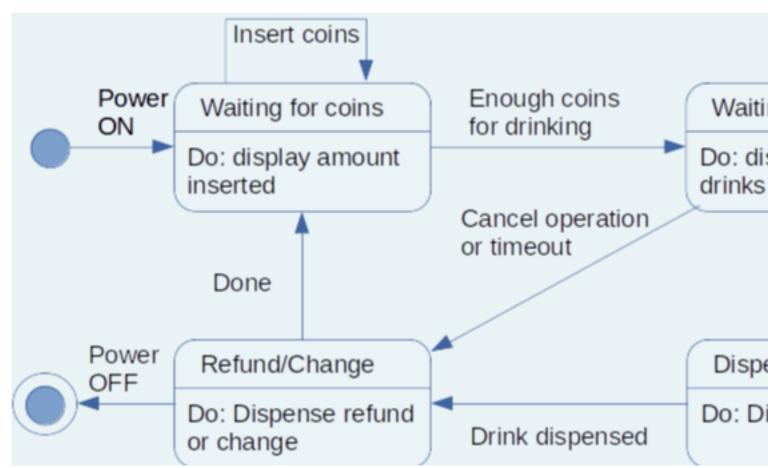
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act2 act0 act2 act2 act2

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

Test case 3: act2 act2 act2 act0 act1 act0Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?



- 1.Transition coverage: 70%
- 2.Transition coverage: 60%
- 3.Transition coverage: 100%

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

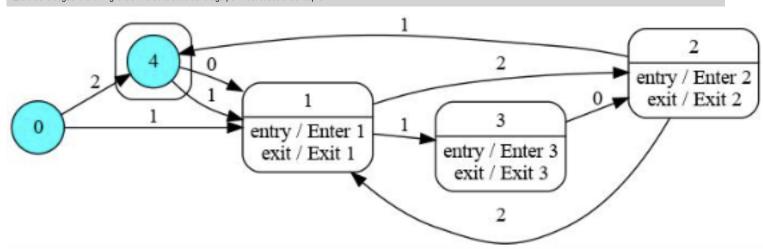
Si consideri il seguente insieme di test cases:

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



- 1.State coverage: 90%
- 2.State coverage: 100%
- 3.State coverage: 80%

Risposta: 2

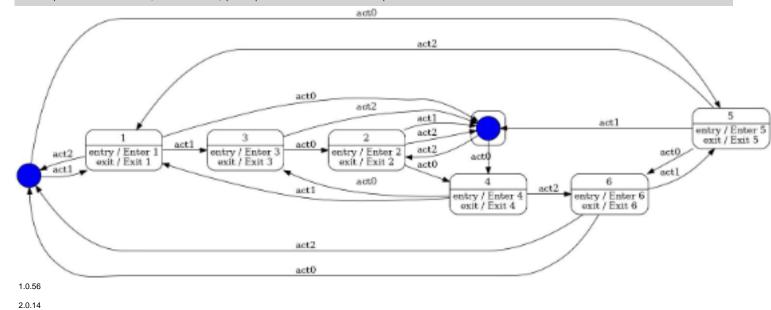
Un processo software può essere rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del processo software e gli archi le transizioni da una fase all'altra. Gli archi

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

Uno scenario è una sequenza di stati.

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



Risposta: 1

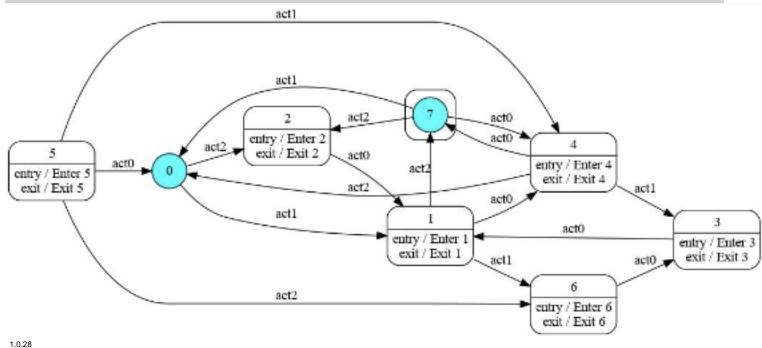
3.0.24

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 figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità 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)?



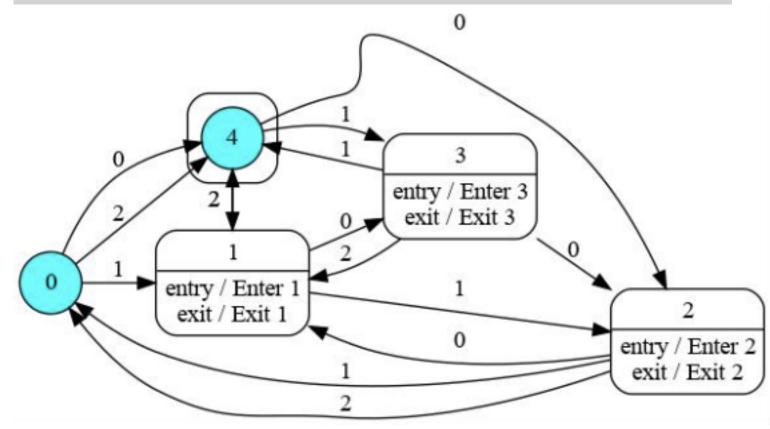
2.0.12 3.0.42

Risposta : 1

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 act2 act1 act2 act1 act2 act1 act2 act1

Test case 3: act2 act0 act2 act1 act2 act1 act0 act2 act1 act0 act2 act0 act0 act0 act0 act1 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



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

Risposta: 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.

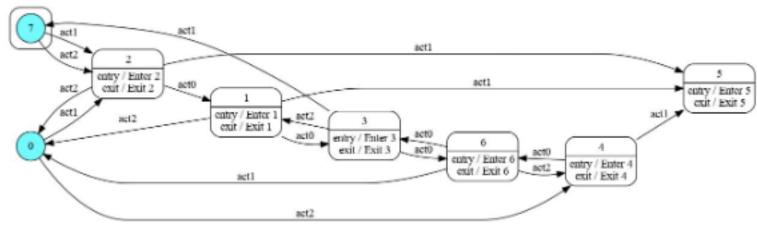
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act0 act2 act1 act2

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?



- 1.Transition coverage: 80%
- 2.Transition coverage: 100%
- 3.Transition coverage: 50%

Risposta: 3

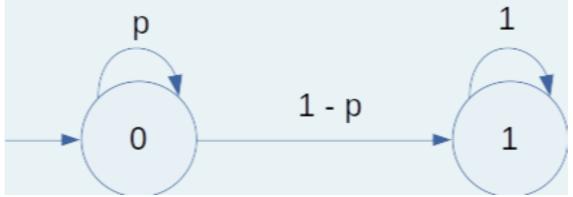
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il costo dello stato (fase) x è c(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta il completamento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



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

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

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

Risposta: 2

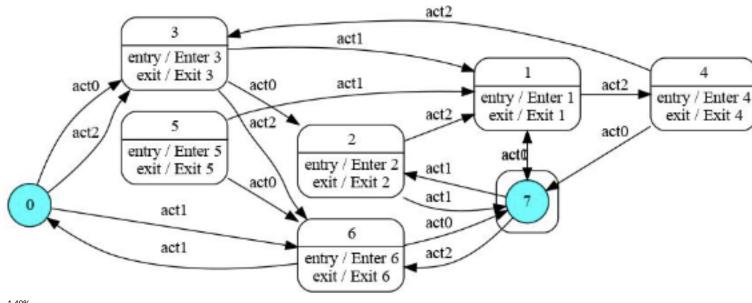
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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 act0Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?



1.40%

2.90%

3.70%

Risposta: 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.

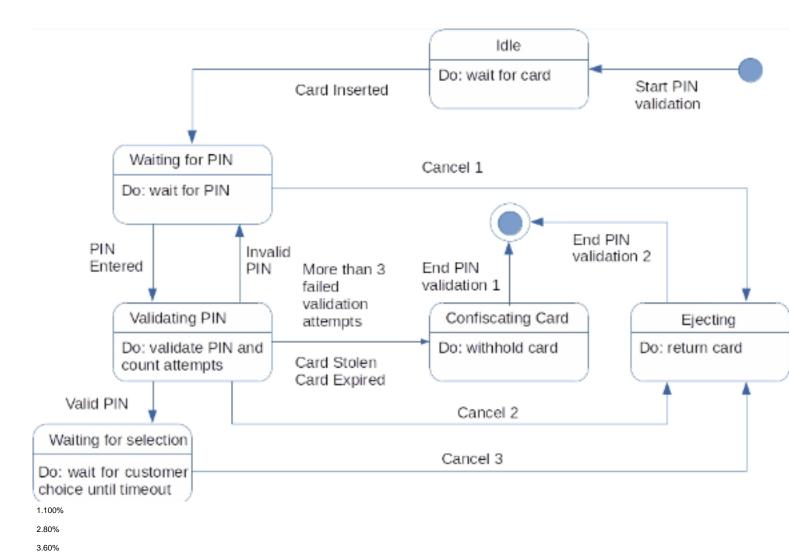
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

Test case 2: act0 act1 act2 act1 act2 act0 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?

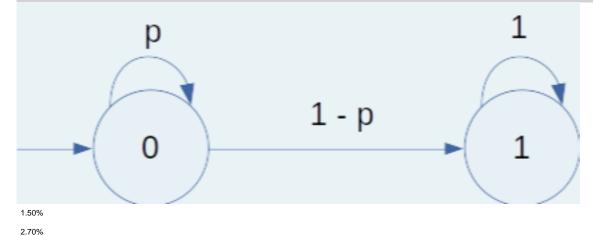


La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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 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?

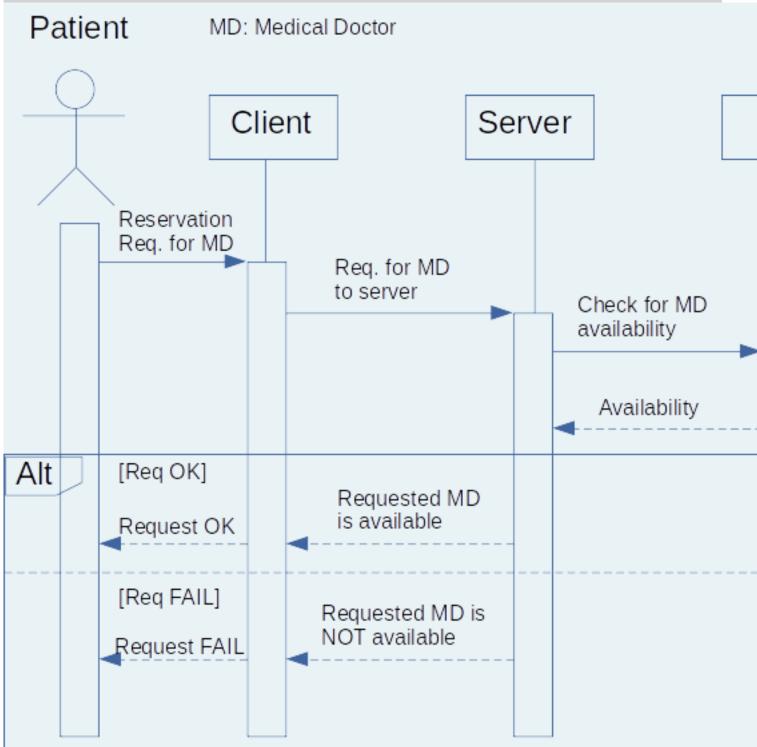


3.90%

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 figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità 0.1 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.



1.0.27

2 0 03

3.0.07

Risposta : 2

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.

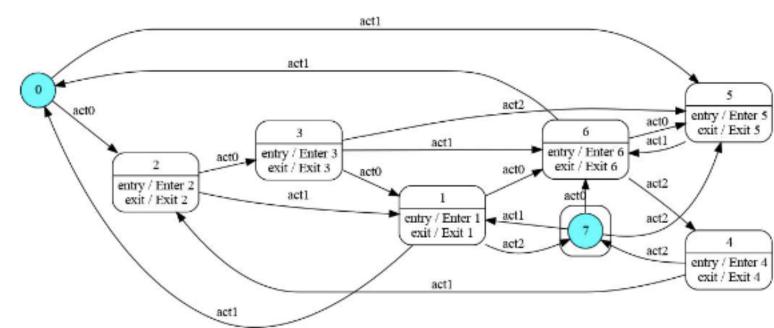
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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?



1.Transition coverage: 70%

2.Transition coverage: 40%

3.Transition coverage: 100%

Risposta : 2

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.

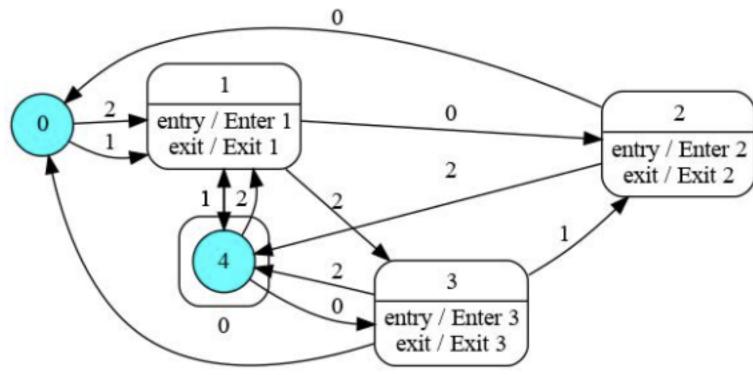
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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



1.State coverage: 70%

2.State coverage: 100%

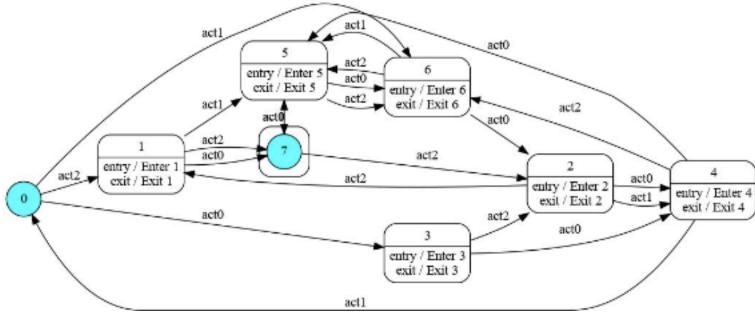
3.State coverage: 90%

Risposta: 3

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 act1 act2 act1 act2 act1 act2 act0 act1

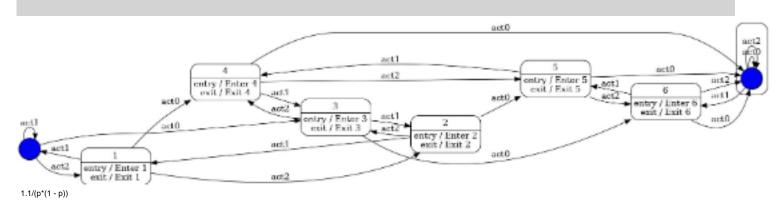
Test case 3: act1 act1 act2 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



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

Risposta: 1

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.



2.1/(1 - p)

3.(1 - p)/p

Risposta: 2

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.

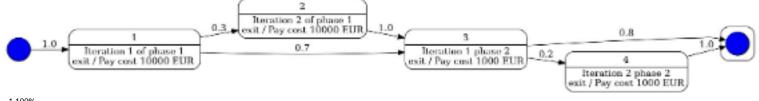
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act1 act1 act1

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

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

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



1.100%

2.80%

3.60%

Risposta: 2

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.

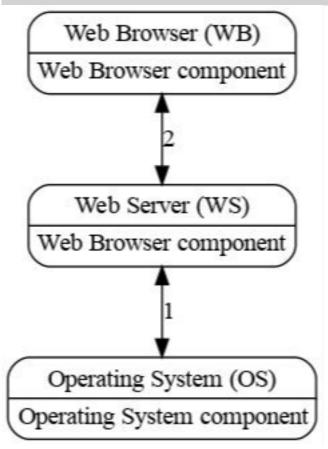
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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

Test case 2: act1

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

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



- 1.State coverage: 60%
- 2.State coverage: 75%
- 3.State coverage: 100%

Risposta : 2

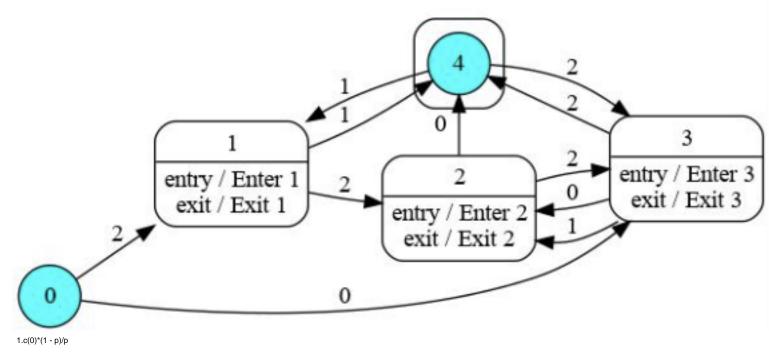
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il costo dello stato (fase) x è c(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta il completamento 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 $X = x(0), x(1), x(2), \\ e C(X) = c(x(0)) + c(x(1)) + c(x(2)) + \\ e C(X) = c(x(0)) + c(x(1)) + c(x(2)) + \\ e C(X) = c(x(0)) + c(x(1)) + c(x(2)) + \\ e C(X) = c(x(0)) + c(x(1)) + c(x(2)) + \\ e C(X) = c(x(0)) + c(x(1)) + c(x(2)) + \\ e C(X) = c(x(0)) + c(x(1)) + \\ e C(X) = c(x(0)) + c(x(1)) + \\ e C(X) = c(x(0)) + \\ e$

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



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

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

Risposta: 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.

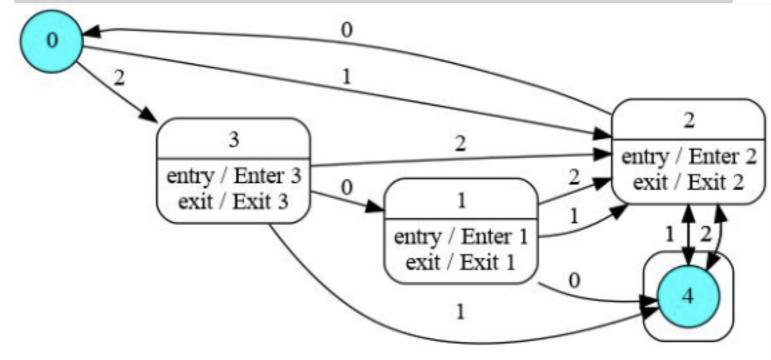
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: 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?



- 1.Transition coverage: 30%
- 2.Transition coverage: 40%
- 3.Transition coverage: 80%

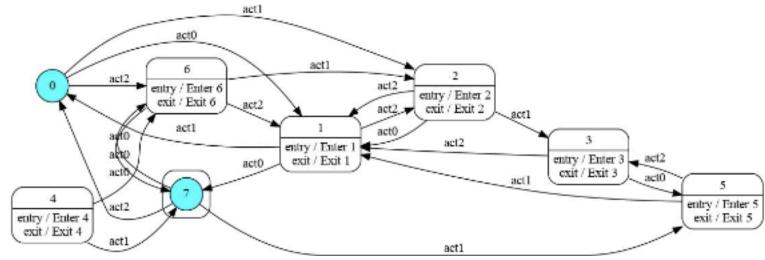
Risposta: 2

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 act0 act1 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



- 1.State coverage: 87%
- 2.State coverage: 100%
- 3.State coverage: 60%

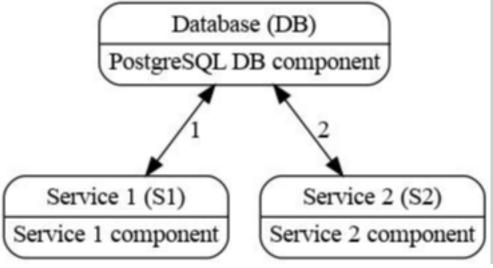
Risposta: 1

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

Uno scenario è una sequenza di stati.

Qual'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 ?



1.0.27

2.0.03

3.0.07

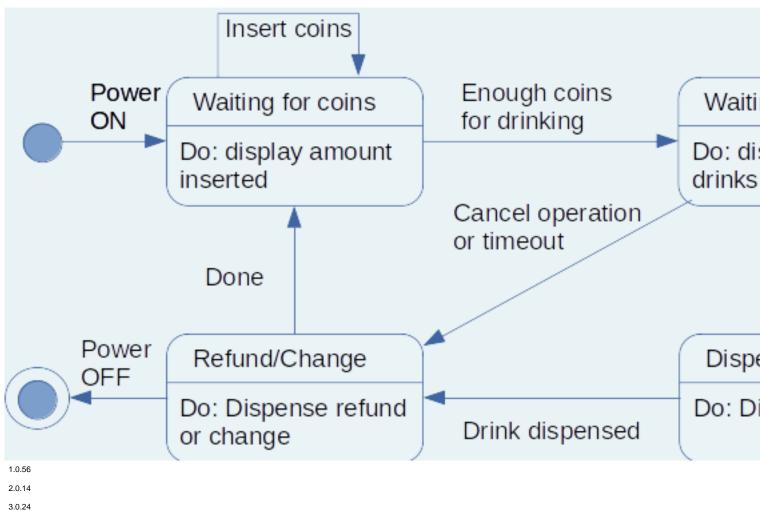
Risposta : 2

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.

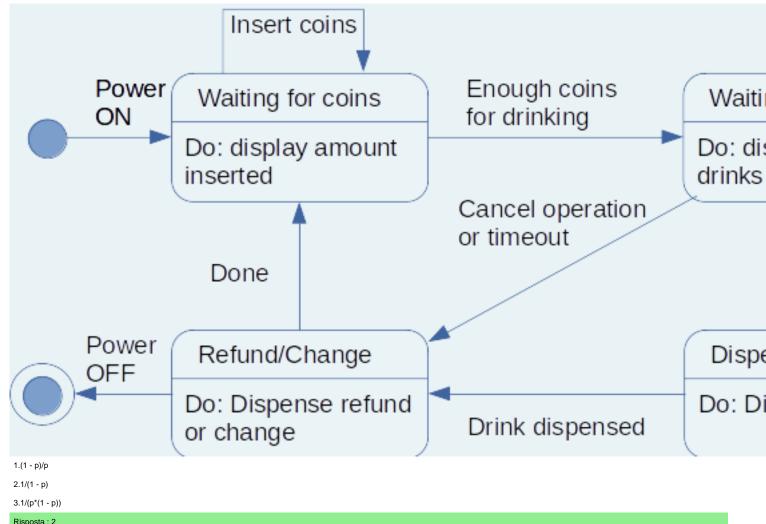
Ad esempio lo state diagram in figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità dello 0.3 di dover essere ripetuta (a causa di errori) ed F2 ha una probabilità 0.2 di dover essere ripetuta (a causa di errori).

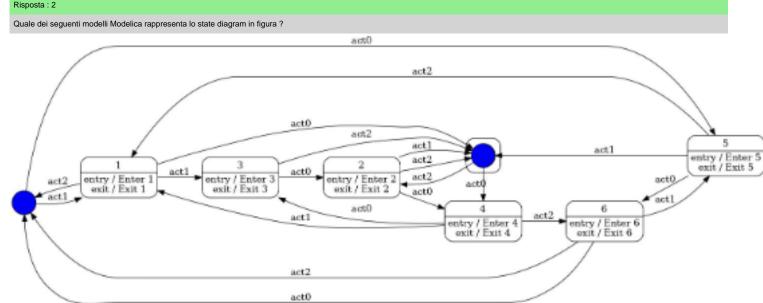
Uno scenario è una sequenza di stati.

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



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.





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 := 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) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) 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) == 0) then x := 2;elseif (pre(x) == 0) and (pre(u) == 0) then x := 3;elseif (pre(x) == 0) and (pre(u) == 0) then x := 3;elseif (pre(x) == 0) and (pre(u) == 0) then x := 3;elseif (pre(x) == 0) then x := 3;elseif (pre(x)

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 := 6;elseif (pre(x) == 2) and (pre(u) == 2) then x := 6;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) = 3) and (pre(u) = 2) then x := 1; elseif (pre(x) = 3) and (pre(u) = 3) an == 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;

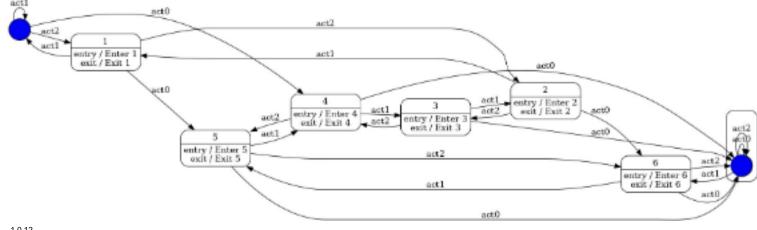
Risposta: 3

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 figuraRappresenta 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 seguenza 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)?



1.0.12

2.0.32

3.0.08

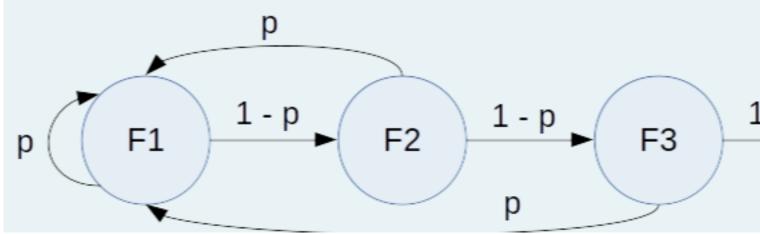
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta. Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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?



- 1.Transition coverage: 80%
- 2.Transition coverage: 60%
- 3.Transition coverage: 35%

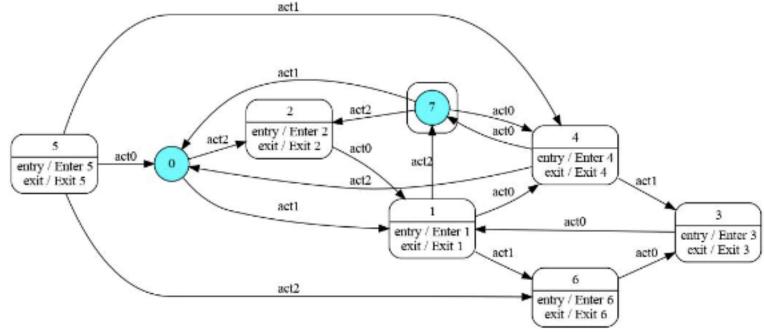
Risposta: 3

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

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



1.State coverage: 60%

2.State coverage: 90%

3.State coverage: 70%

Risposta : 2

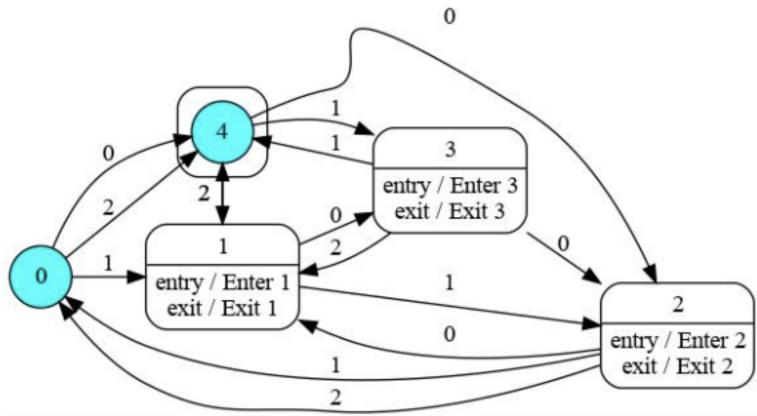
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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

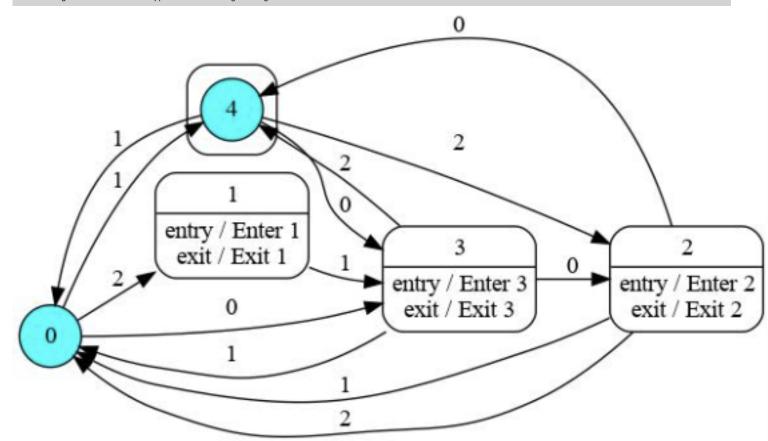
Test case 3: act0 act0 act0

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



- 1.State coverage: 25%
- 2.State coverage: 80%
- 3.State coverage: 60%

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



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) == 2) and (pre(u) == 0) then x := 0;elseif (pre(x) == 2) and (pre(u) == 0) 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 := 3;elseif (pre(x) == 1) and (pre(u) == 0) then x := 4;elseif (pre(x) == 1) and (pre(u) == 2) then x := 1;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pr

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) == 0) and (pre(u) == 2) 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) == 0) 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;

Risposta: 2

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

Web Browser

Forms and Query Manager

Database

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() then x := 0; elsewhen sample(0,T) then if (pre(x) == 0) and (pre(u) == 0) then x := 3; elseif (pre(x) == 0) and (pre(u) == 1) then x := 2; elseif (pre(x) == 1) and (pre(u) == 2) then x := 2; elseif (pre(x) == 1) and (pre(u) == 2) then x := 2; 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 := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 3) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) then x := 0; elseif (pre

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) and (pre(u) == 2) and (pre(u) == 0) then x := 4;elseif (pre(x) == 2) and (pre(u) == 2) and (pre(u) == 2) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 0;elseif (pre(x) == 3) and (pre(u) == 2) 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;

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() then x := 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) == 1) and (pre(u) == 2) then x := 0;elseif (pre(x) == 1) and (pre(u) == 2) then x := 2;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 := 1;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 2;elseif (pre

Risposta: 3

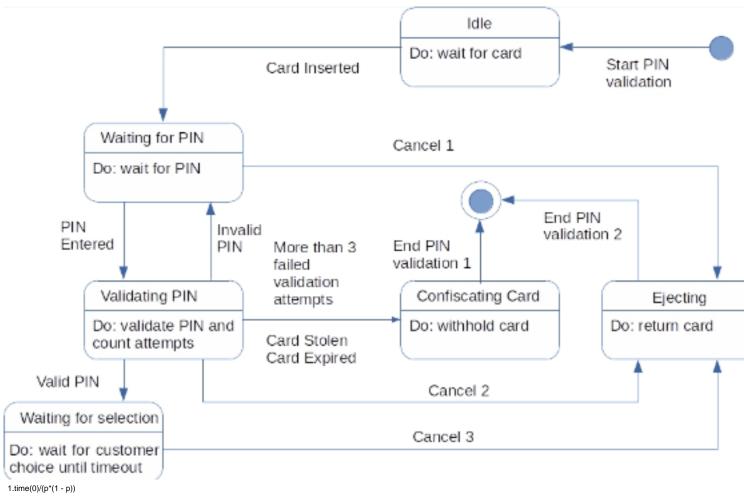
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il tempo necessario per completare la fase x è time(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta 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 $X = x(0), x(1), x(2), \dots$ è Time(X) = time(x(0)) + time(x(1)) + time(x(2)) + \dots

 $Ad\ esempio\ se\ X=0,\ 1\ abbiamo\ Time(X)=time(0)\ +\ time(1)=time(0)\ \ (poich\`e\ time(1)=0).$

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



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

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

Risposta: 3

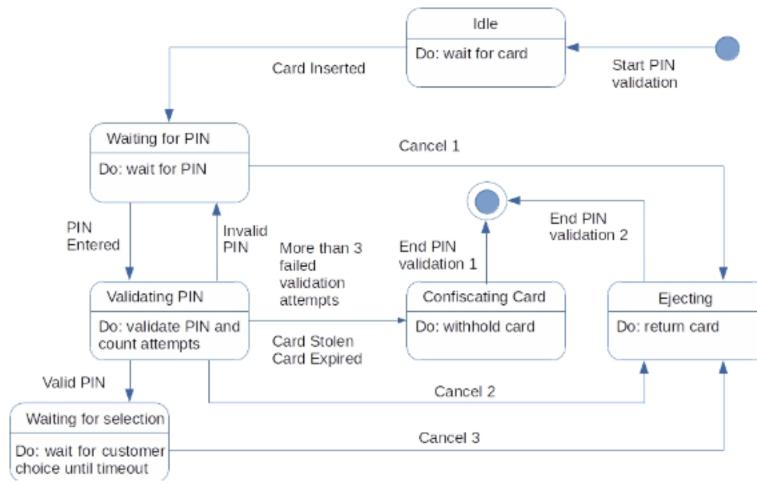
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: 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?



1.Transition coverage: 70%

2.Transition coverage: 30%

3.Transition coverage: 40%

Risposta: 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.

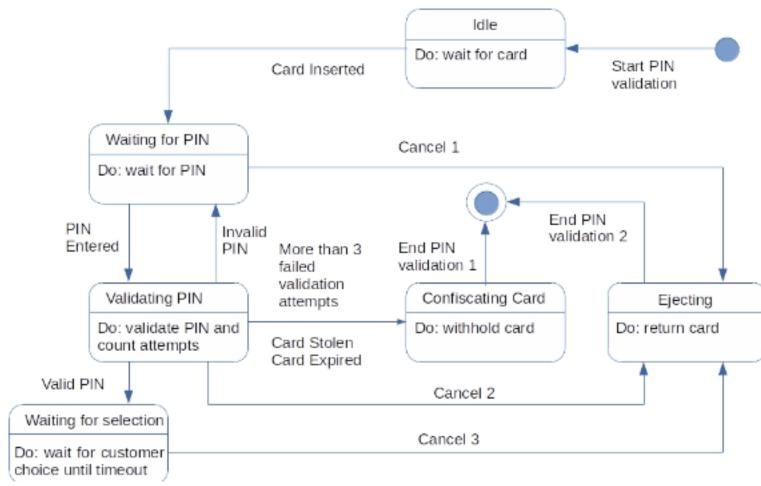
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: 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?



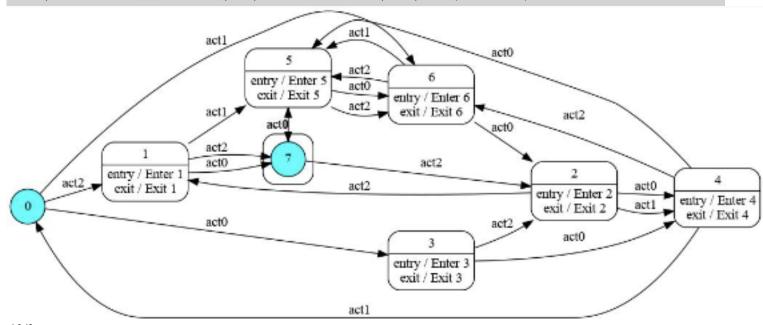
- 1.Transition coverage: 50%
- 2.Transition coverage: 30%
- 3.Transition coverage: 80%

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.

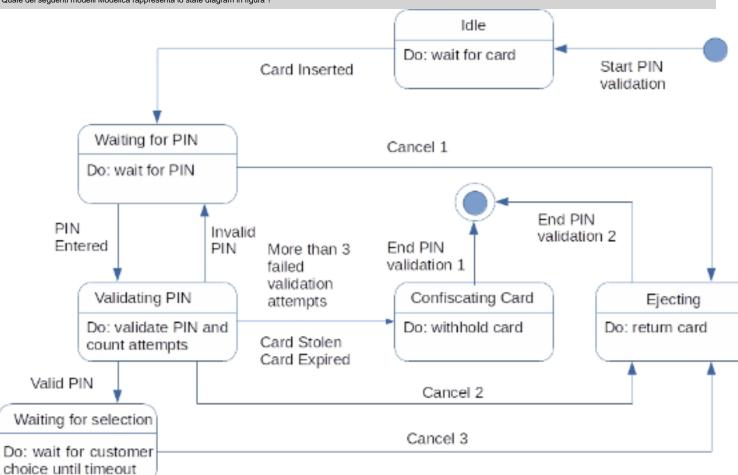
Ad esempio lo state diagram in figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità dello 0.4 di dover essere ripetuta (a causa di errori) ed F2 ha una probabilità 0.3 di dover essere ripetuta (a causa di errori).

Uno scenario è una sequenza di stati.

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



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



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) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 1) 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 := 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;

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) == 2) and (pre(u) == 2) then x := 0; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 3) and (pre(u) == 3) and (pre(u) == 3) and (pre(u) == 3) and (pre(u) == 4) and (pre(u) == 3) and (pre(u) == 2) 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) == 1) and (pre(u) == 2) then x := 3; elseif (pre(x) == 1) 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 := 3; elseif (pre(x) == 4) and (pre(u) == 5) then x := 0; elseif (pre(x) == 4) and (pr

Risposta : 2

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.

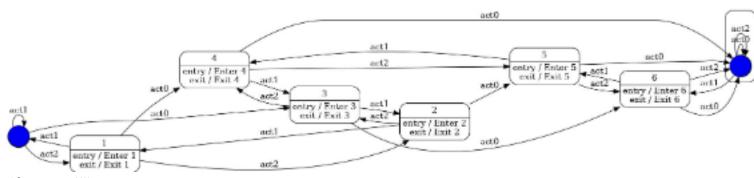
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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



1.State coverage: 80%

2.State coverage: 50%

3.State coverage: 100%

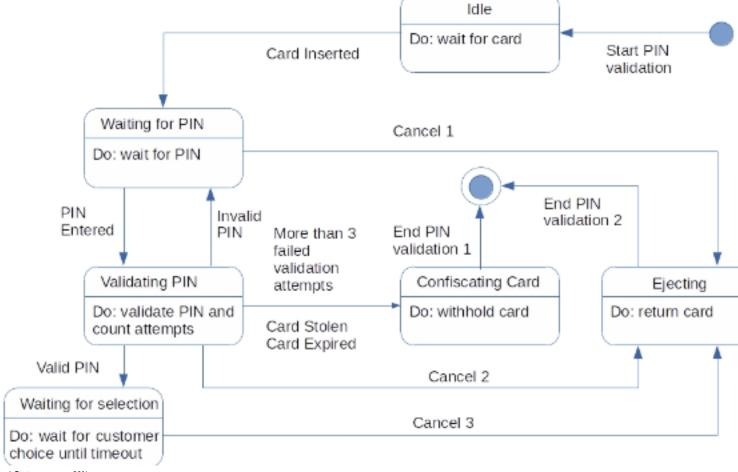
Risposta: 3

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



1.State coverage: 60%

2.State coverage: 87%

3.State coverage: 100%

Risposta : 2

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.

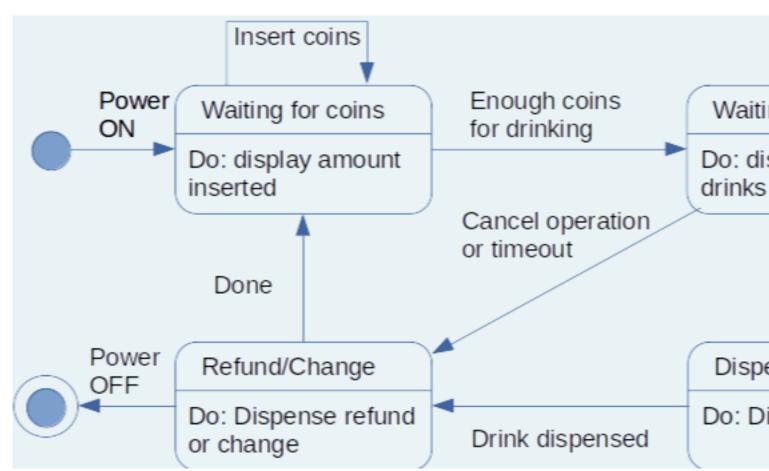
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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?



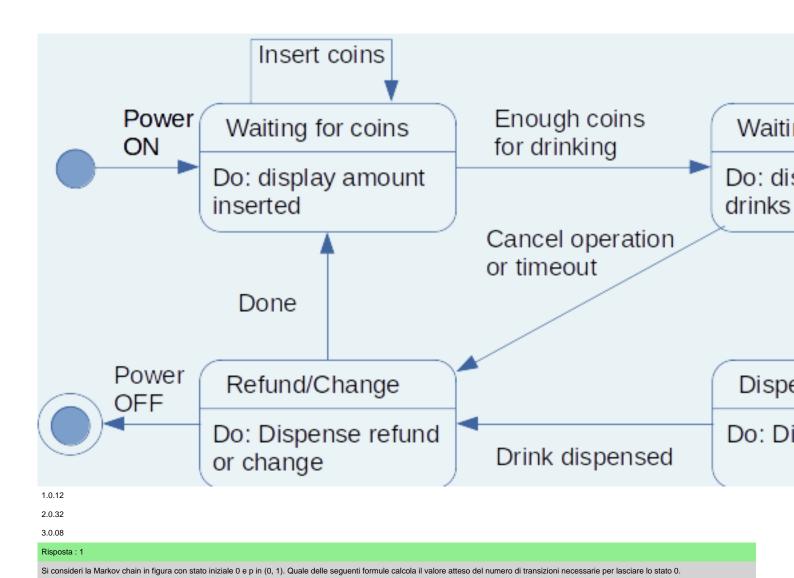
- 1.Transition coverage: 30%
- 2.Transition coverage: 80%
- 3.Transition coverage: 50%

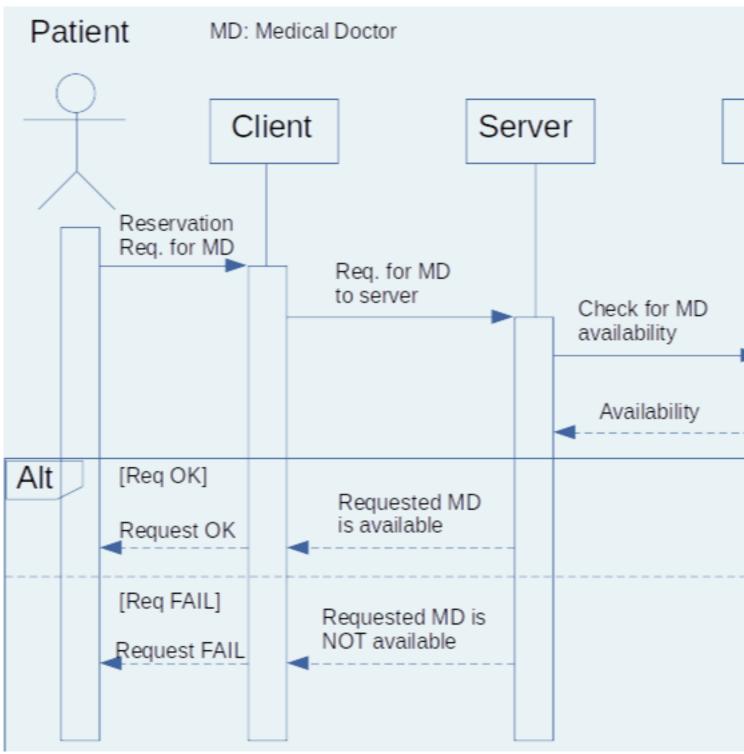
Un processo software può essere rappesentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del processo software e gli archi le transizioni da una fase all'altra. Gli archi sono etichettati con le probabilità della transizione e gli stati sono etichettati con il costo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

Ad esempio lo state diagram in figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità 0.2 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)?





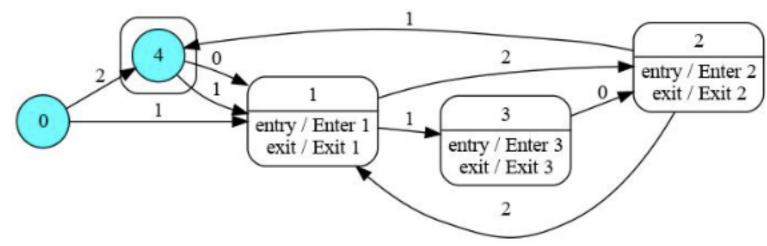
1.(1 - p)/p

2.1/(1 - p)

3.1/(p*(1 - p))

Risposta: 2

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



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) == 1) and (pre(u) == 2) then x := 0;elseif (pre(x) == 1) and (pre(u) == 2) then x := 2;elseif (pre(x) == 2) and (pre(u) == 2) then x := 4;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;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) == 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) then x := 0; elseif (pre(x) == 2) and (pre(u) == 2) 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;

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() then x := 0; elsewhen sample(0,T) then if (pre(x) == 0) and (pre(u) == 0) then x := 3; elseif (pre(x) == 0) and (pre(u) == 1) then x := 2; elseif (pre(x) == 1) and (pre(u) == 2) then x := 2; elseif (pre(x) == 1) and (pre(u) == 2) then x := 2; elseif (pre(x) == 3) and (pre(u) == 3) and (pre(u) == 3) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) == 1) then x := 0; elseif (pre(x) == 4) and (pre(u) =

Risposta: 1

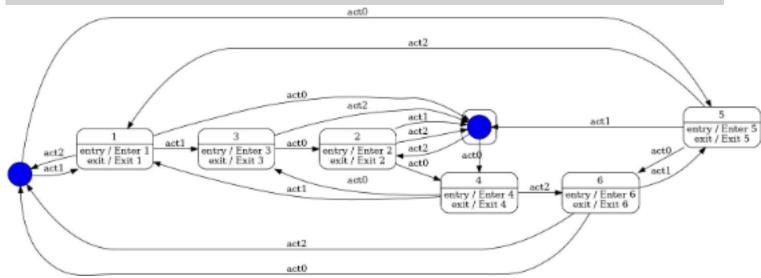
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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?



- 1.Transition coverage: 30%
- 2.Transition coverage: 70%
- 3.Transition coverage: 40%

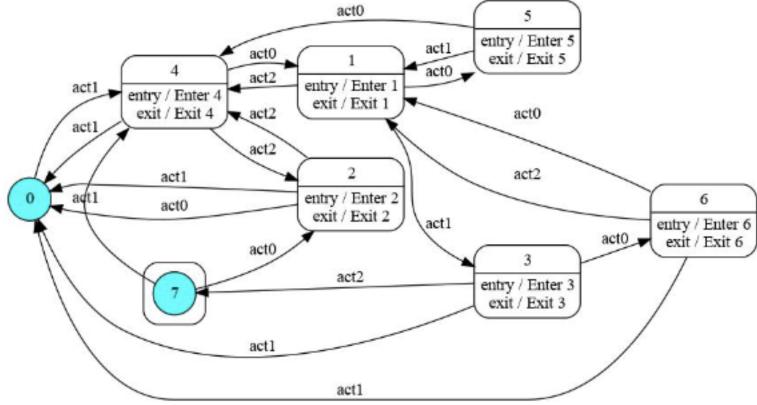
Risposta: 3

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



- 1.State coverage: 80%
- 2.State coverage: 60%
- 3.State coverage: 25%

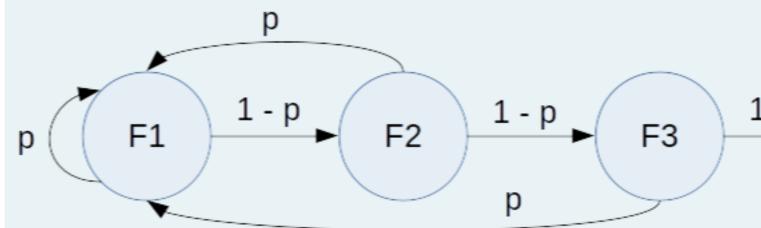
Risposta: 3

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 figuraRappresenta 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)?



1.0.28

2.0.42

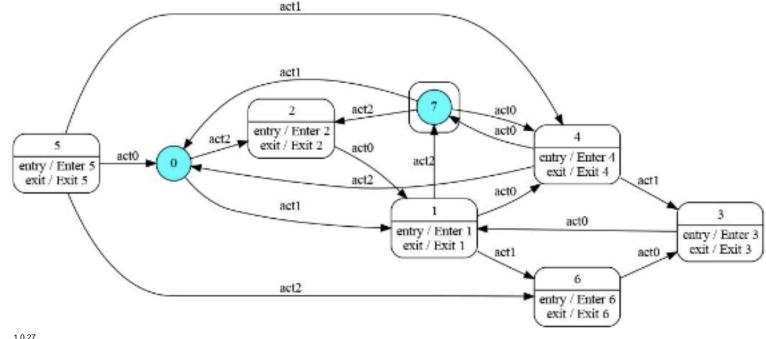
3.0.12

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

Uno scenario è una sequenza di stati.

Qual'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 ?

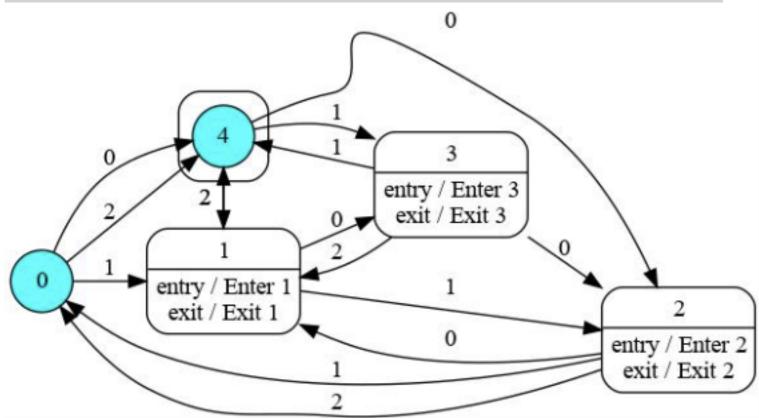


1.0.27

2.0.07

Risposta: 3

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



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 := 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 := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) and ((pre(x) == 3) and (pre(u) == 0) then x := 2; elseif (pre(x) == 4) and (pre(u) == 0) then x := 2; elseif (pre(x) == 4) and (pre(u) == 0) then x := 1; elseif (pre(x) == 4) and (pre(u) == 0) then x := 2; elseif (pre(x) == 4) and (pre(u) == 4) and (pre(u == 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;

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 := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 4;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 1) and (pre(u) == 1) and (pre(u (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) = 3) a == 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); //

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) == 2) then x := 3;else (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) == 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) == 2) and (pre(u) == 1) then x := 0; elseif (pre(x) == 2) then x := 0; elseif (pre(x) == 3) th == 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;

Risposta: 1

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

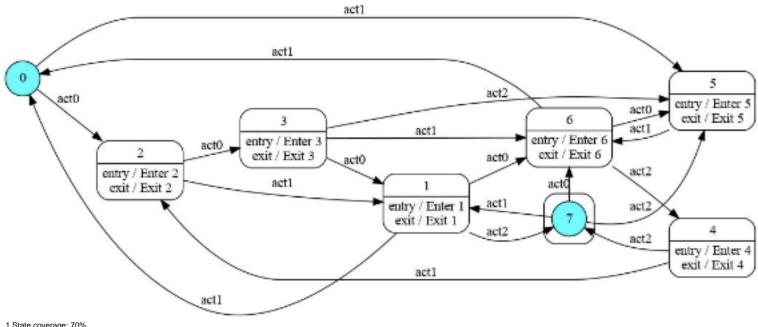
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

Test case 1: act1 act2 act1 act2 act2

Test case 2: act2 act0.

Test case 3: act0 act0 act0.

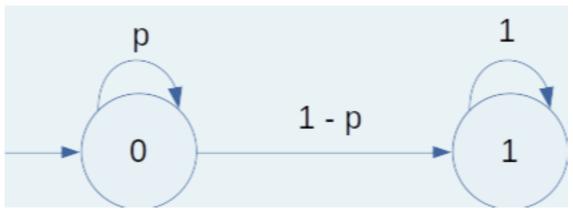
Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



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

Risposta: 3

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



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) == 2) and (pre(u) == 2) then x := 0; 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) == 2) then x := 2; 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 := 1;elseif (pre(x) == 2) and (pre(u) == 0) then x := 0;elseif (pre(x) == 2) and (pre(u) == 0) 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 := 3;elseif (pre(x) == 1) and (pre(u) == 0) then x := 4;elseif (pre(x) == 1) and (pre(u) == 2) then x := 1;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 2) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) =

Risposta: 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.

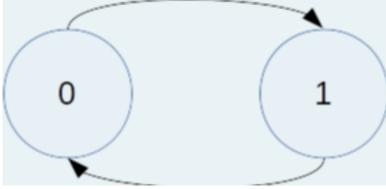
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: 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?



- 1.Transition coverage: 80%
- 2.Transition coverage: 40%
- 3.Transition coverage: 30%

Risposta : 2

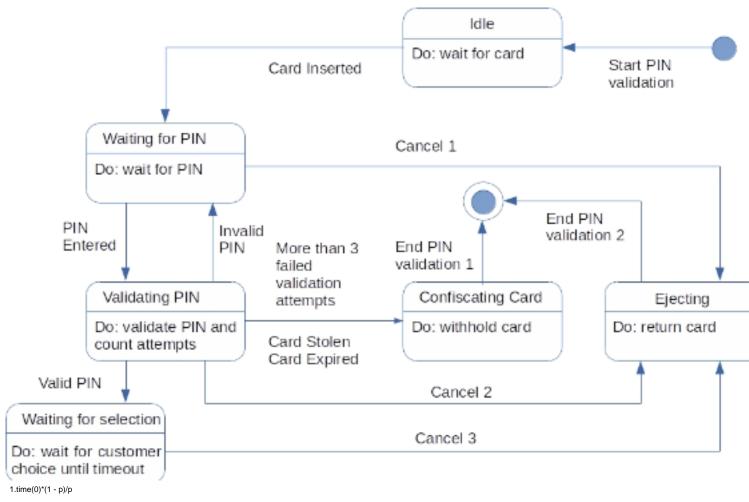
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il tempo necessario per completare la fase x è time(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta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

 $Quindi \ il \ costo \ Time(X) \ della \ sequenza \ di \ stati \ X = x(0), \ x(1), \ x(2), \ \ \dot{e} \ Time(X) = time(x(0)) + time(x(1)) + time(x(2)) + ...$

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



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

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

Risposta: 3

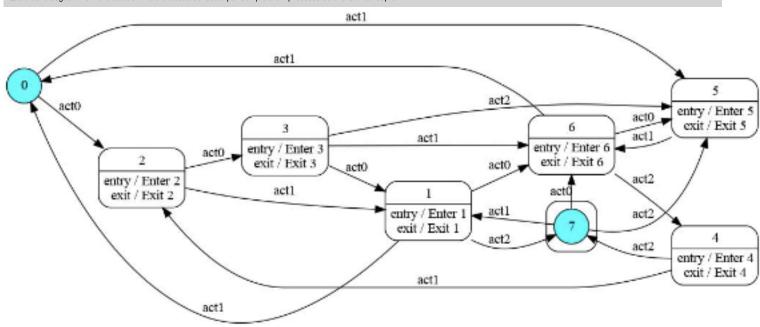
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il costo dello stato (fase) x è c(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta il completamento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



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

Risposta: 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.

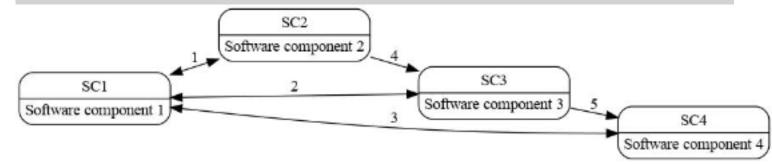
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: 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?



- 1.Transition coverage: 60%
- 2.Transition coverage: 80%
- 3.Transition coverage: 35%

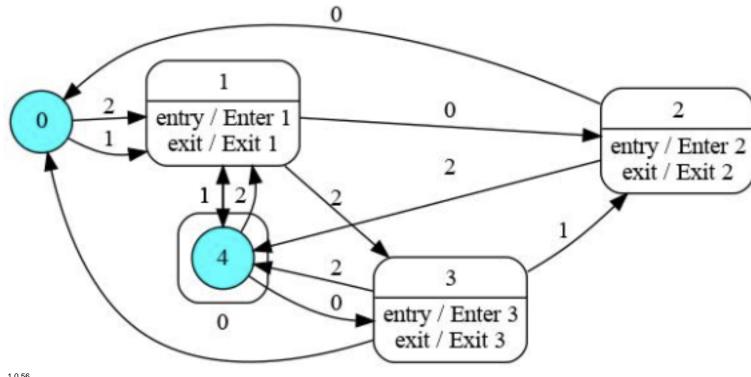
Risposta: 3

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 figuraRappresenta 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?



1.0.56

2.0.14

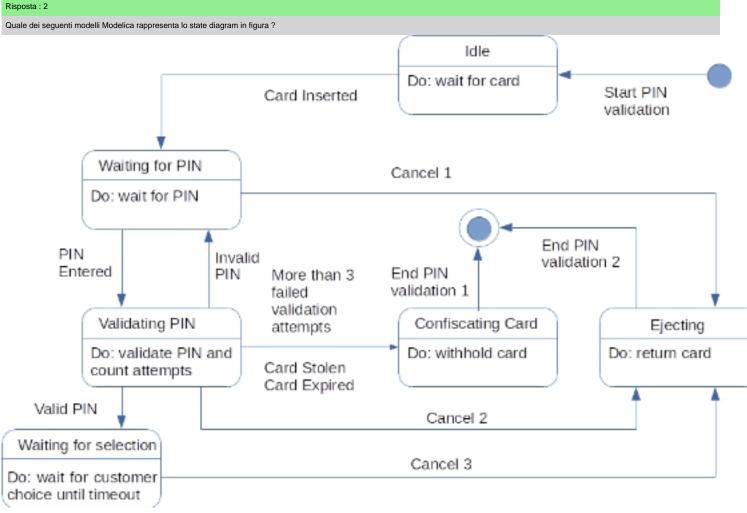
3.0.24

Risposta: 1

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 act1 act1 act0 act2 entry / Enter 5 act0 exit / Exit 5 entry / Enter 6 act2 exit / Exit 6 act1 act2 act0 act0 act2 act0 act2 entry / Enter 1 act2 act2 act0 exit / Exit 1 entry / Enter 2 entry / Enter 4 act1 exit / Exit 2 exit / Exit 4 act0 act2 act0 3 entry / Enter 3 exit / Exit 3 act1 1.State coverage: 80% 2.State coverage: 100% 3.State coverage: 50% Risposta: 2 Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ? Idle Do: wait for card Start PIN



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) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) and (pre(u) == 1) then x := 2;elseif (pre(x) == 0) 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) == 2) then x := 3;elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;elseif (pre(x) == 2) and (pre(u) == 2) then x := 3;elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;elseif (pre(x) == 2) then x := 3;elseif (p

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() 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) == 1) and (pre(u) == 2) then x := 4; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 2) 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 := 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) == 1) then x := 2; elseif (pre(x) == 2) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 1; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 4) and (pre(

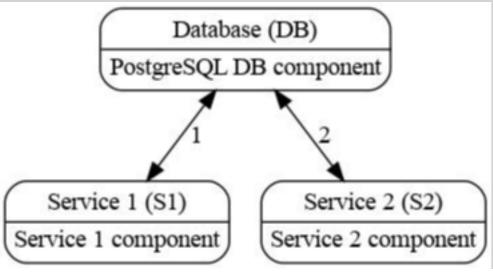
Risposta · 2

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 figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità 0.2 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)?



1.0.32

2.0.12

3.0.08

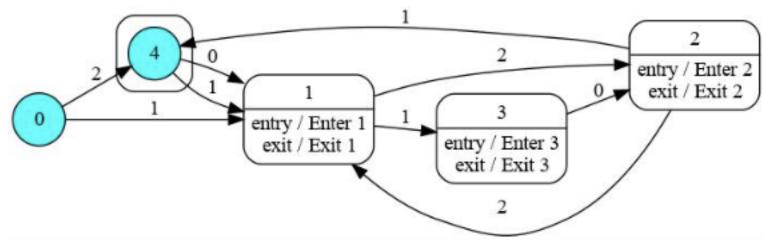
Risposta : 2

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?



- 1.Transition coverage: 50%
- 2.Transition coverage: 75%
- 3.Transition coverage: 25%

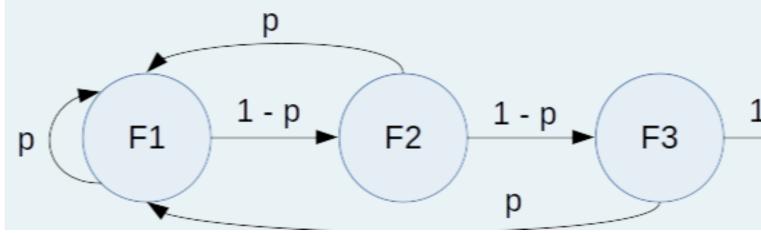
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il costo dello stato (fase) x è c(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta il completamento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



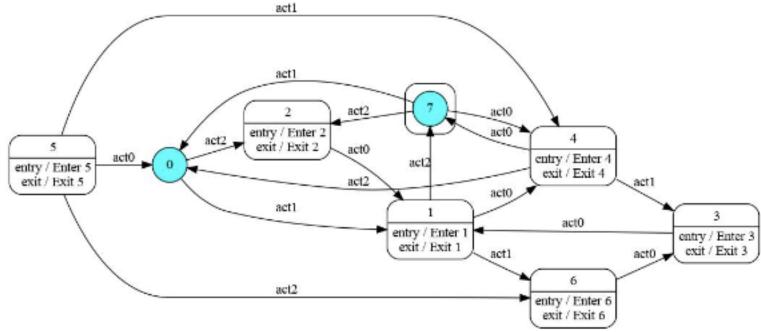
1.c(0)*(1 - p)/p

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

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

Risposta: 3

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



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

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() then x := 0; elsewhen sample(0,T) then if (pre(x) == 0) and (pre(u) == 0) then x := 3; elseif (pre(x) == 0) and (pre(u) == 1) then x := 4; elseif (pre(x) == 1) and (pre(u) == 1) then x := 3; 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) == 2) then x := 0; elseif (pre(x) == 3) and (pre(u) == 2) then x := 4; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 3) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 1) then x := 2; elseif (pre(x) == 4) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 2) then x := 2; elseif (pre(x) == 4) and (pre(u) == 4) and (pre(

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() then x := 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) == 2) then x := 0;elseif (pre(x) == 2) and (pre(u) == 2) then x := 4;elseif (pre(x) == 2) and (pre(u) == 2) then x := 4;elseif (pre(x) == 2) and (pre(u) == 2) then x := 4;elseif (pre(x) == 2) and (pre(u) == 2) then x := 2;elseif (pre(x) == 3) and (pre(u) == 0) then x := 3;elseif (pre(x) == 3) and (pre(u) == 2) then x := 3;elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;else x := pre(x); // defaultend if;end when;end FSA;

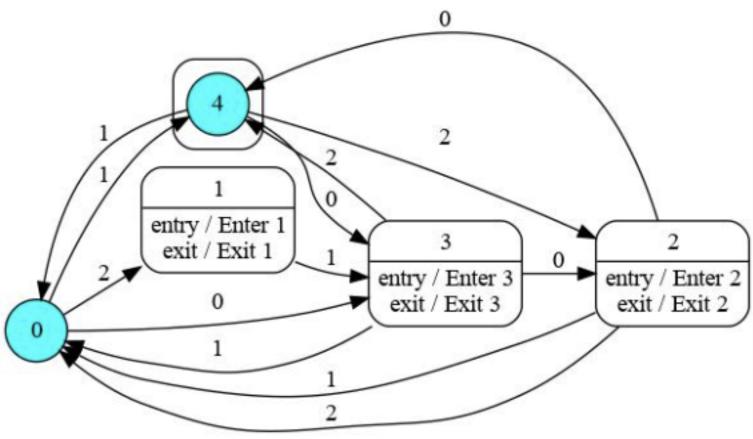
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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 act0 act0 act2 act2 act2 act1

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

Test case 3: act0 act1 act0 act0 act0 act0 act2 act1 act0 act2 act2 act2 act2 act2 act0 act1 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



1.State coverage: 87%

2.State coverage: 50%

3.State coverage: 100%

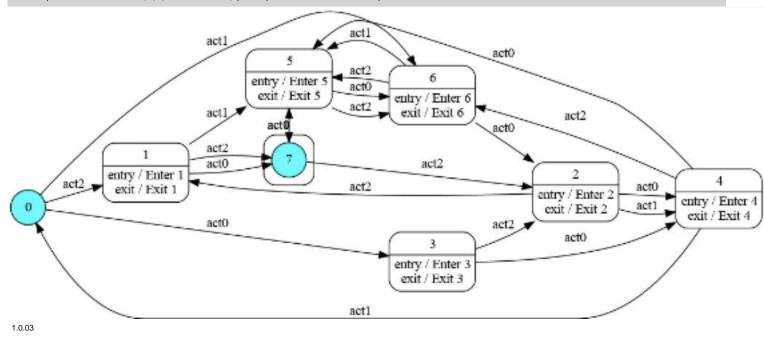
Risposta: 1

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

Uno scenario è una sequenza di stati.

Qual'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 ?



2.0.27

3.0.07

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 figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 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?

Web Browser

Forms and Query Manager

Database

1.0.56

2.0.24

3.0.14

Risposta: 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.

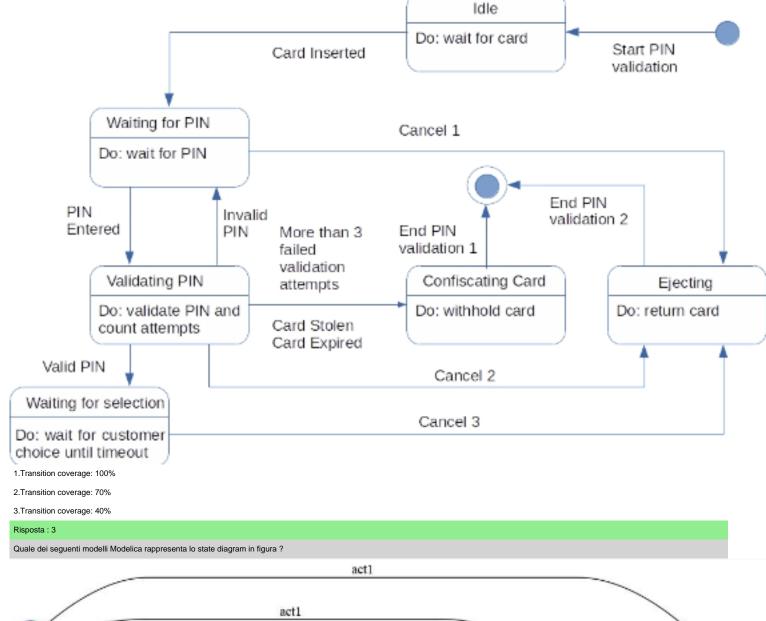
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

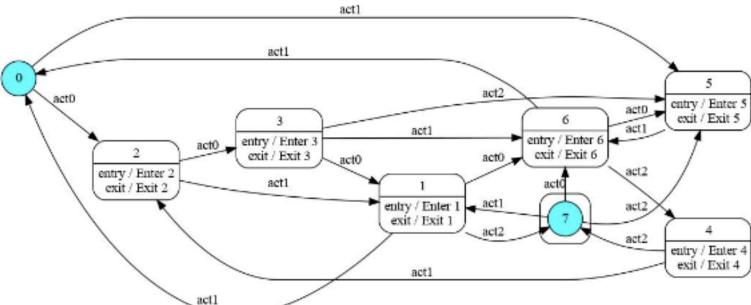
Test case 1: act1 act2 act0 act1

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?





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

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() then x := 0;elsewhen sample(0,T) thenif (pre(x) == 0) and (pre(y) == 1) then y := 1;elseif (pre(y) == 0) then y := 1;elseif (pre(y) == 0) then y := 2;elseif (pre(y) == 1) and (pre(y) == 1) then y := 3;elseif (pre(y) == 2) then y := 3;elseif (pre(y) == 2) and (pre(y) == 2) then y := 3;elseif (pre(y) == 3) and (pre(y) == 4) and (pre(y) == 3) and (pre

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() 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) == 1) and (pre(u) == 2) then x := 0;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 1) and (pre(u) == 2) then x := 3;elseif (pre(x) == 3) and (pre(u) == 2) then x := 3;elseif (pre(x) == 3) and (pre(u) == 3) and (pre(u) == 3) then x := 3;elseif (pre(x) == 4) and (pre(u) == 3) then x := 3;elseif (pre(x) == 4) and (pre(u) == 3) then x := 3;elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;elseif (pre(x) == 4) and (pre(u) == 2) then x := 3;elseif (pre(x) == 4) 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

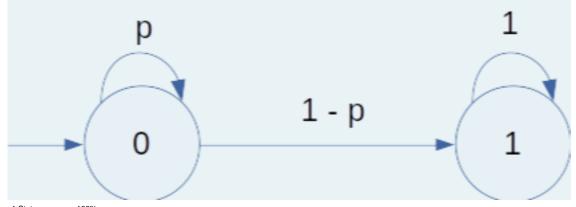
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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 act1 act1 act0 act2 act2 act2 act0

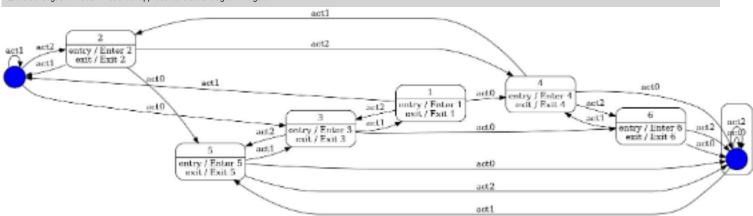
Test case 3: act1 act2 act2 act1 act0 act1 act0 act1 act2 act2 act0 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



- 1.State coverage: 100%
- 2.State coverage: 87%
- 3.State coverage: 50%

Risposta : 2

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



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) == 1) then x := 3;elseif (pre(x) == 1) 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 := 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) == 3) and (pre(u) == 2) then x := 1;elseif (pre(x) == 4) and (pre(u) == 4) and (pre(u) == 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 := 4; elseif (pre(x) == 0) 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) == 3) and (pre(u) == 3) and (pre(u) == 2) then x := 3; elseif (pre(x) == 3) and (pre(u) == 3) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 2) then x := 0; elseif (pre(x) == 4) and (pre(u) == 4) and (pre

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() then x := 0; elsewhen sample(0,T) then if (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) == 0) then x := 3; elseif (pre(x) == 0) then x := 3; elseif (pre(x) == 1) and (pre(u) == 0) then x := 3; elseif (pre(x) == 0) then x := 4; elseif (pre(x) == 0)

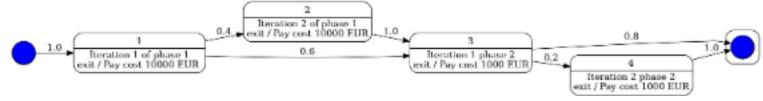
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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 act0 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



1.State coverage: 75%

2.State coverage: 50%

3.State coverage: 100%

Risposta: 2

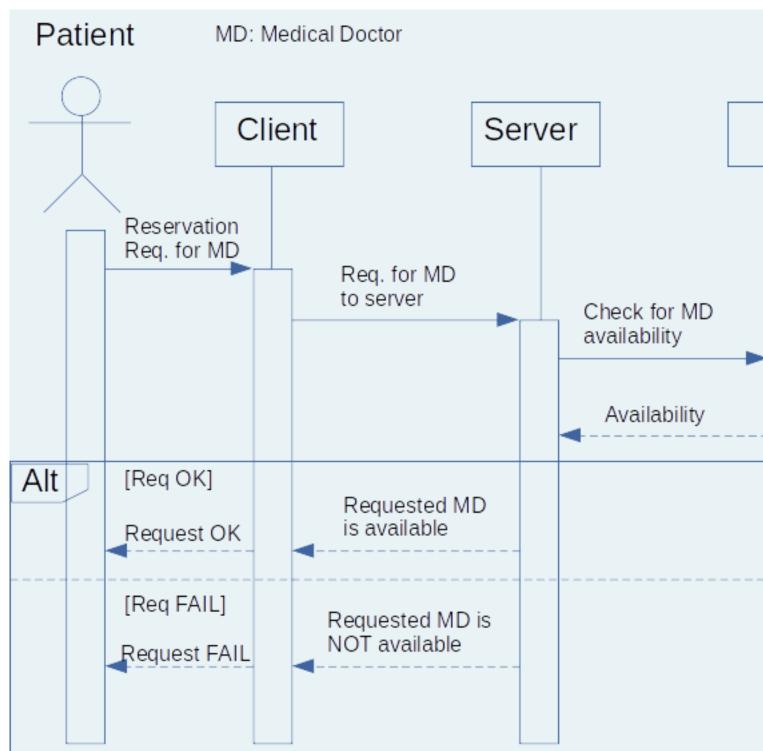
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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 act0 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



- 1.State coverage: 100%
- 2.State coverage: 75%
- 3.State coverage: 60%

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

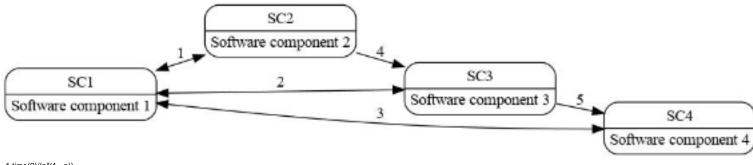
fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappreenta il completamento del processo software, e quindi time(1) = 0.

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.

 $Quindi il costo Time(X) \ della \ sequenza \ di \ stati \ X = x(0), \ x(1), \ x(2), \ \ \grave{e} \ Time(X) = time(x(0)) + time(x(1)) + time(x(2)) + ...$

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

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



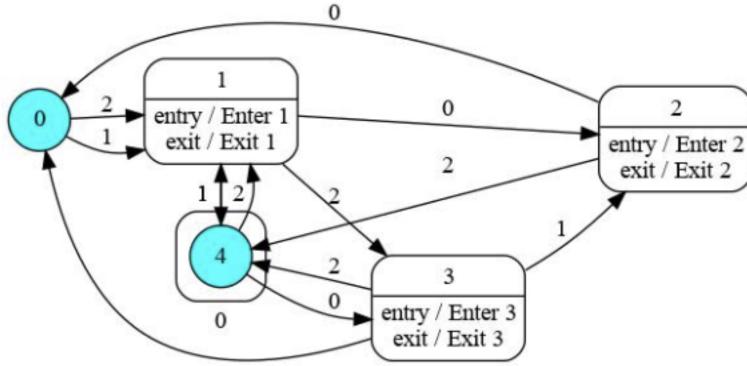
1.time(0)/(p*(1 - p))

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

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

Risposta: 3

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.



1.1/(1 - p)

2.1/(p*(1 - p))

3.(1 - p)/p

Risposta: 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.

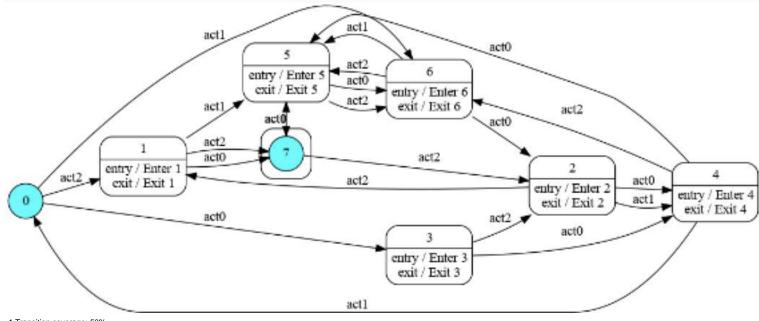
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

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



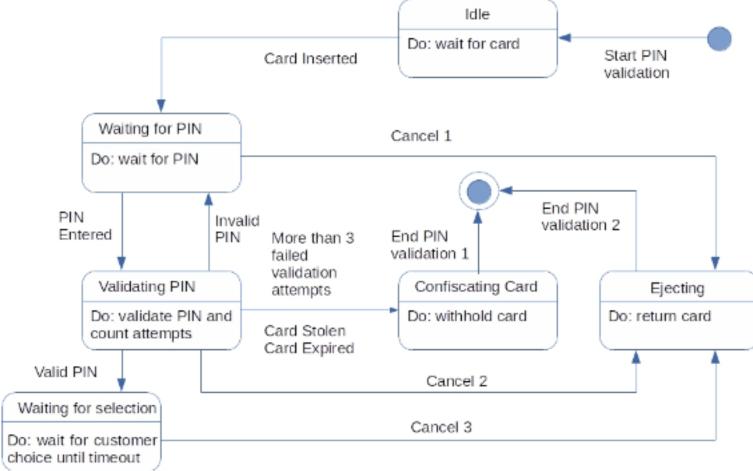
- 1.Transition coverage: 50%
- 2.Transition coverage: 75%
- 3.Transition coverage: 100%

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 figuraRappresenta un processo software con 2 fasi F1 ed F2. F1 ha costo 10000 EUR ed F2 ha costo 1000 EUR. F1 ha una probabilità 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 seguenza di stati.

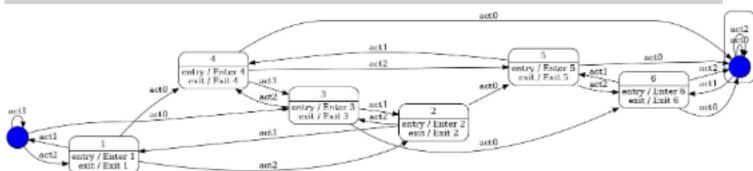
Qual'e' la probabilità dello scenario: 1, 2, 3? In altri terminti, qual'è la probabilità che non sia necessario ripetere la prima fase (ma non la seconda)?



3.0.28

Risposta: 3

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



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

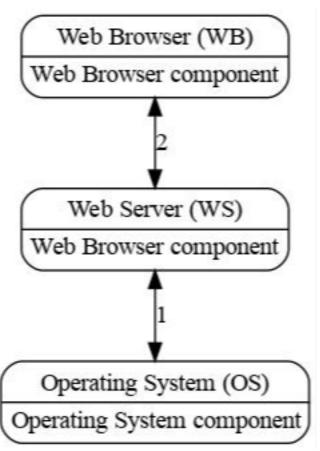
Risposta: 3

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:

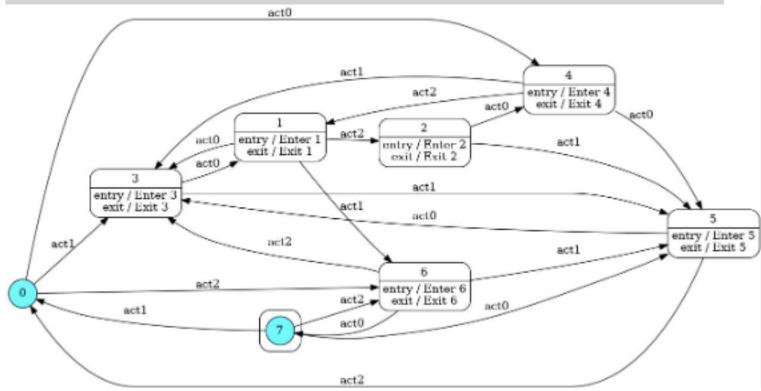
Test case 1: act1 act2 act0

Test case 2: act2 act2 act2 act2 act2 act2 act0



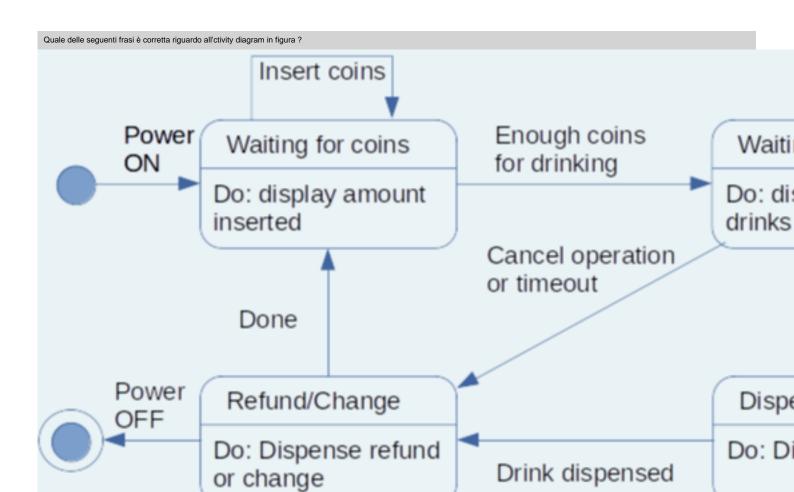
- 1.Transition coverage: 35%
- 2.Transition coverage: 90%
- 3.Transition coverage: 50%

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?



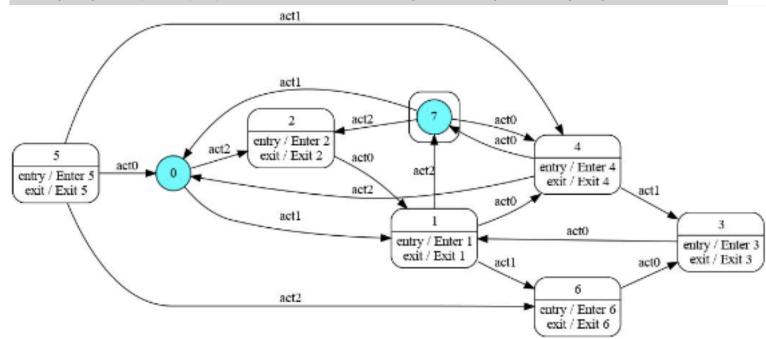
- 1.Se ci sono abbastanza monete è sempre possibile ottenere la bevanda selezionata.
- 2. Anche se ci sono abbastanza monete potrebbe non essere possibile ottenere la bevanda selezionata.
- 3.Una volta inserite le monete bisogna necessariamente consumare almeno una bevanda.

Risposta: 1



- 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.

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 ?



- 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

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 \le 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]) \le 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].
```

act1 act2 act1 act0 act0 entry / Enter 1 entry / Enter 4 2 exit / Fxit 1 exit / Tixit 4 act2 entry / Enter 2 exit / Exit 2 act2 act I act0 entry / Enter 5 entry / Enter 3

entry / Enter 6

exit / Exit 6

act0

act1

act2

1.50%

act1

act0

2 100% 3 80%

Risposta: 2

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:

exit / Exit 5

Quale delle seguenti è la branch coverage conseguita?

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

act2

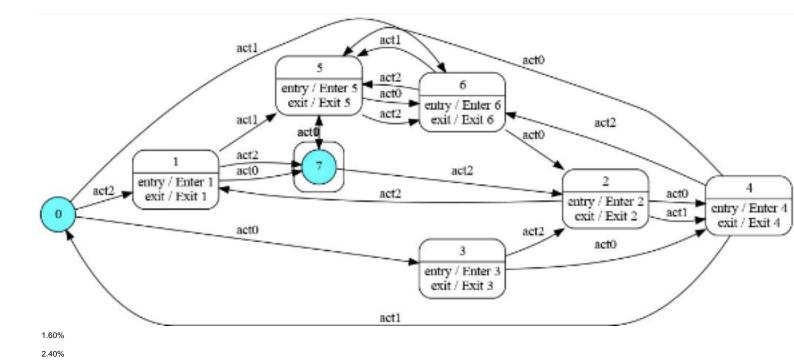
act0

2) Start PIN validation, card inserted, PIN Entered, Cancel 2, End PIN Validation 2Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra

act2

act]

exit / Exit 3



3.80%

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 2Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra

Web Browser

Forms and Query Manager

Database

1.90%

2.80%

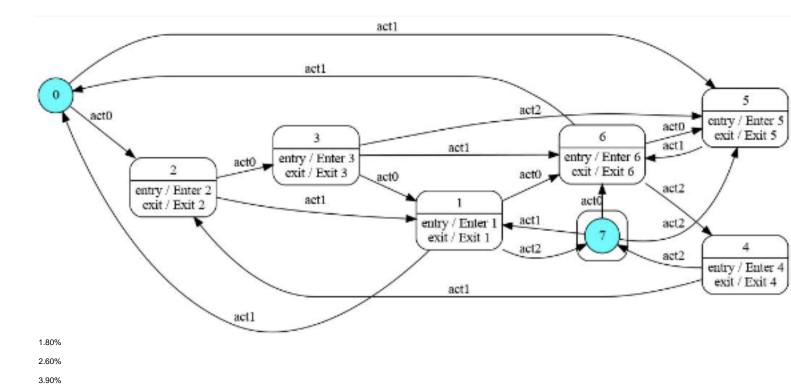
3.60%

Risposta: 1

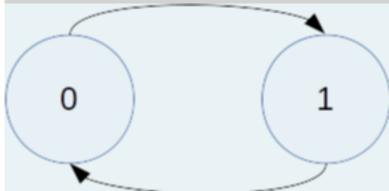
La state coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) rggiunti almeno una volta.

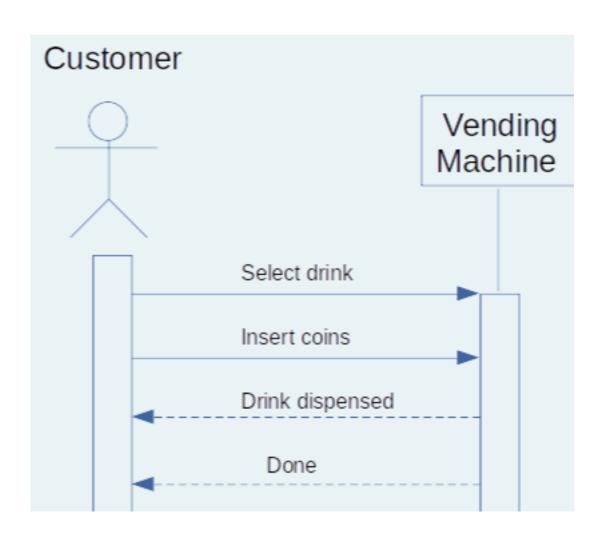
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

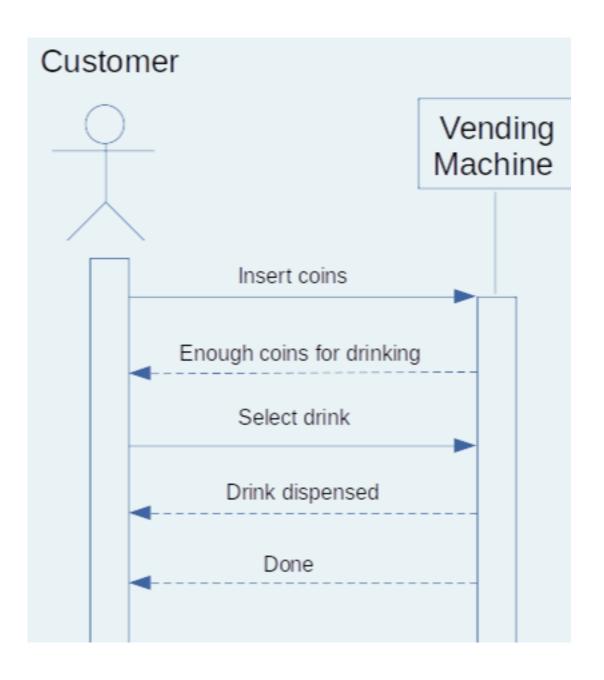
- 1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2;
- 2) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2;
- 3) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered,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

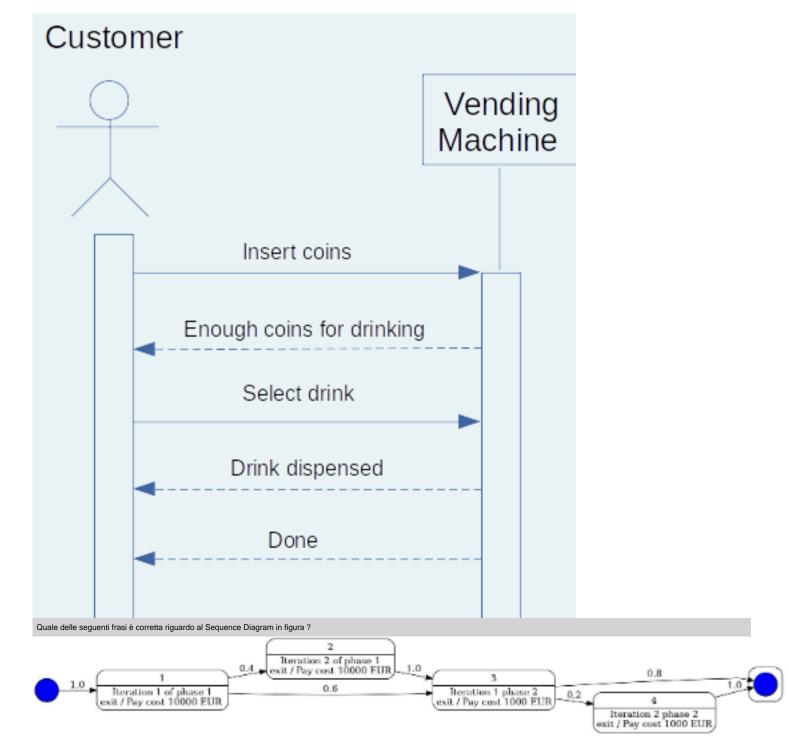


Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?





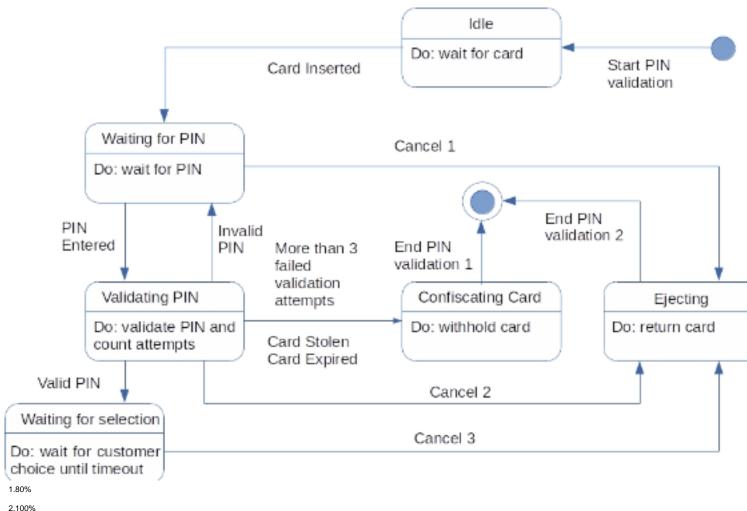




- 1.Il paziente richiede al server una visita con uno specifico medico e, dopo una verifica sul database, riceve conferma dal server della disponibilità o meno del medico richiesto.
- 2. Periodicamente il client comunica ai pazienti le disponibilità dei medici.
- 3.Il paziente richiede al client una visita con uno specifico medico e, dopo una verifica sul database, riceve conferma dal client della disponibilità o meno del medico richiesto.

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



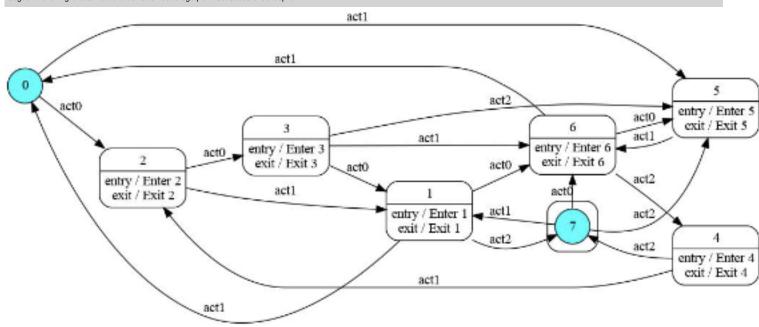
3.90%

1.100%

Risposta: 2

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.

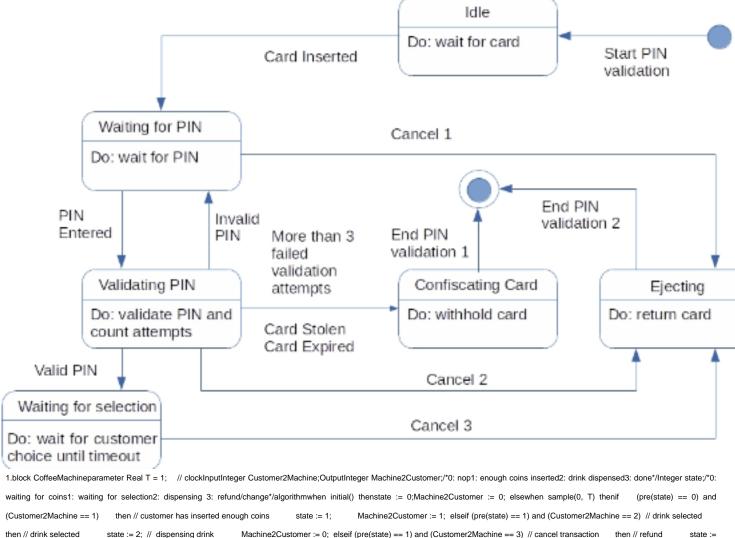
- Si consideri lo state diagram in figura ed il seguente insieme di test cases:
- 1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2
- 2) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 2, End PIN Validation 2
- 3) Start PIN validation, 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



3.90%



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?



then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction 0: // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 3: Machine2Customer := 2:elseif (pre(state) == 3) // refund/change state := 0: Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); CoffeeMachine:

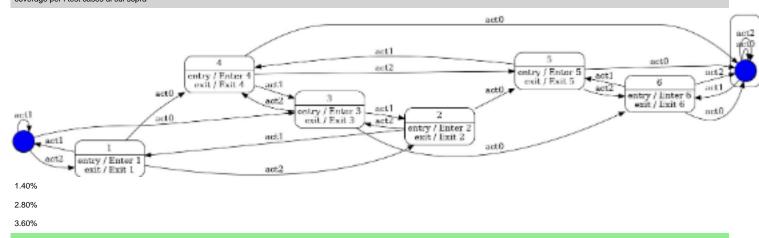
2.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer/2Machine;OutputInteger Machine2Customer/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer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

Risposta: 2

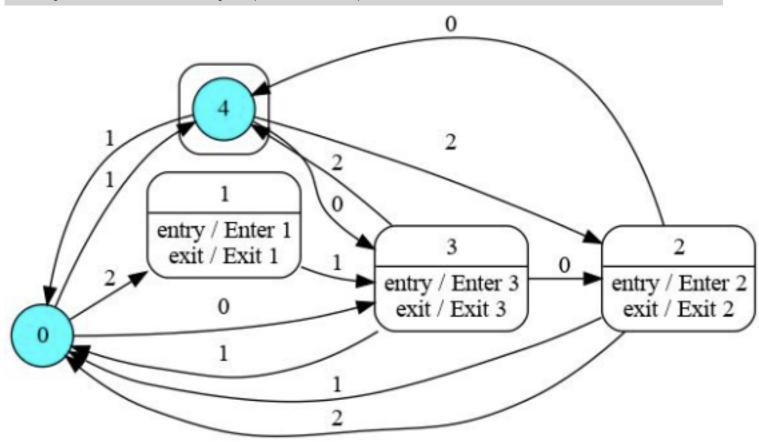
La transition coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta. Si consideri lo state diagram in figura ed il seguente insieme di test cases:

- 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, 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



Si consideri la seguente Markov Chain:

Quale dei seguenti modelli Modelica fornisce un modello ragionevole per la Markov Chain di cui sopra?



1.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 0, 1-p, 0;0, p, 1-p, 0;p, 0, 0, 0, 0, 0, 0, 0]; Integer x; Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState]; algorithmwhen initial() thenstate1024 := Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020); x := F1;r1024 := 0; elsewhen sample(0,1) then(r1024, state1024) := Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024)); if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; else x := End;end if; end when; end System;

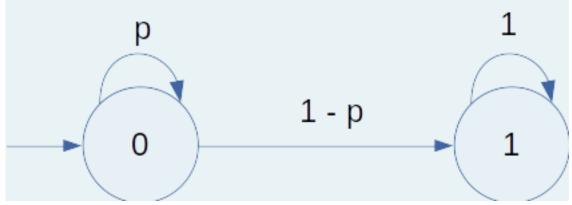
2.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 0, 1-p, 0;p, 1-p, 0, 0;p, 0, 0, 0, 0] 0. 0. 1];Integer Real r1024:Integer state1024[Modelica,Math,Random,Generators,Xorshift1024star,nState];algorithmwhen initial() thenstate1024 x: Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657. 30020);x F1:r1024 0:elsewhen sample(0,1) then(r1024.state1024) $Modelica.Math.Random.Generators.Xorshift 1024 star.random(pre(state 1024)); if (r1024 <= A[x, F1]) then \ x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2]) then \ x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2]) then \ x := F3; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2] + A[x, F2] + A[x, F3] + A$ A[x, F3]) then x := F3; else x := End; end if; end when; end System;

3.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 1-p, 0, 0;p, 0, 1-p, 0;p, 0, 0, 0, 0, 0]

1-p;0, 0, 0, 1]:Integer x; Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen initial() then state1024 := Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020); x := F1; r1024 := 0;elsewhen sample(0,1) then(r1024,state1024) := Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F3]) then x := F3; else x := End;end if;end when;end System;

Si consideri la seguente Markov Chain:

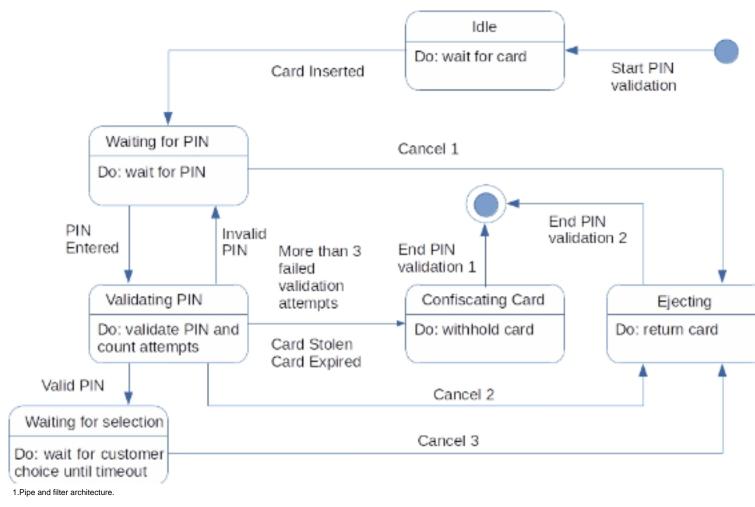
Quale dei seguenti modelli Modelica fornisce un modello ragionevole per la Markov Chain di cui sopra?



2.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [0, 1, 0, 0; p, 0, 1-p, 0; 0, p, 0, 1-p; 0, 0, 0, 0, 1]; Integer x; Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState]; algorithmwhen initial() then state1024 := Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020); x := F1; r1024 := 0;elsewhen sample(0,1) then(r1024,state1024) := Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F3; else x := End;end if;end when;end System;

3.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] =[0, 1, 0, 0;p, 0, 0, 1-p;0, 0, p, 0, 0] r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen thenstate1024 0, 0, 1];Integer Real initial() 1-p;0, X; Modelica.Math.Random.Generators.Xorshift1024star.initialState(614657, 30020);x := F1;r1024 := 0;elsewhen sample(0,1) then(r1024,state1024) Modelica.Math.Random.Generators.Xorshift1024star.random(pre(state1024));if (r1024 <= A[x, F1]) then x := F1; elseif (r1024 <= A[x, F1] + A[x, F2]) then x := F2; elseif (r1024 <= A[x, F1] + A[x, F2] + A[x, F2]) A(x, F3) then x := F3; else x := End:end if:end when:end System:

Quale pattern architetturale meglio descrive l'architettura in figura ?

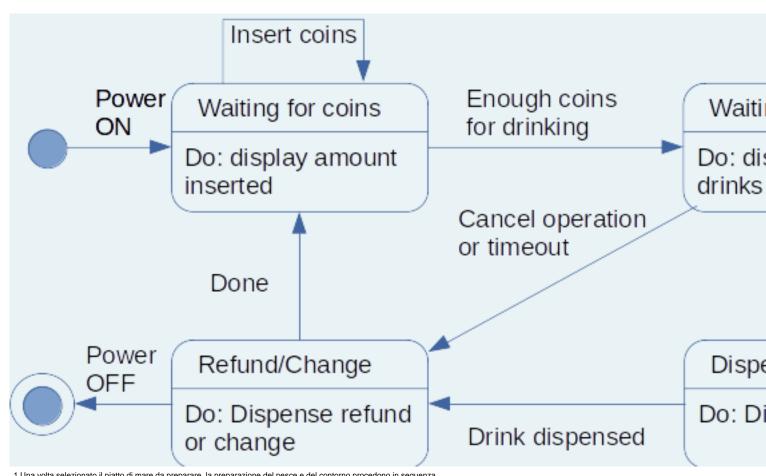


2.Model View Controller.

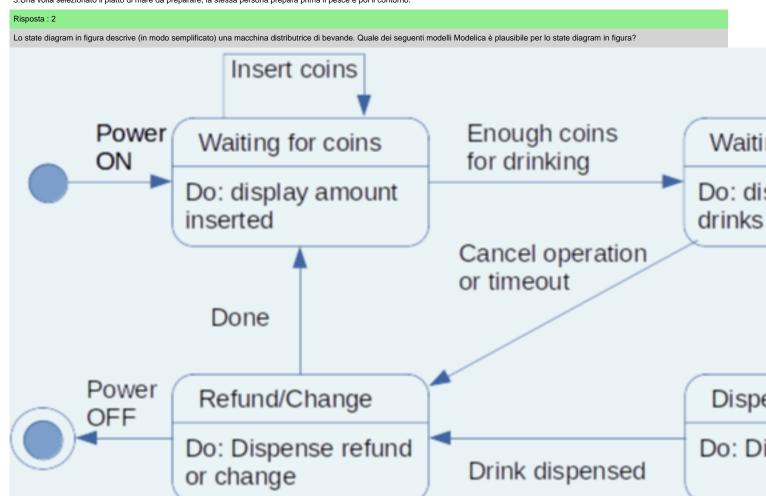
3.Layred architecture.

Risposta: 3

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



- 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



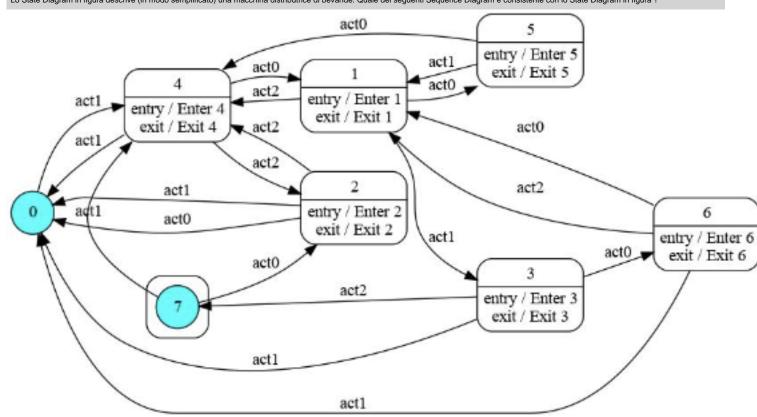
1.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer:/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer := 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 := 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:

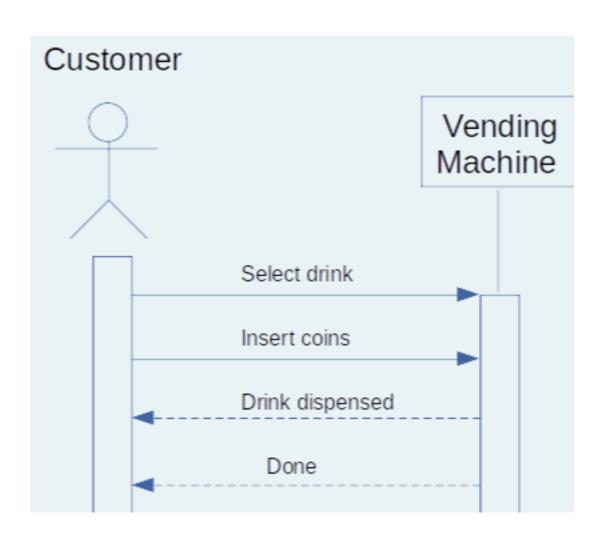
2.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer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

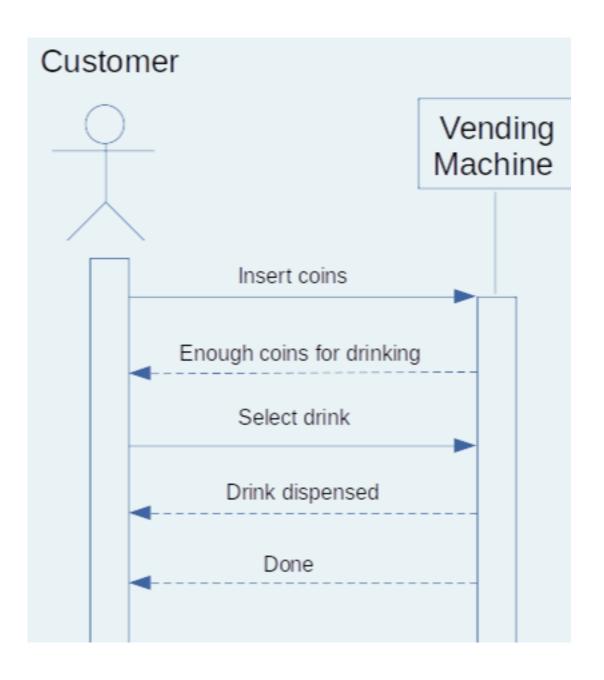
3.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer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

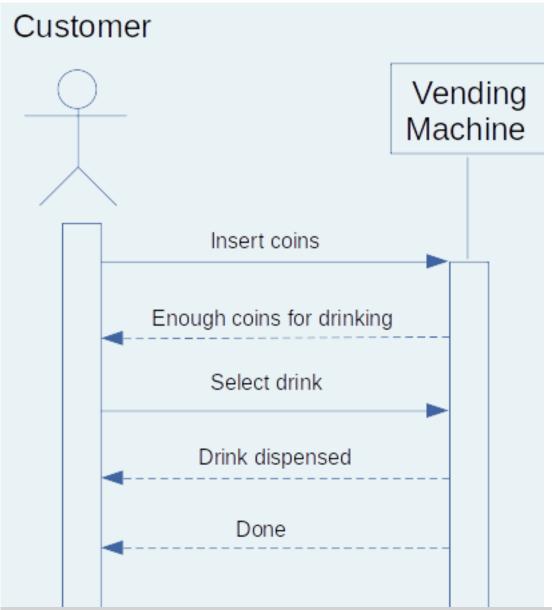
Risposta: 1

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?

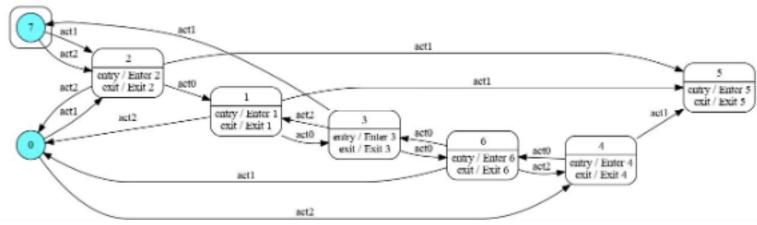








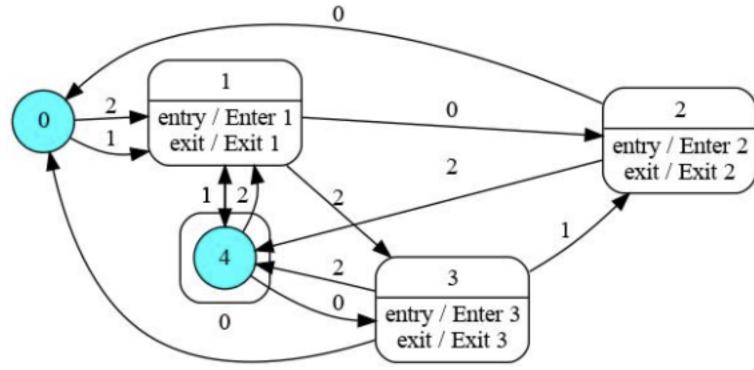
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura?



- 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.

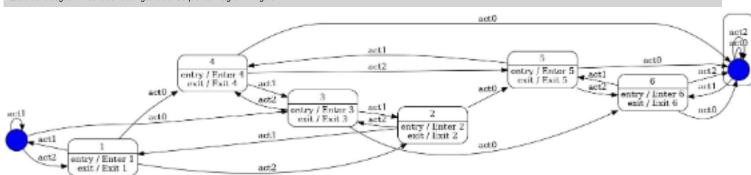
Risposta: 3

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 ?



- 1.La macchina non dà resto.
- 2. Una volta inserite monete per due bevande è possibile ottenerle senza reinserire le monete.
- 3.Una volta selezionata la bevanda non è possibile cancellare l'operazione.

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



- 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.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.
- 3. Periodicamente il client comunica ai pazienti le disponibilità dei medici.

Risposta: