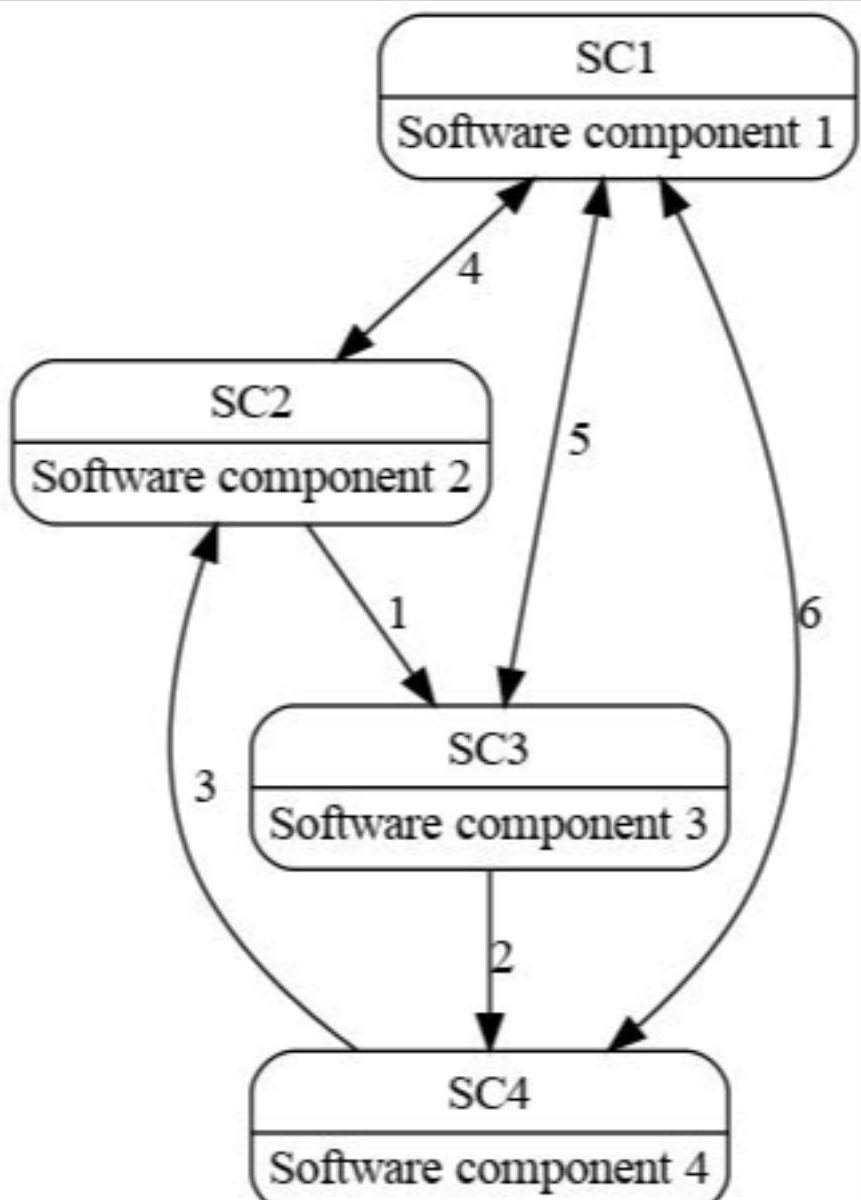


Si consideri la seguente architettura software:

Quale dei seguenti modelli Modelica meglio la rappresenta ?



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, sc3.output1);connect(sc3.input2, sc4.output2);connect(sc4.output3, sc2.input3);end SysArch

2.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, sc3.output1);connect(sc3.input2, sc4.output2);connect(sc4.output3, sc2.output3);end SysArch

3.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.output1, sc3.input1);connect(sc3.output2, sc4.input2);connect(sc4.output3, sc2.input3);end SysArch

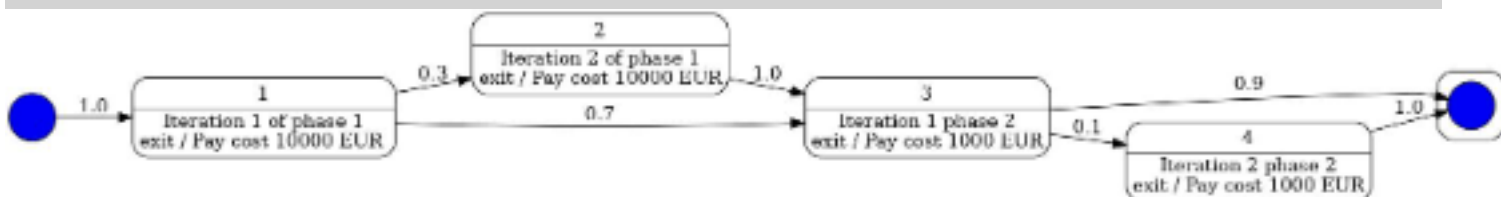
Risposta : 3

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

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

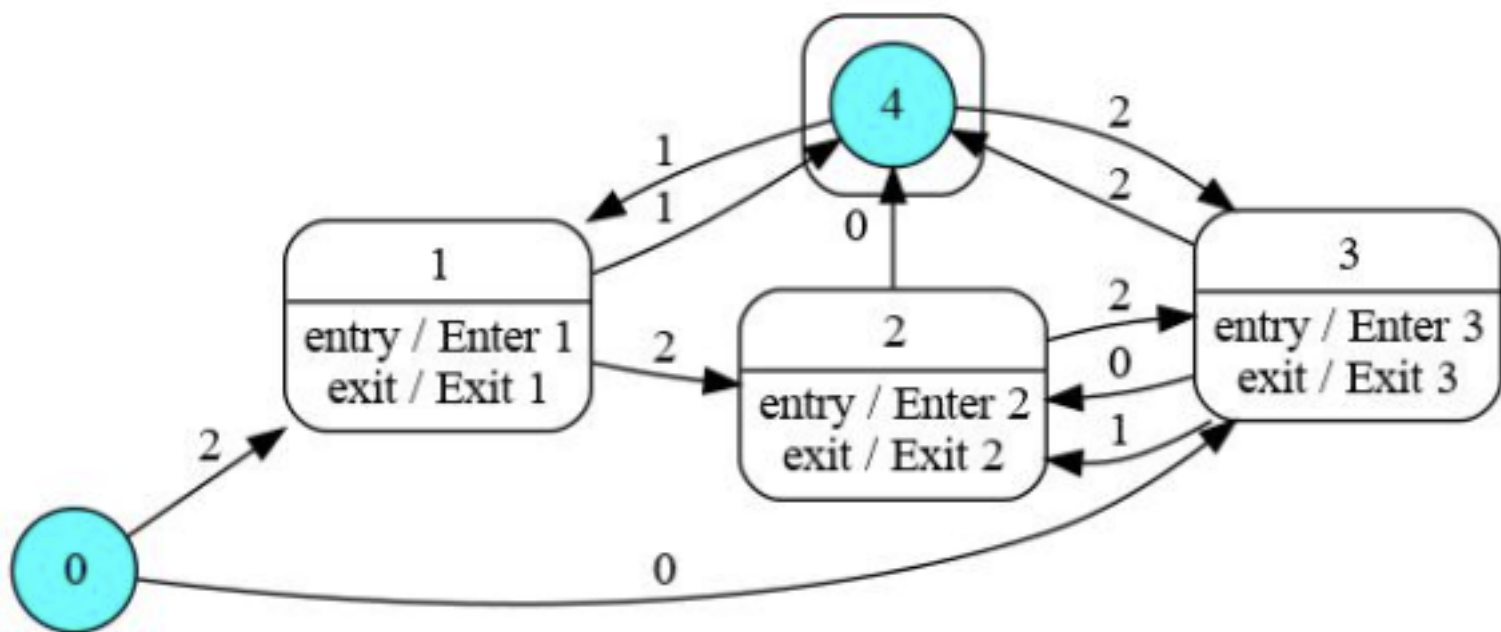
Uno scenario è una sequenza di stati.

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



Risposta : 1

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

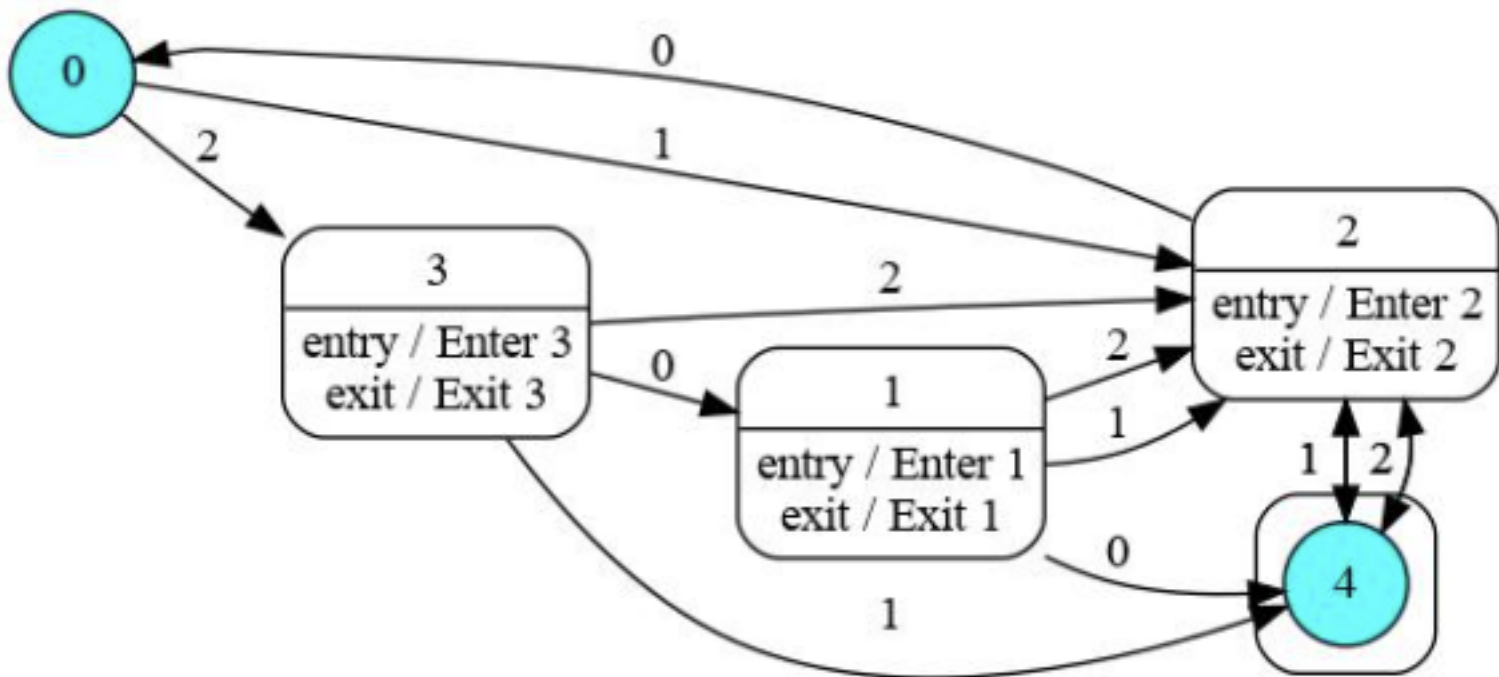


1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() then x := 0;elsewhen sample(0,T) then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

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

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

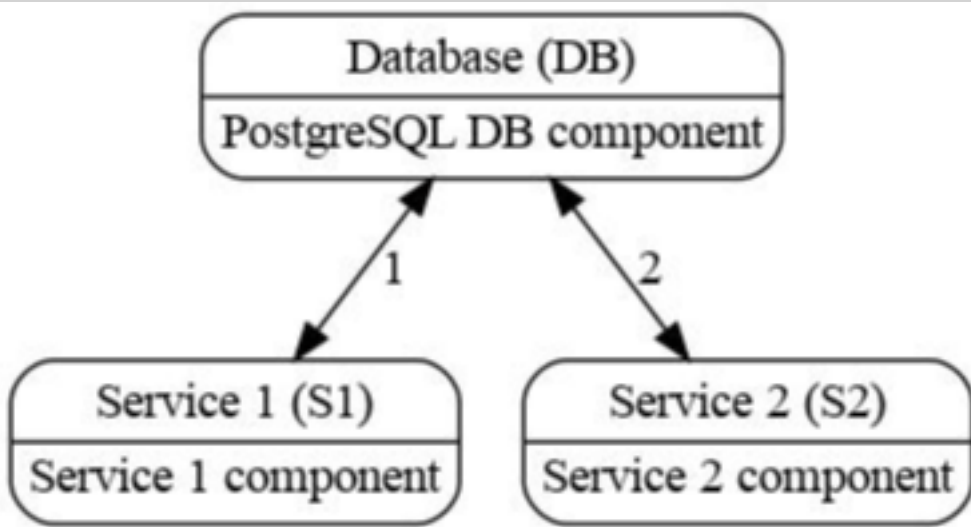


1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() then x := 0;elsewhen sample(0,T) then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

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

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

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



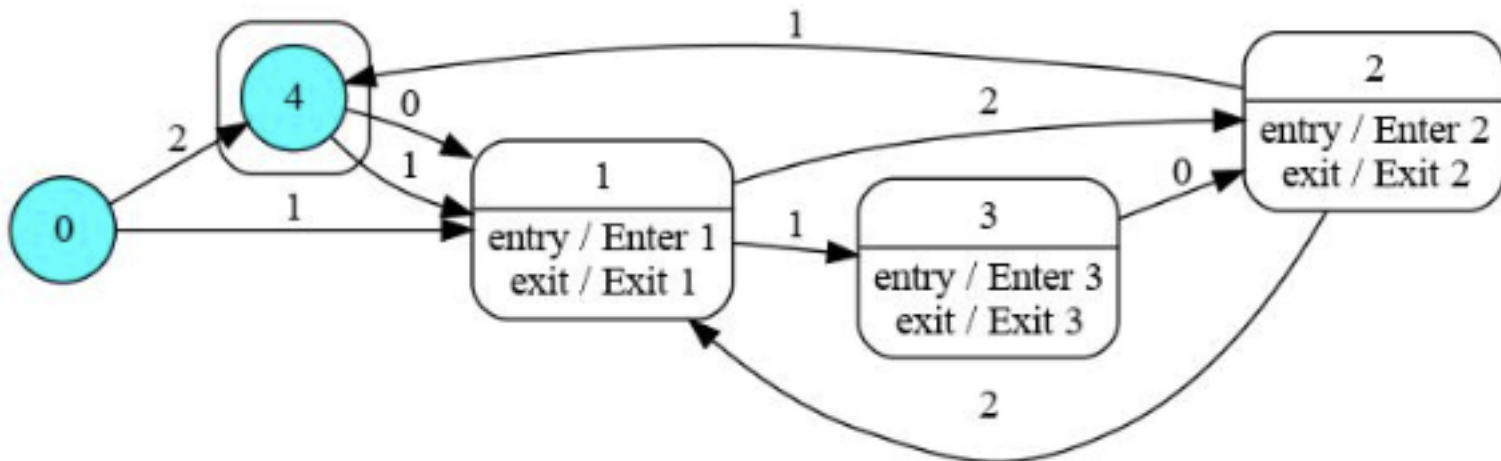
1.block SysArchDB db_c;S1 s1_c;S2 s2_c;connect(db_c.input[1], s1_c.output[1]);connect(db_c.output[1], s1_c.input[1]);connect(s1_c.input[2], s2_c.output[2]);connect(s1_c.output[2], s2_c.input[2]);end SysArch

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

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

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

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

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

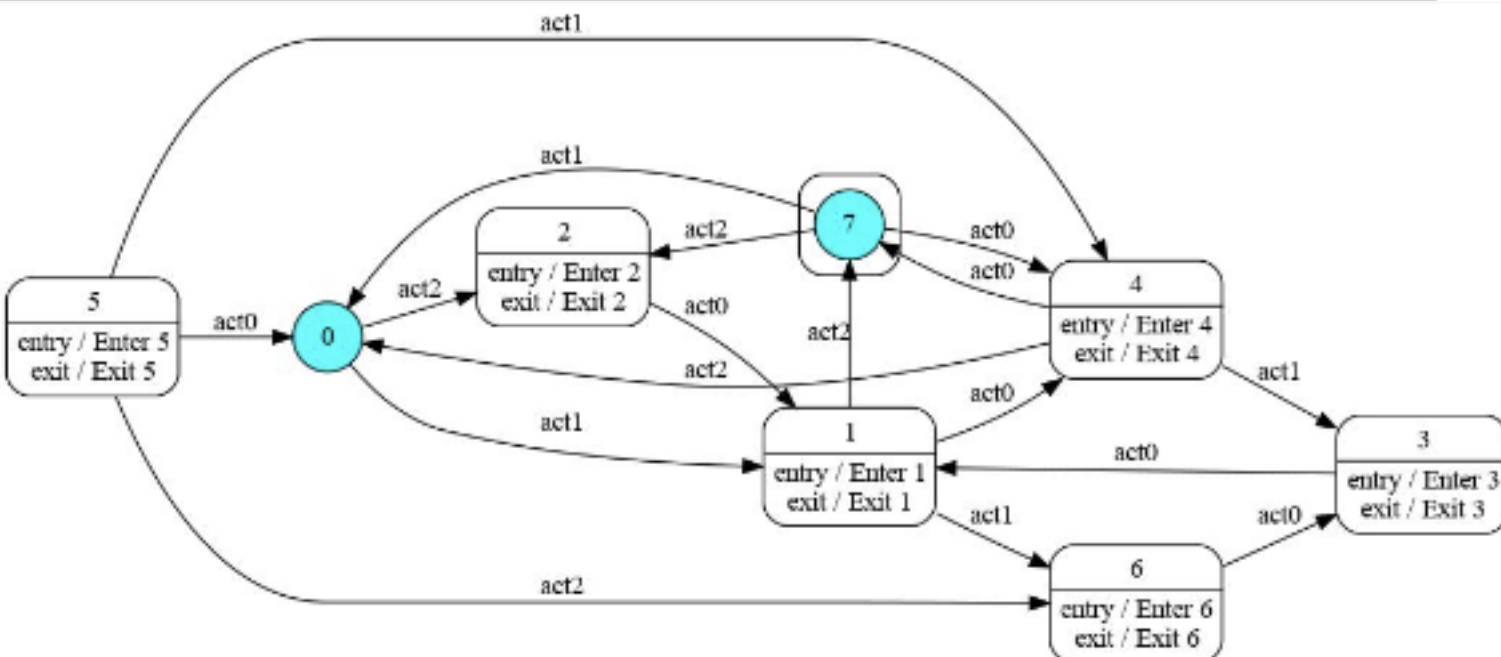
Si consideri lo state diagram in figura

Si consideri il seguente insieme di test cases:

Test case 1: act1 act2

Test case 2: act2 act0 act1 act0 act0

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%

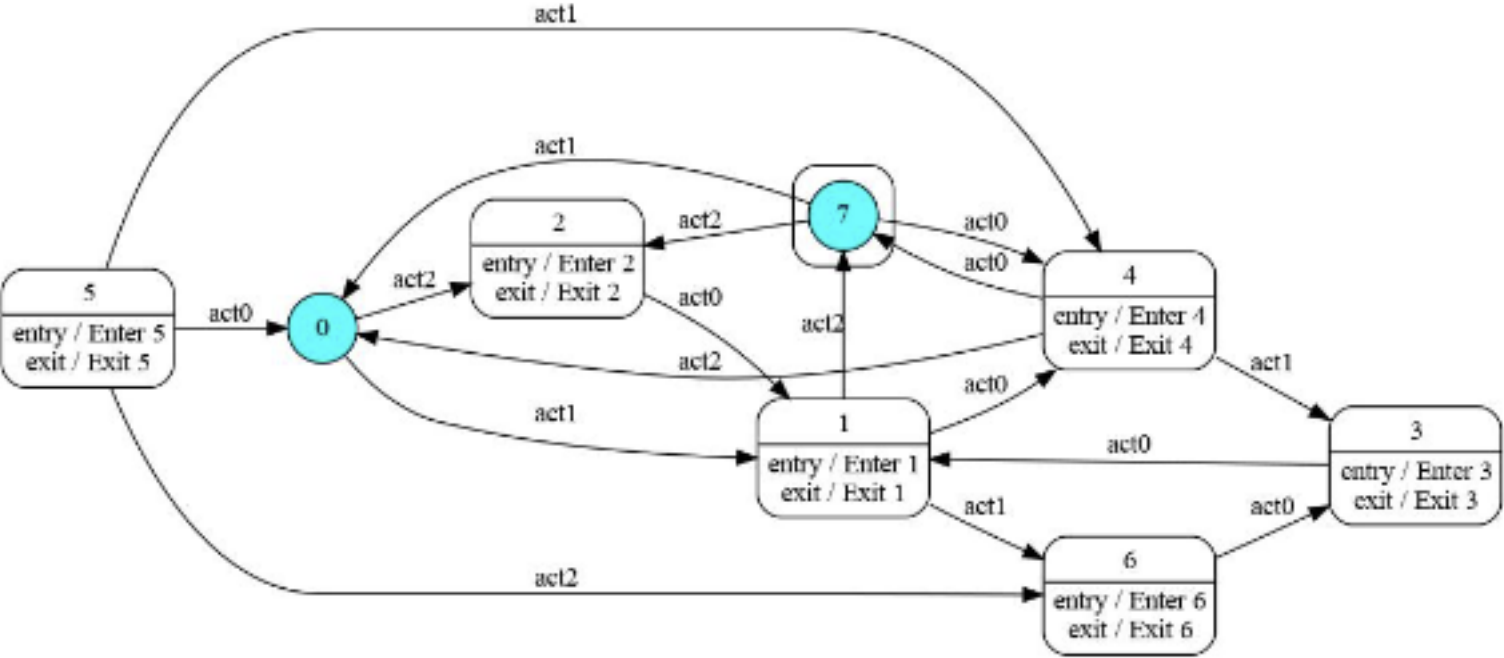
Risposta : 1

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

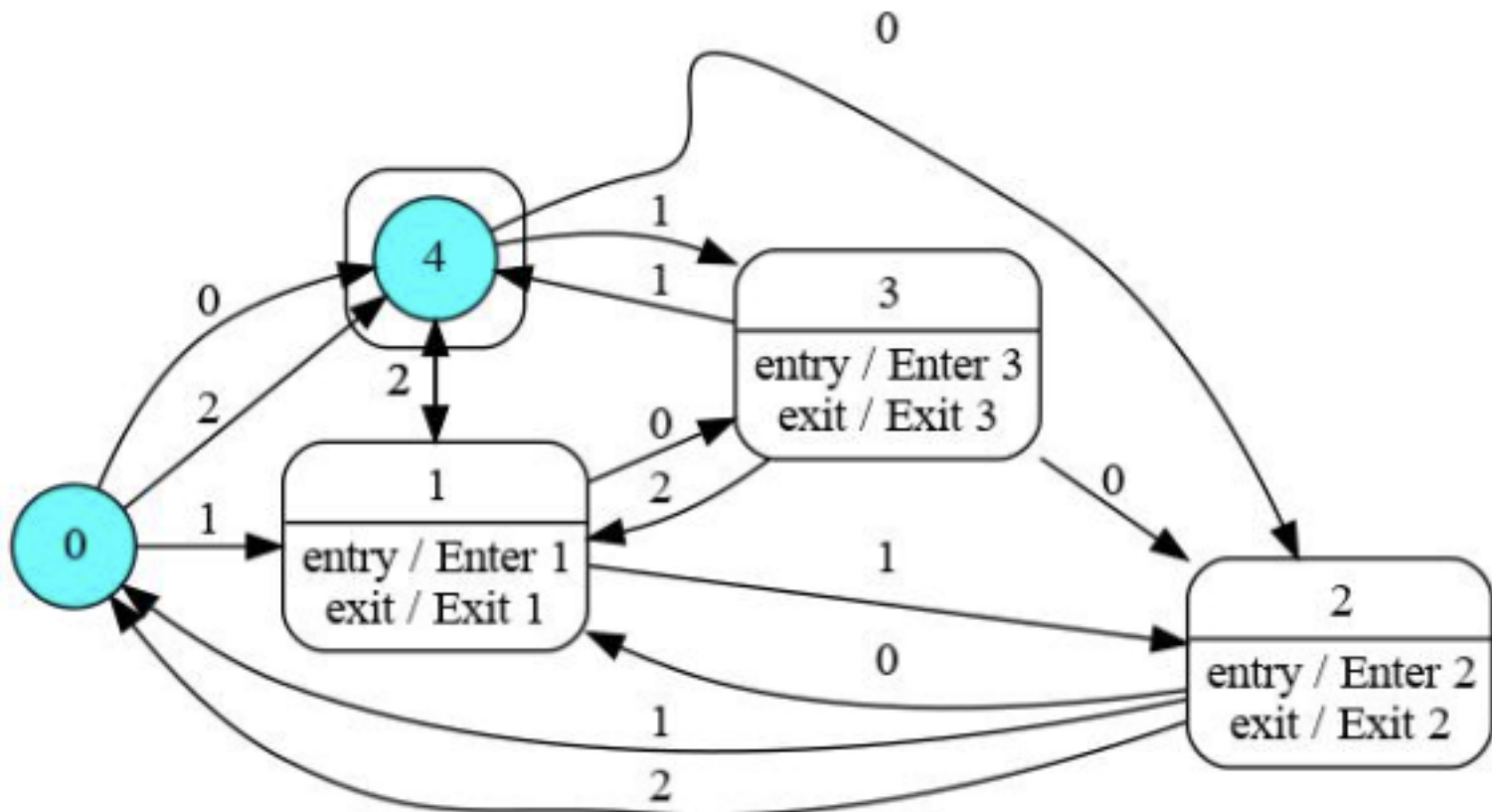
Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra?



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

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

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

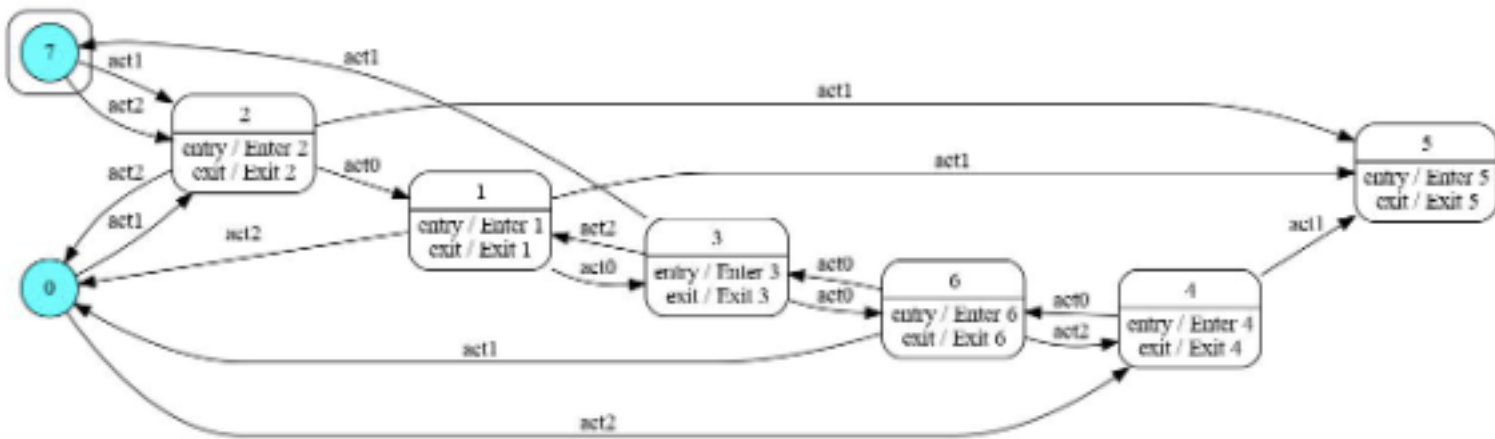
ed il seguente insieme di test cases:

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%

Risposta : 2

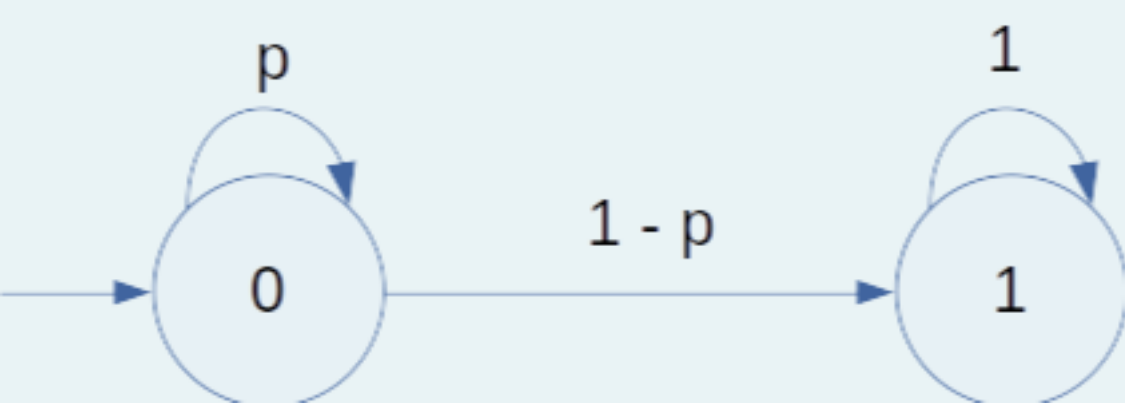
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il tempo necessario per completare la fase x è $time(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappresenta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

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.

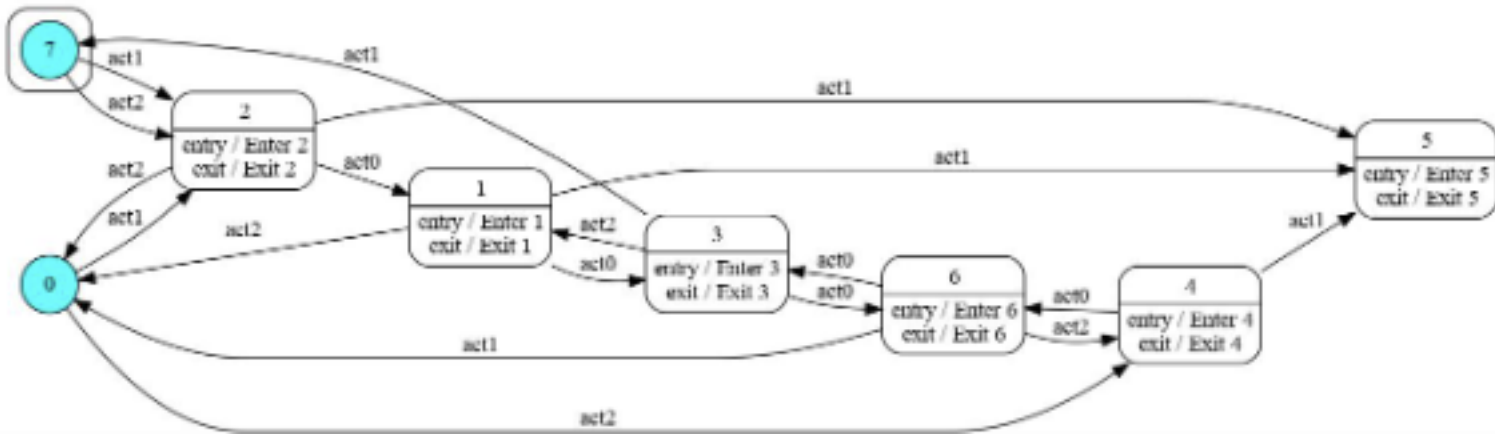
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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%

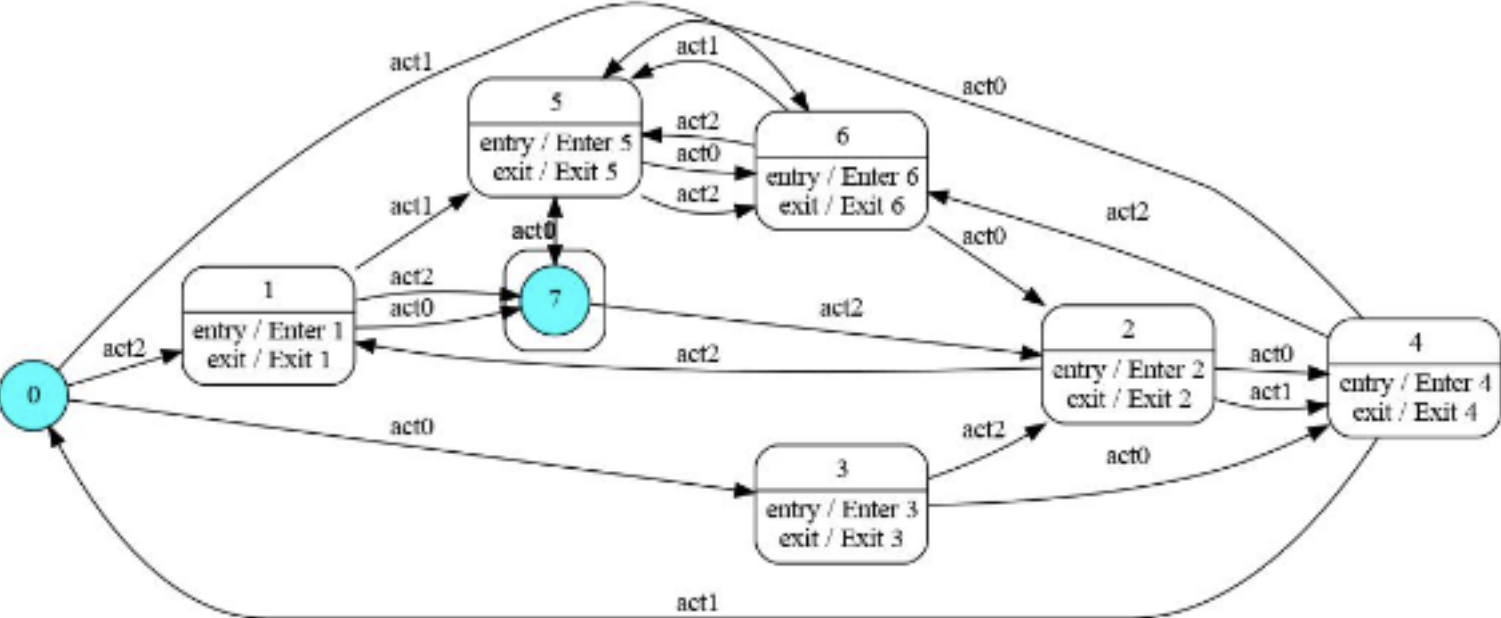
Risposta : 3

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 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%

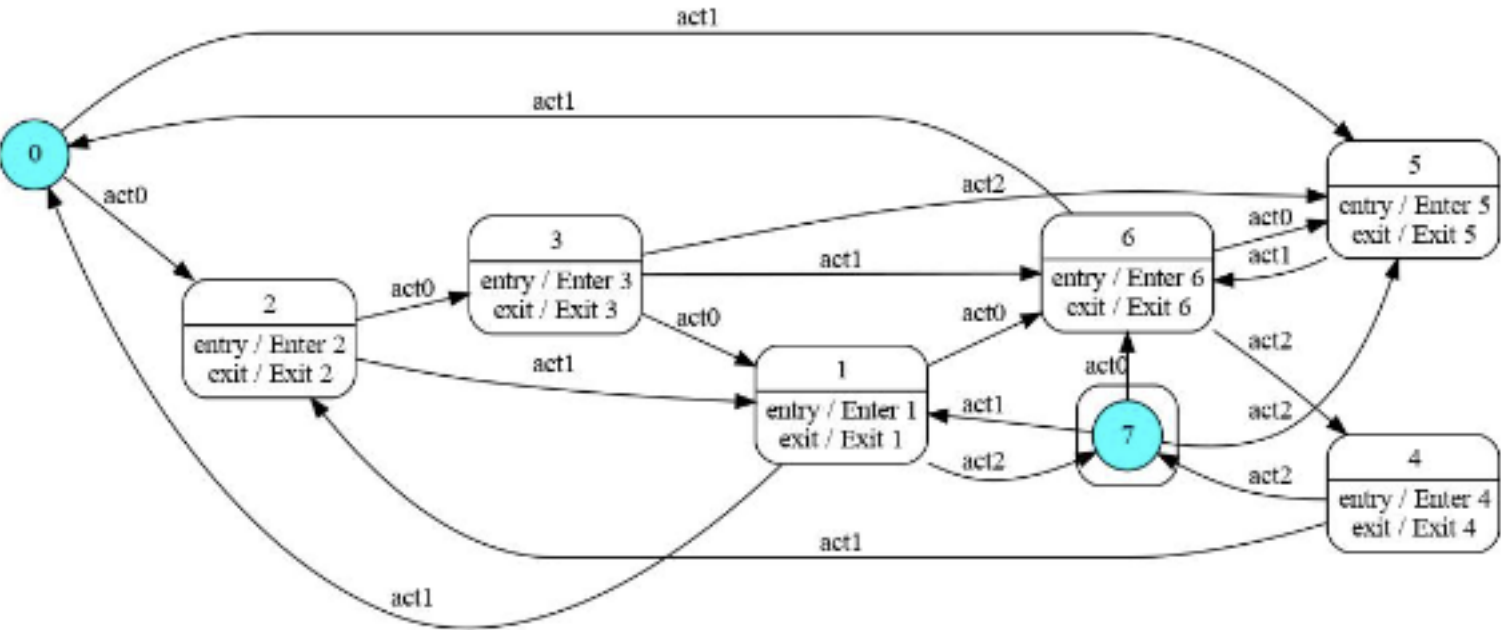
Risposta : 1

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%

Risposta : 3

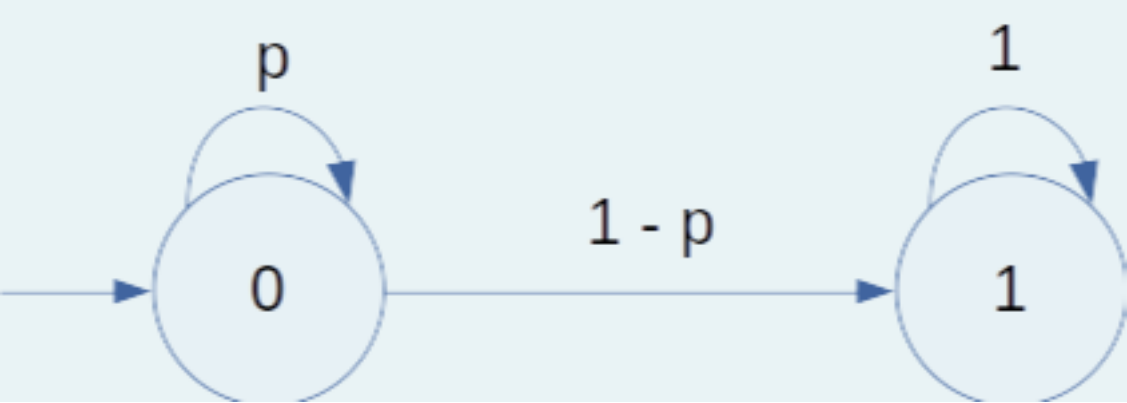
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il costo dello stato (fase) x è c(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappresenta il completamento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

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

Quindi il costo C(X) della sequenza di stati X = x(0), x(1), x(2), 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)/(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

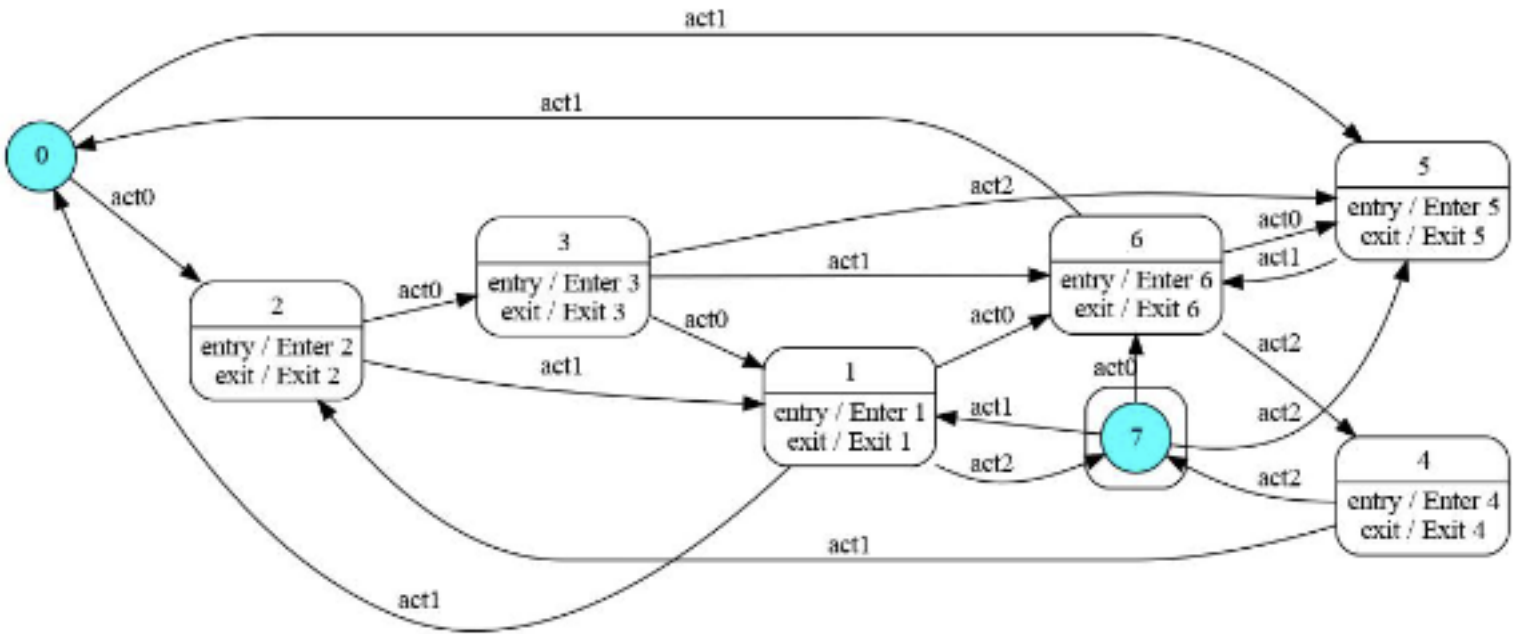
Si consideri il seguente insieme di test cases:

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

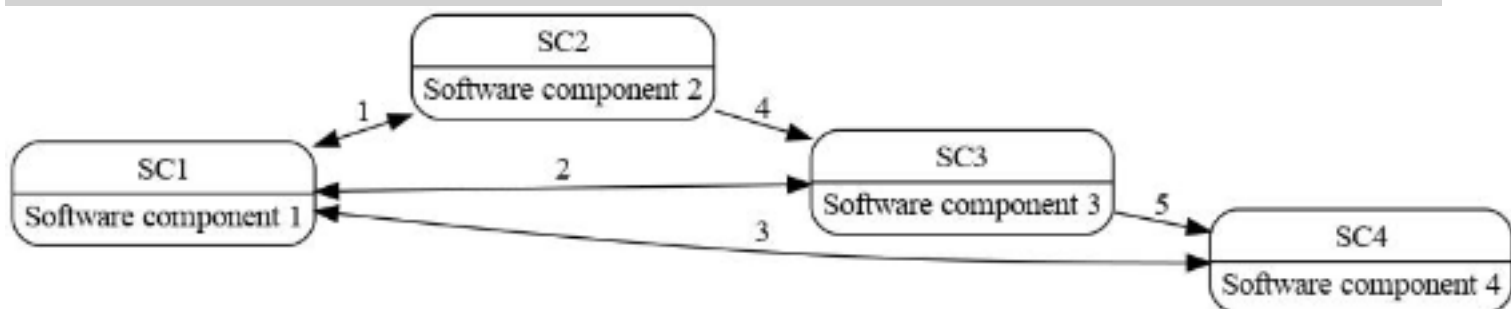


- 1.State coverage: 70%
- 2.State coverage: 80%
- 3.State coverage: 90%

Risposta : 3

Si consideri la seguente architettura software:

Quale dei seguenti modelli Modelica meglio la rappresenta ?



1.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.input4, sc3.output4);connect(sc3.input5, sc4.output5);end SysArch

2.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.input4, sc3.output4);connect(sc2.output4, sc3.input4);connect(sc3.input5, sc4.output5);connect(sc3.output5, 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.

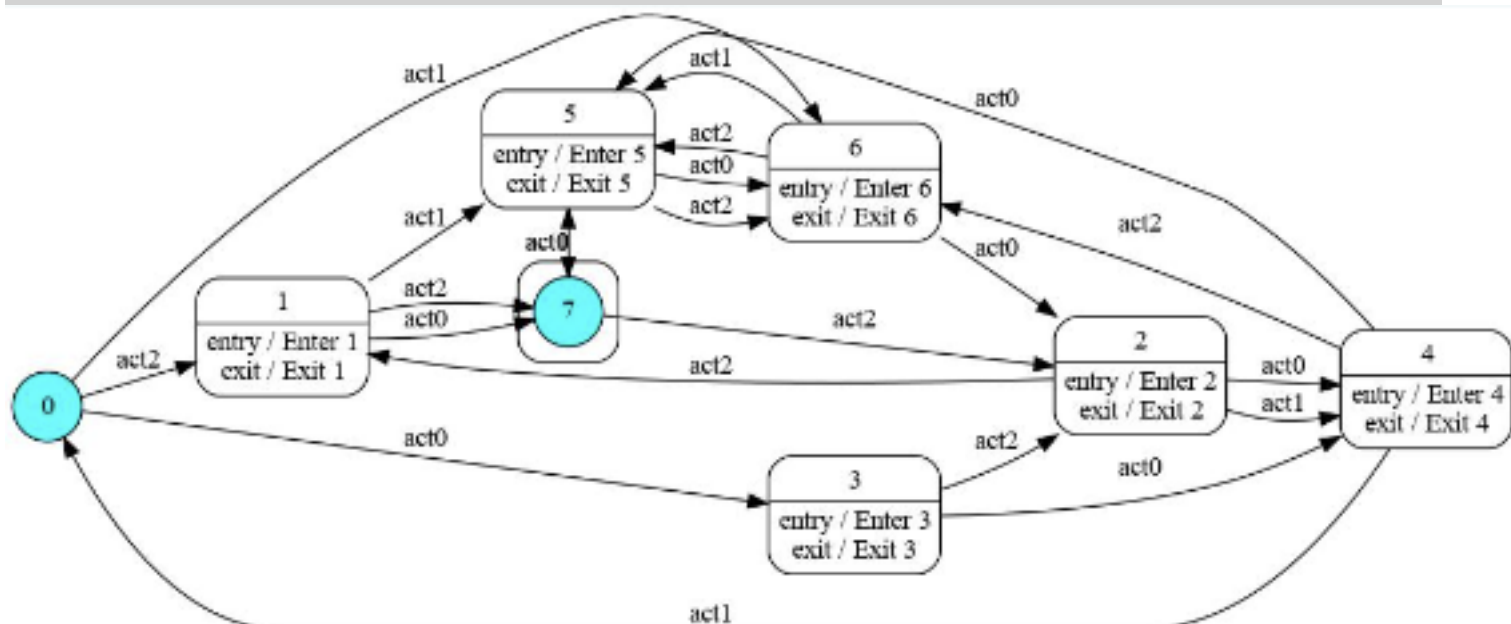
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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?



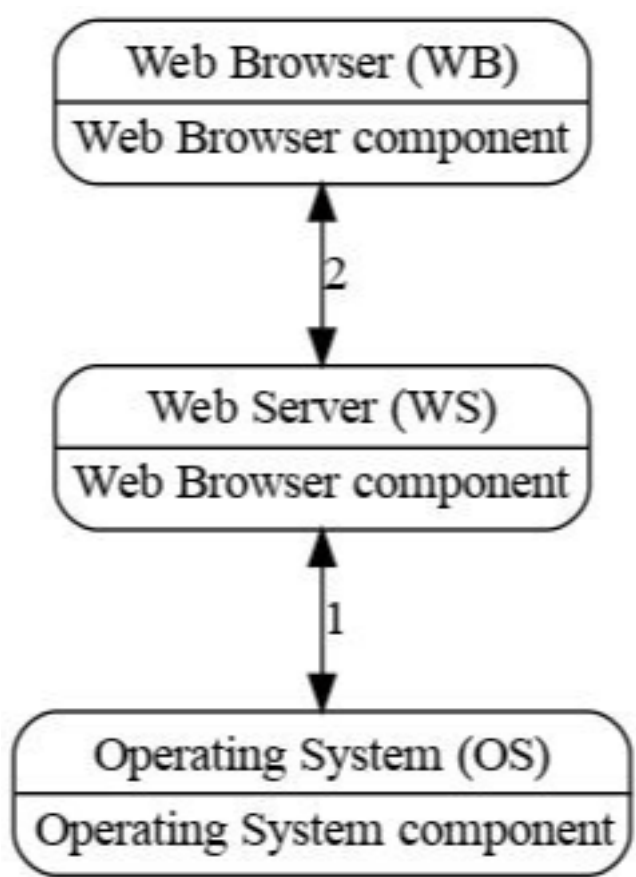
1.Transition coverage: 55%

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.output1, ws_c.input1);connect(wb_c.input2, ws_c.output2);connect(wb_c.output2, ws_c.input2);end
SysArch

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

3.block SysArchOS os_c;WS ws_c;WB wb_c;connect(os_c.input1, wb_c.output1);connect(os_c.output1, wb_c.input1);connect(wb_c.input2, ws_c.output2);connect(wb_c.output2, ws_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 <= 2)

        {return ;}

        else {y = y + 1; goto LOOP;}

} /* f() */

int main() { int i, y; int x[N];

// define test cases

    x[0] = 3;

// testing

    for (i = 0; i < N; i++) {

        y = f(x[i]); // function under testing

        assert(y == (abs(x[i]) <= 2) ? 0 : (abs(x[i]) - 2)); // oracle

    }

    printf("All %d test cases passed\n", N);

    return (0);

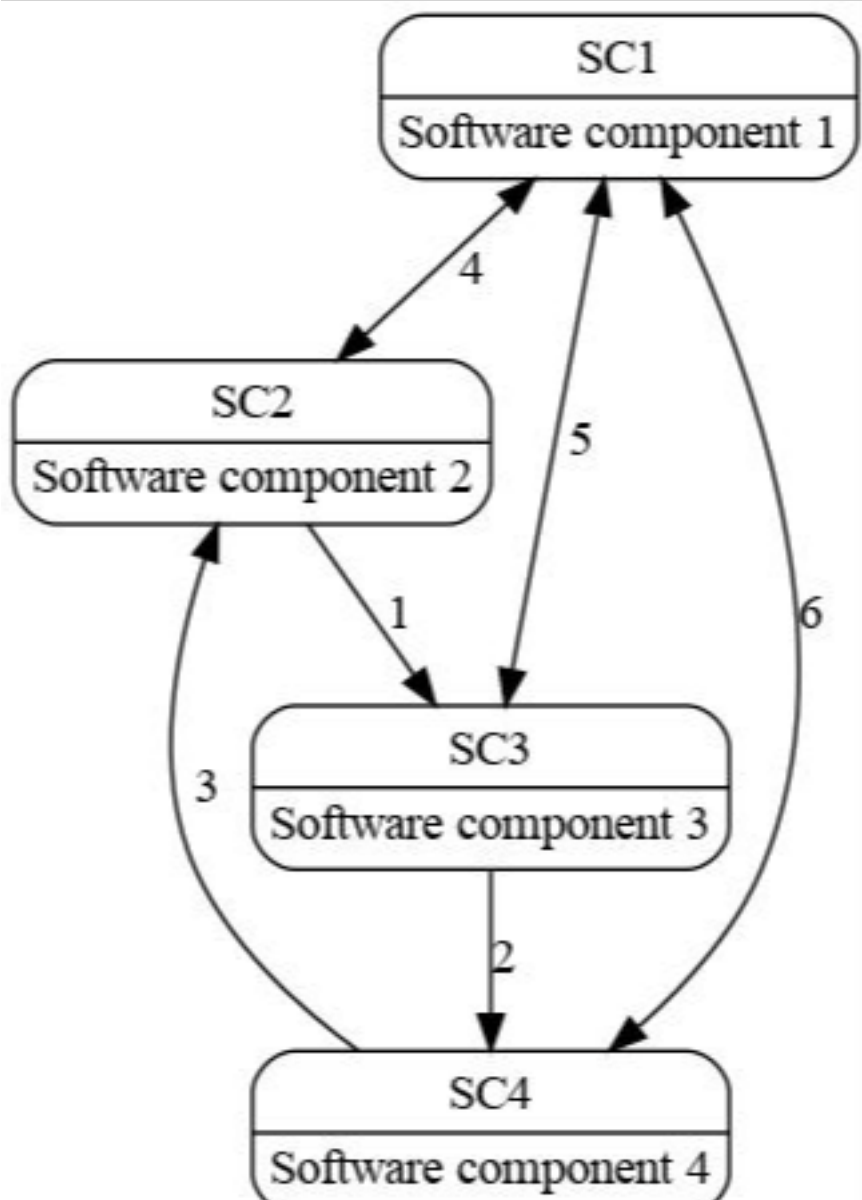
}

-----

```

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

Quale delle seguenti è la branch coverage conseguita?



1.80%

2.50%

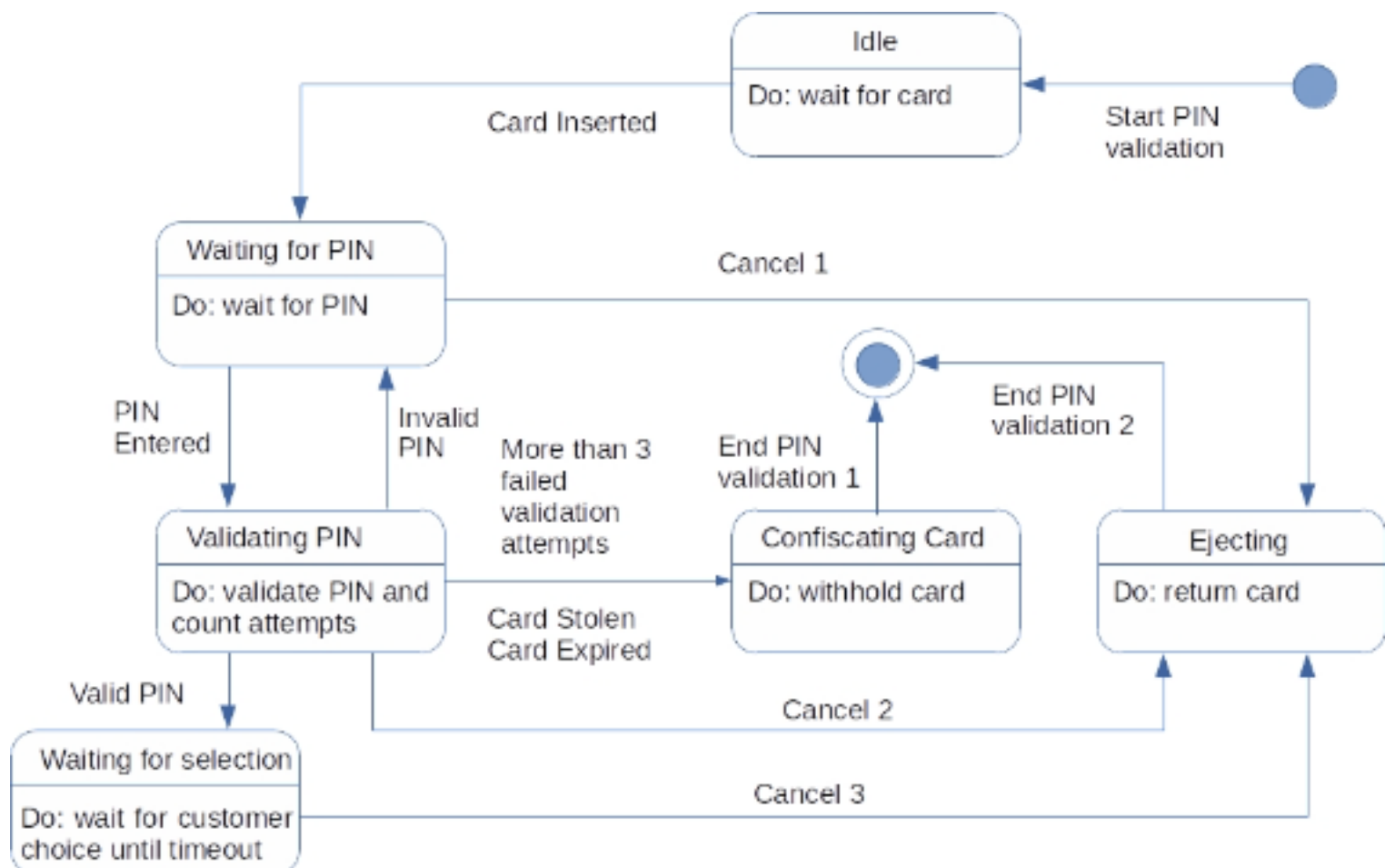
3.100%

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, 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



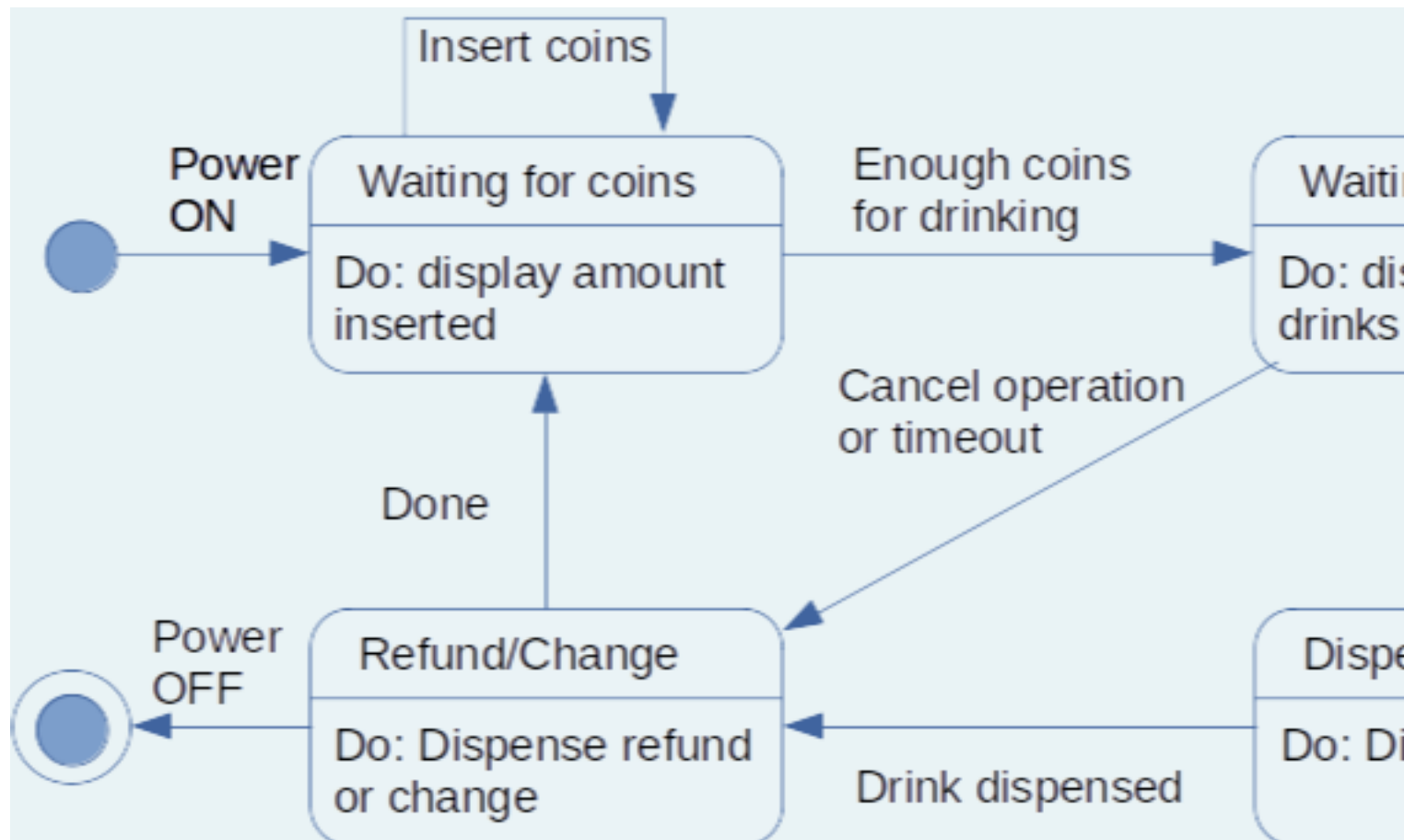
1.80%

2.60%

3.40%

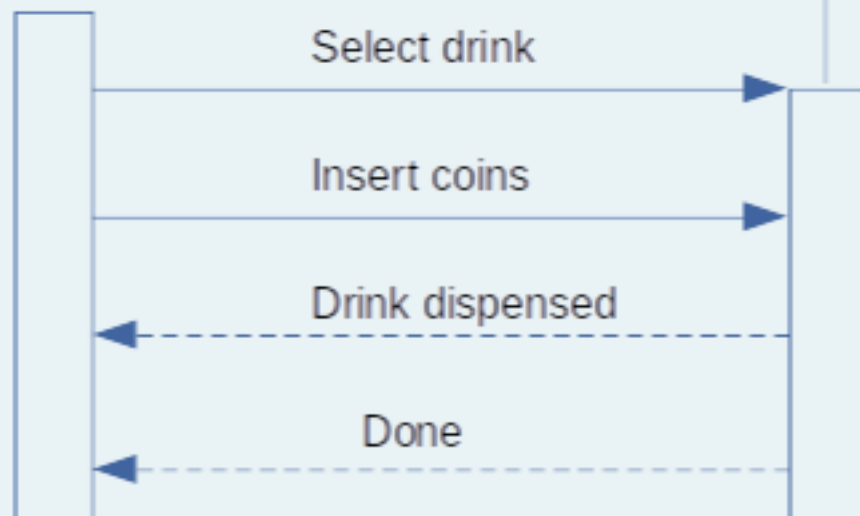
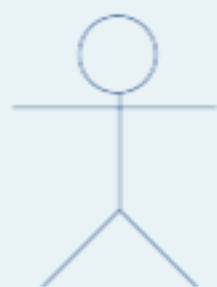
Risposta : 2

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 ?



Customer

Vending Machine



Customer

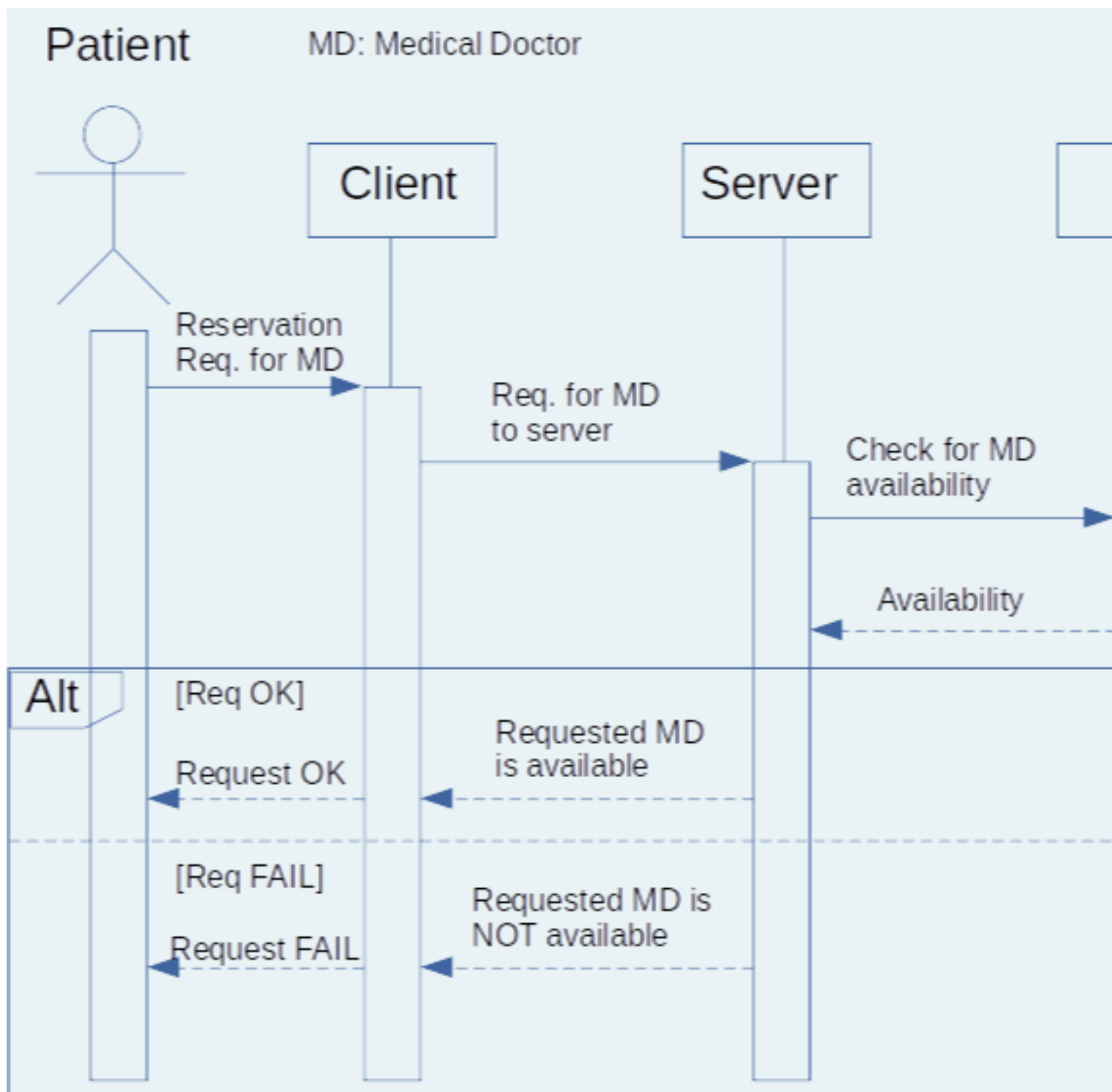


Vending Machine



IVBORw0KGgoAAAANSUHEUgAAAZcAAAACCAYAAACUpFJAAAAKEIEQVRliiWN8+fnr4ZRMpGwSgYBaOAioBpoB0wCkbBKBgFo2D4AQCKdAPUYeTFMQAAAABJRU5ErkJggg==

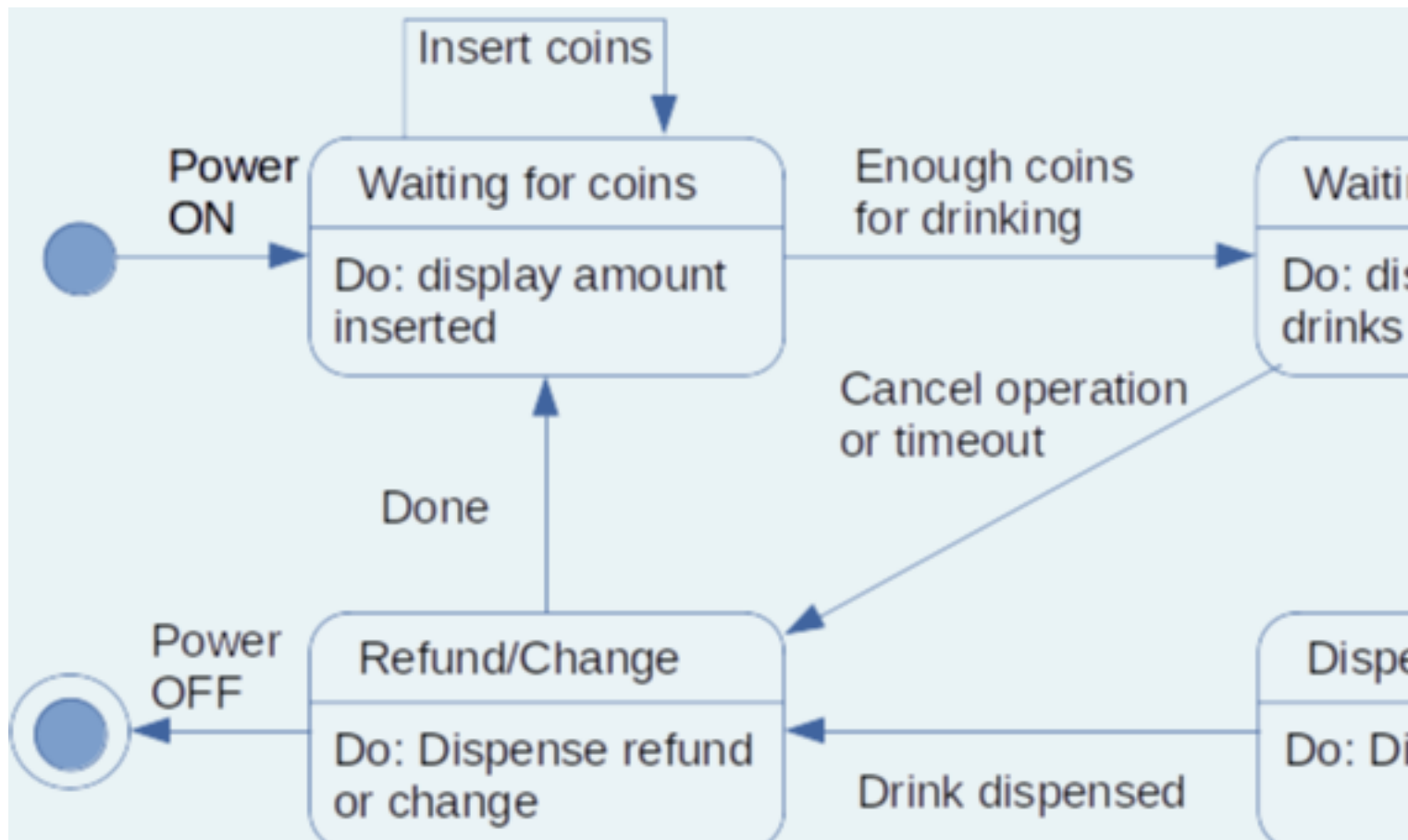
Quale delle seguenti frasi è corretta riguardo al Sequence Diagram in figura ?



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.

Risposta : 3

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

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

```

Risposta : 3

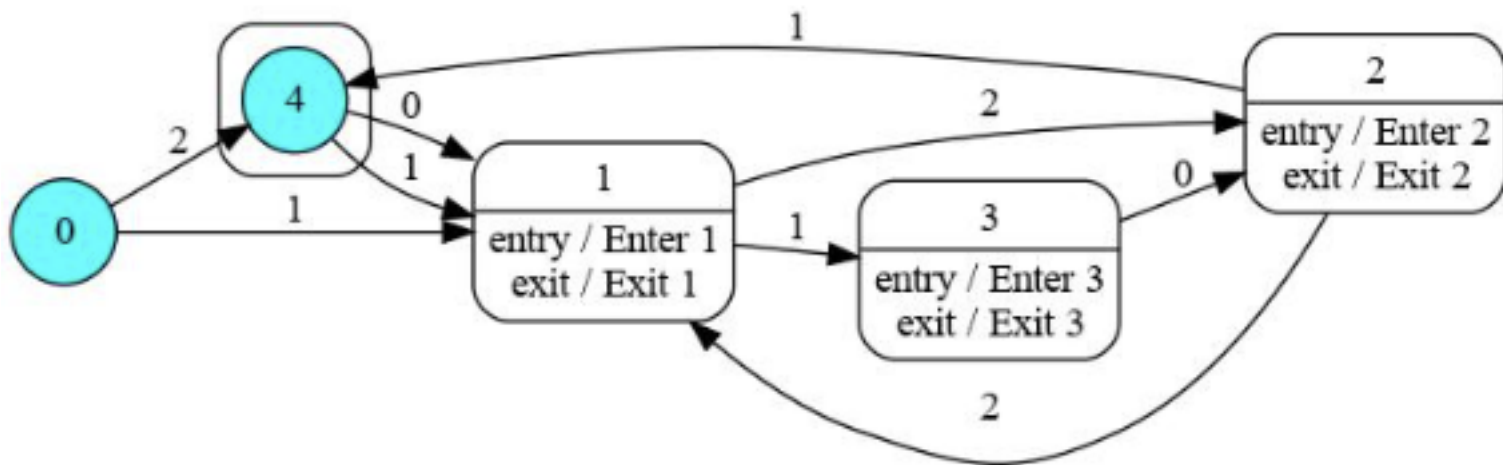
La state coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) rggunti almeno una volta.

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

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

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

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



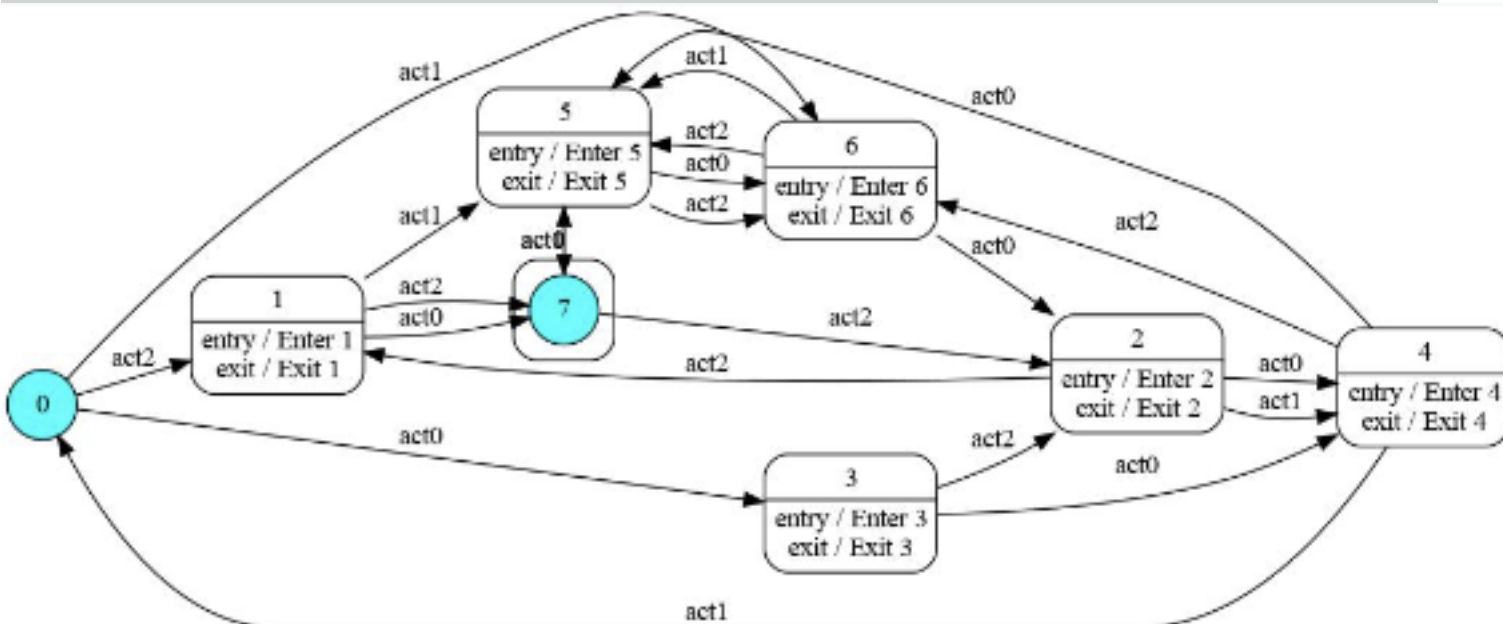
1.80%

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.

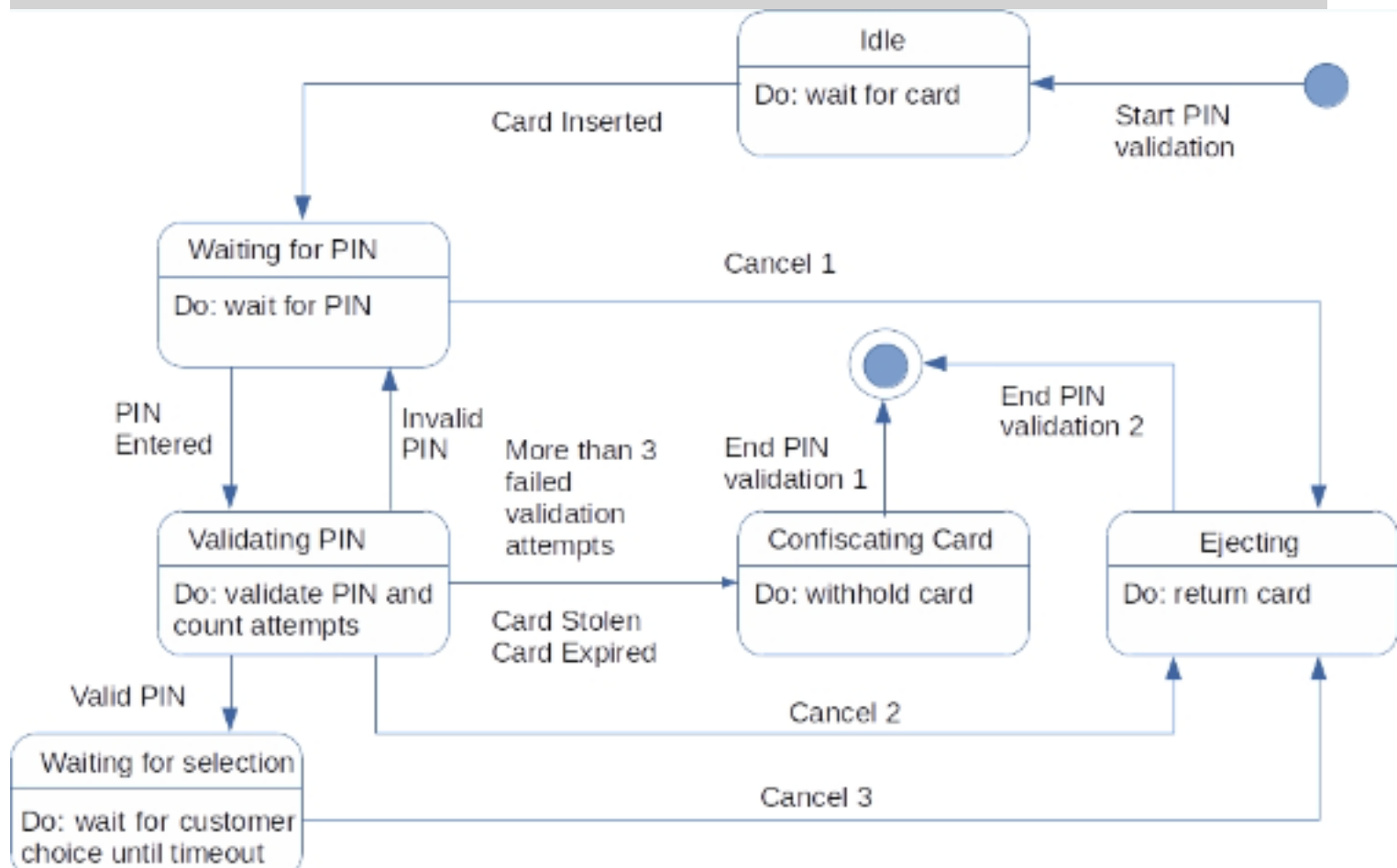
Risposta : 2

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) rggunti 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%

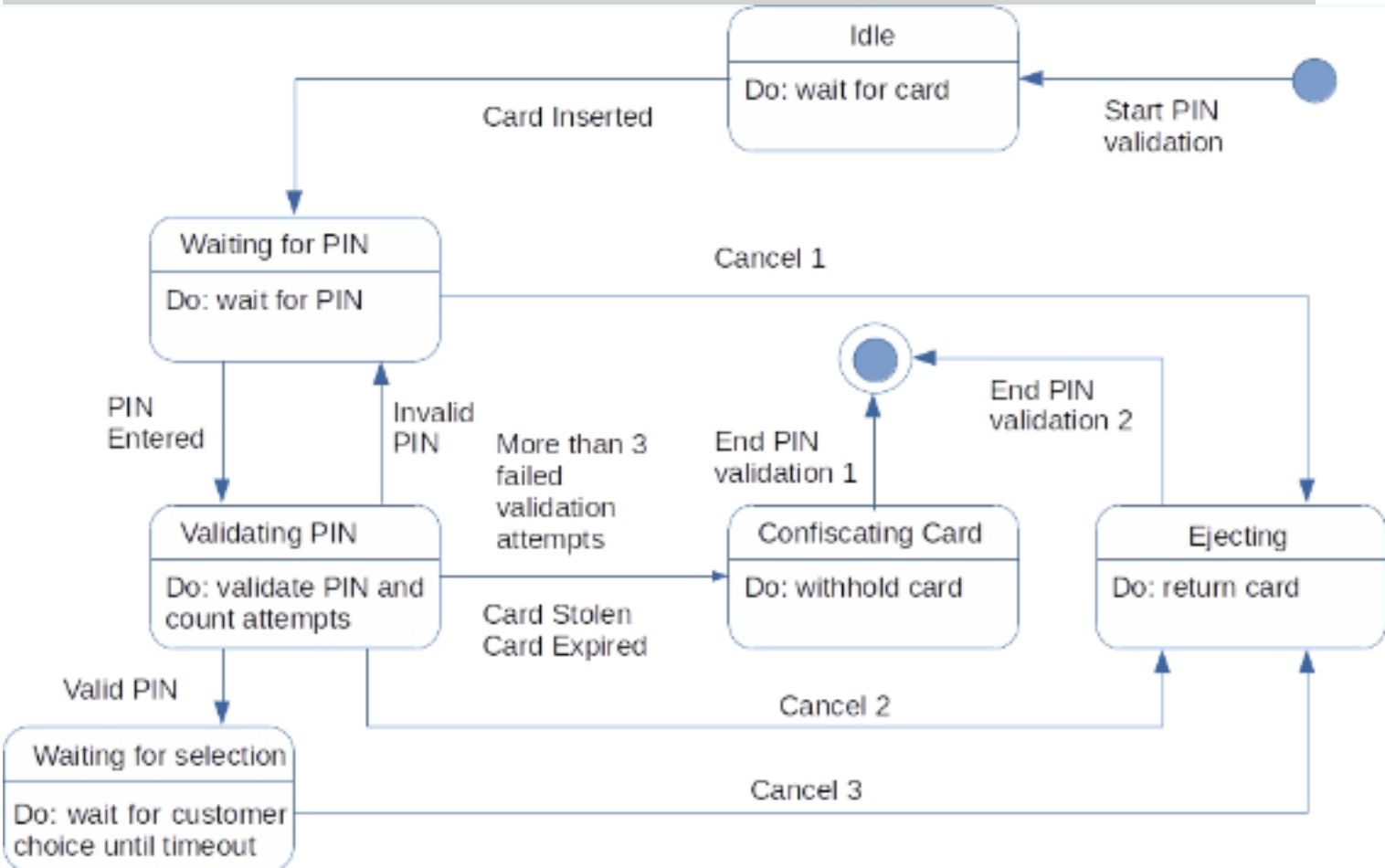
3.100%

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, Cancel 2, End PIN Validation 2



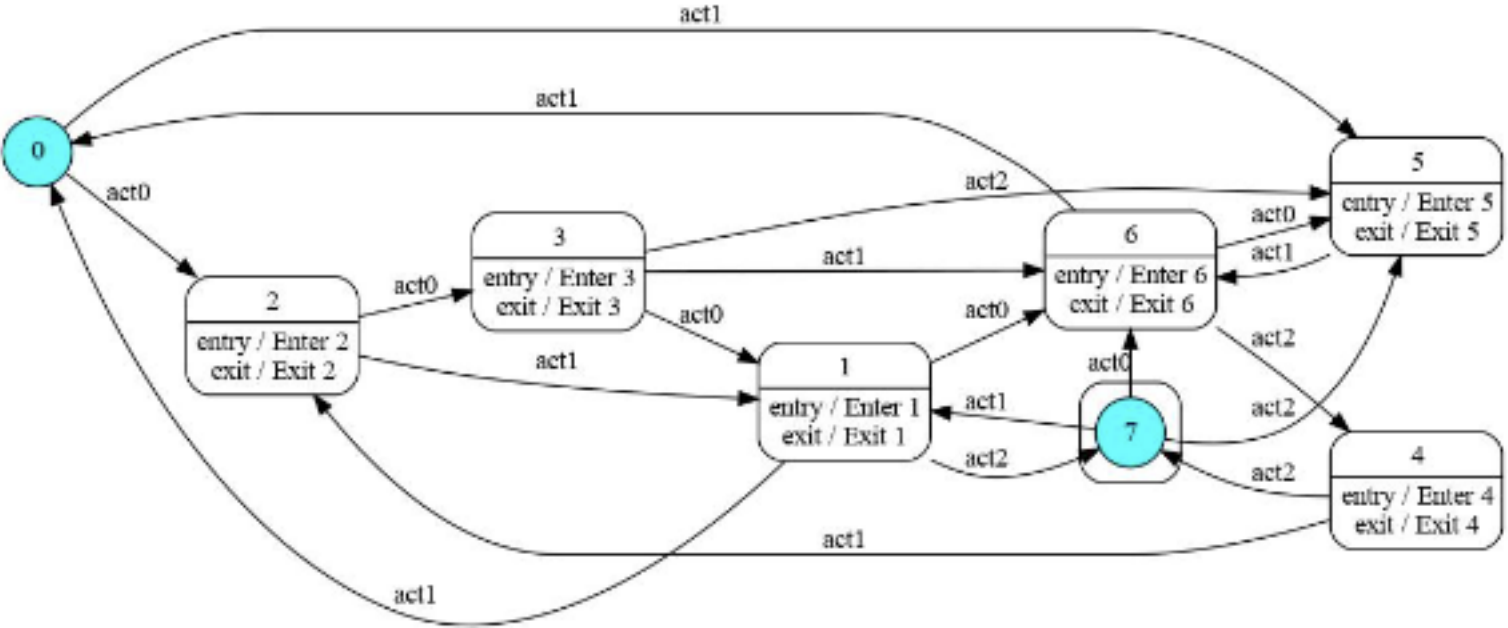
1.80%

2.60%

3.40%

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 ?



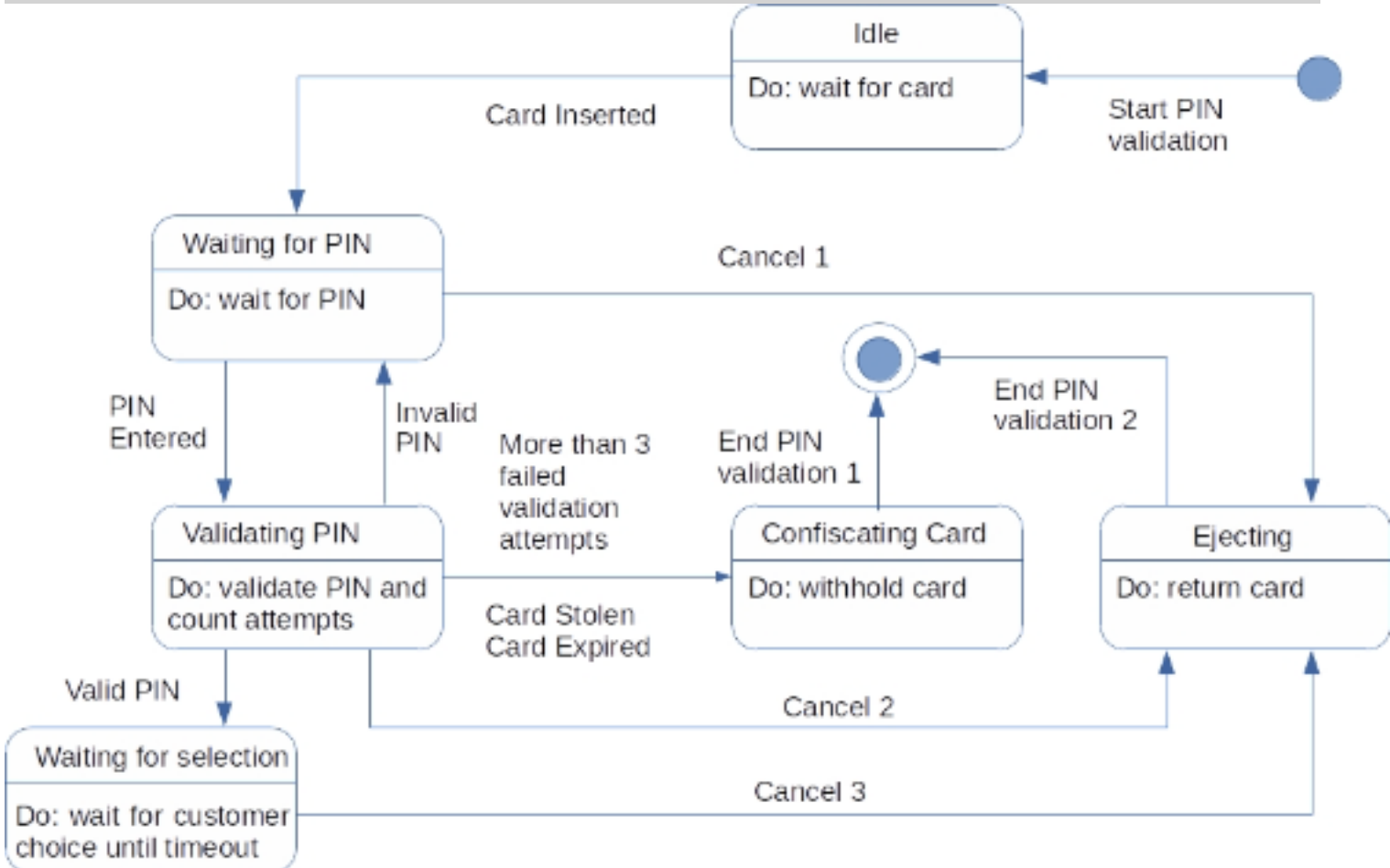
- 1.Una volta inserite monete per due bevande è possibile ottenerle senza reinserire le monete.
- 2.Una volta selezionata la bevanda non è possibile cancellare l'operazione.
- 3.La macchina non dà resto.

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:

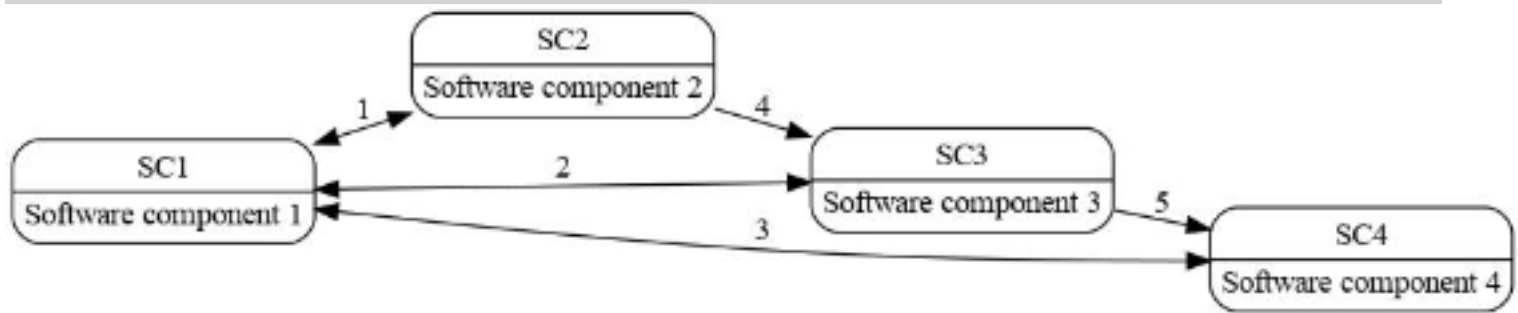
- 1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2
- 2) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 2, End PIN Validation 2
- 3) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, PIN Entered,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



- 1.100%
- 2.80%
- 3.90%

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 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

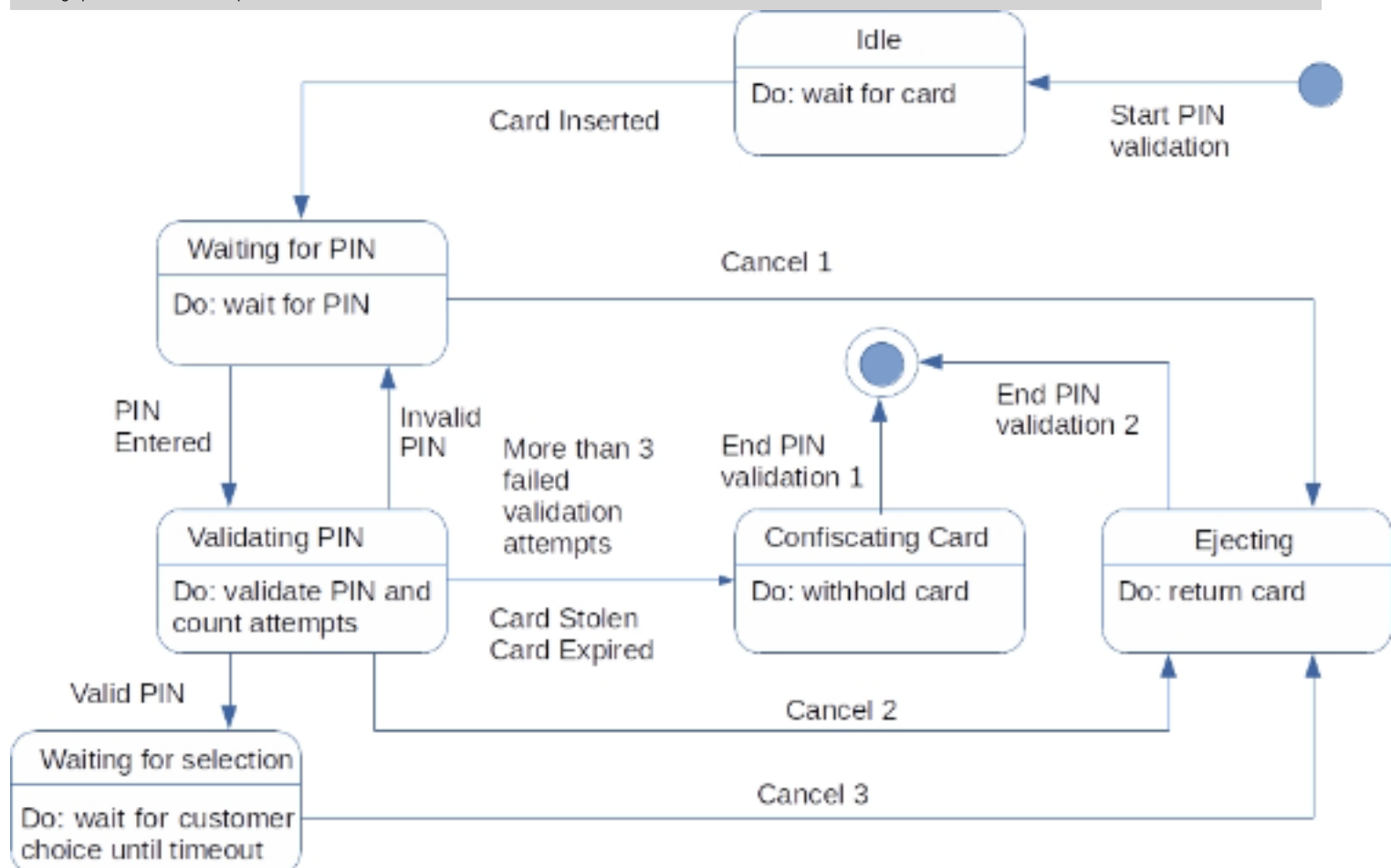
La state coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) rggiunti almeno una volta.

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, 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



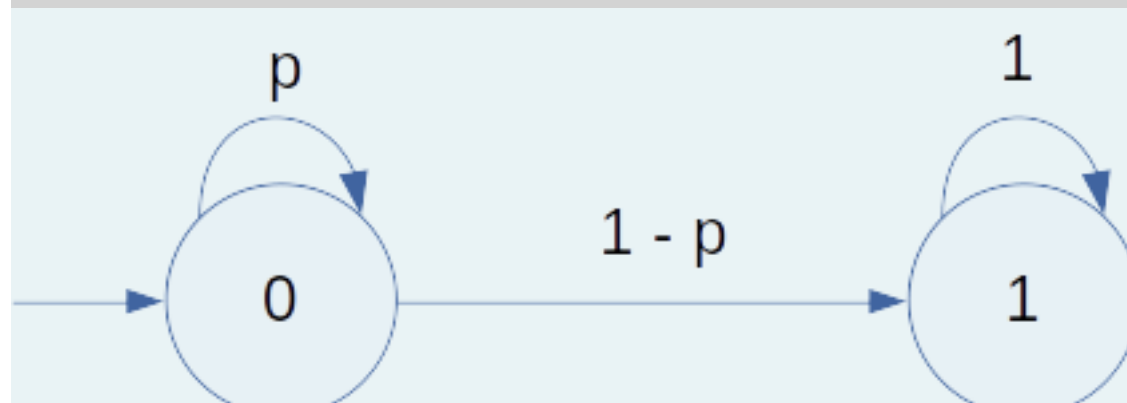
1.80%

2.60%

3.90%

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. $1/(p \cdot (1 - p))$

2. $1/(1 - p)$

3. $(1 - p)/p$

Risposta : 2

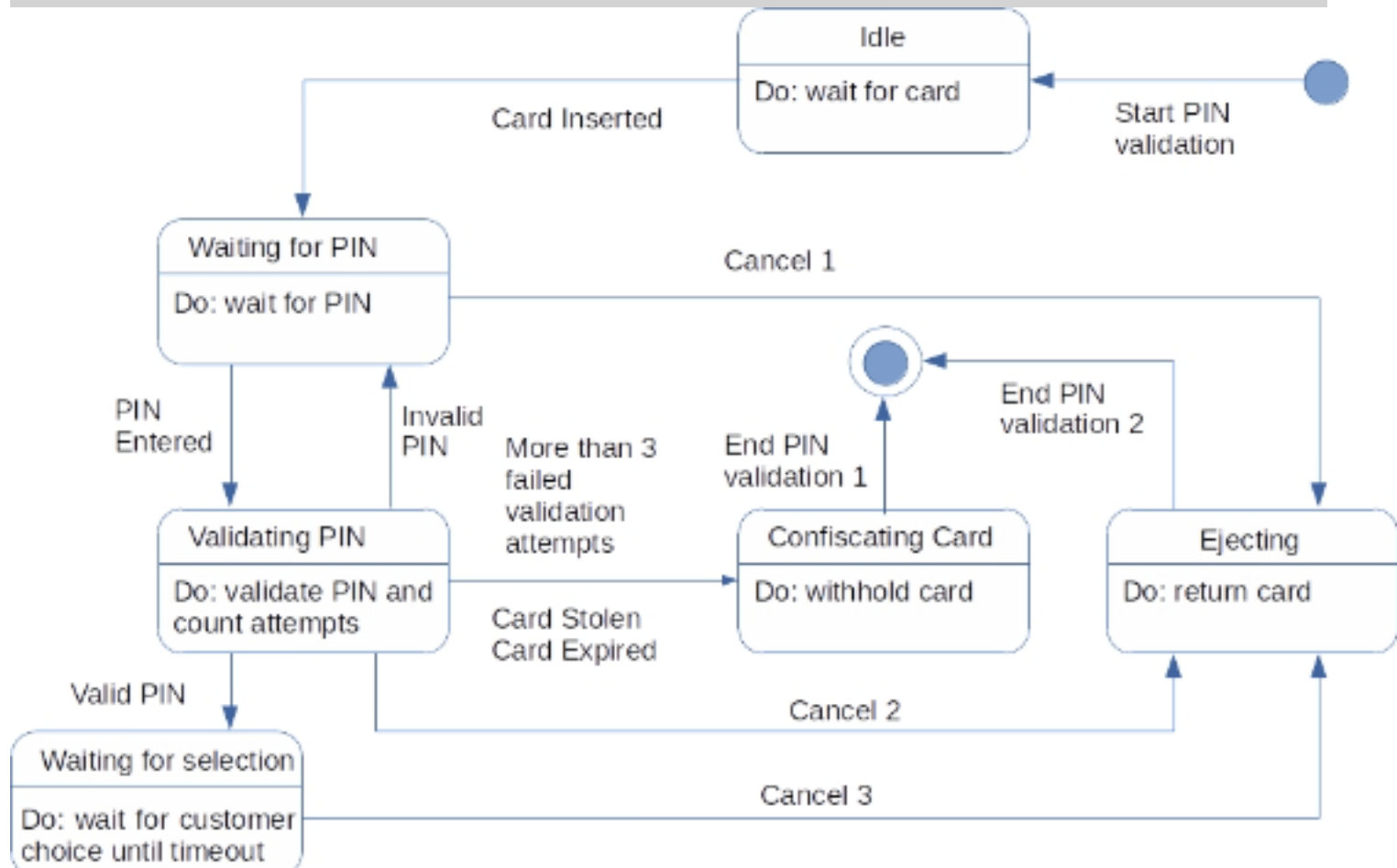
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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

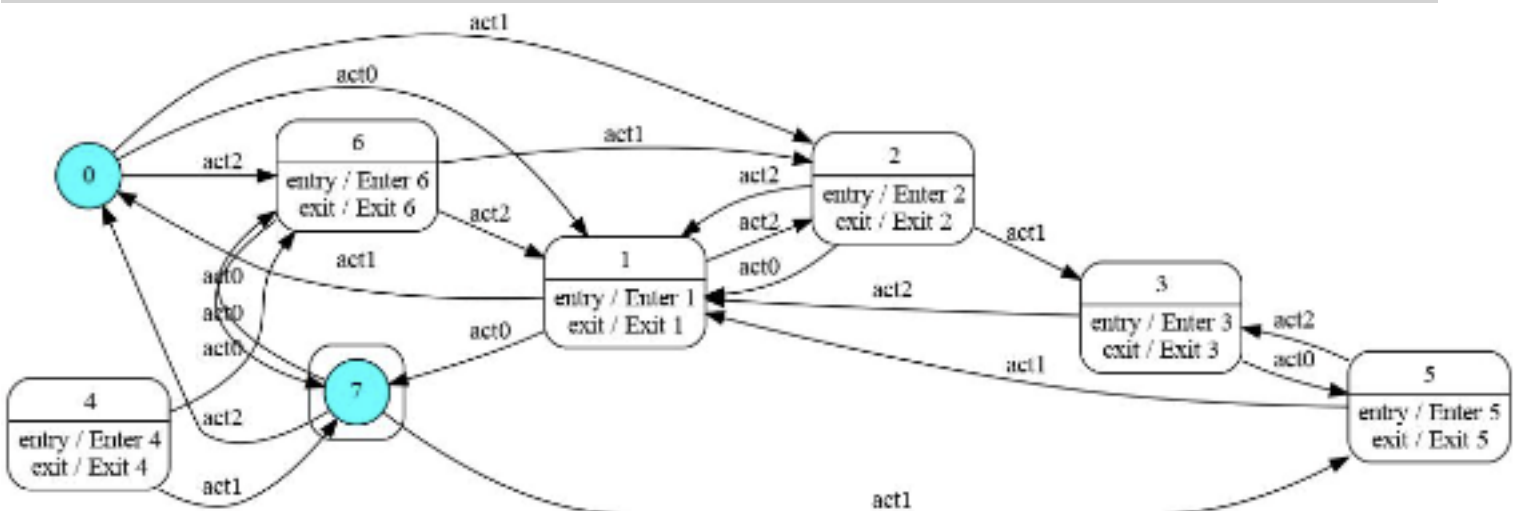
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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%

3.State coverage: 50%

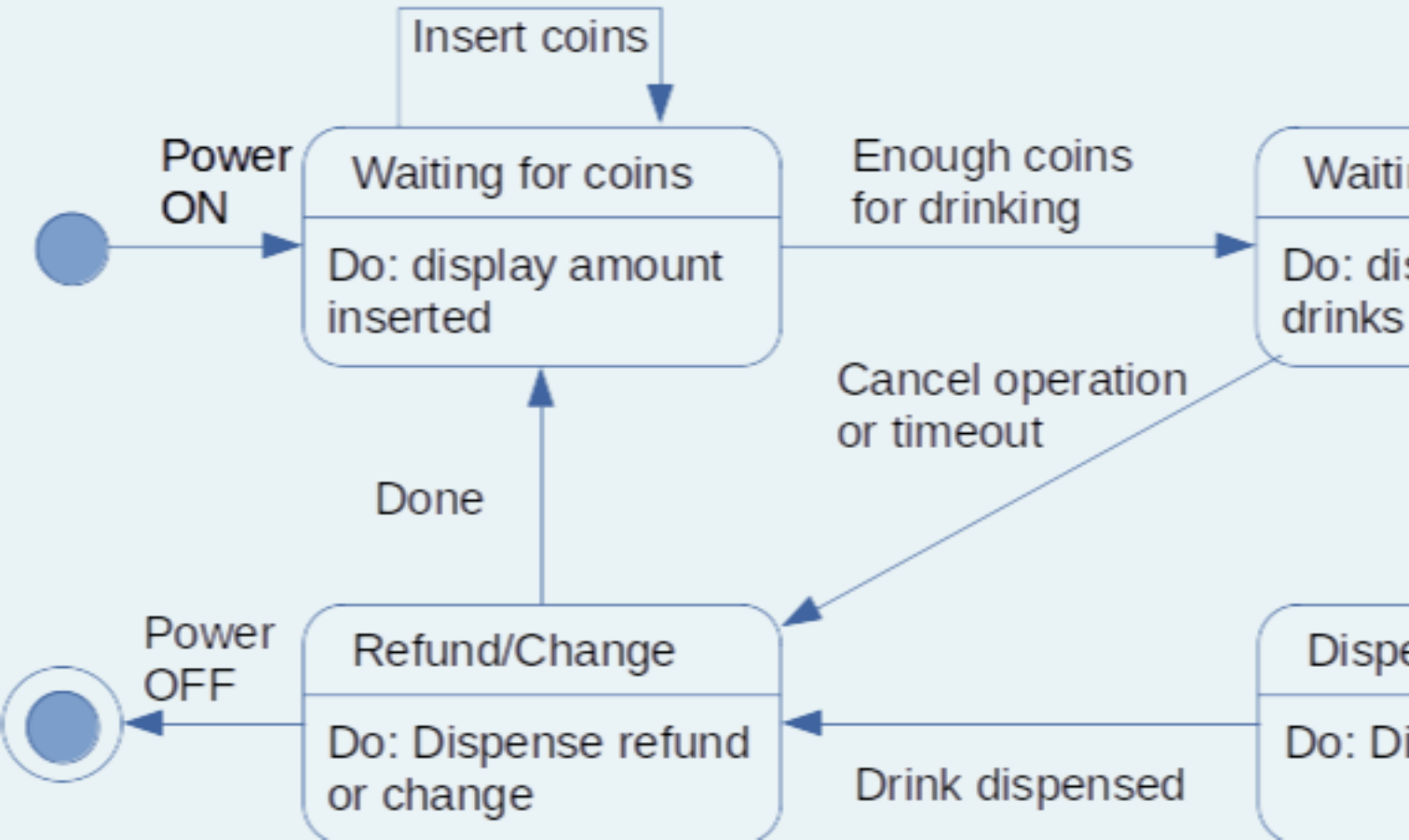
Risposta : 3

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

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

Uno scenario è una sequenza di stati.

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



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:

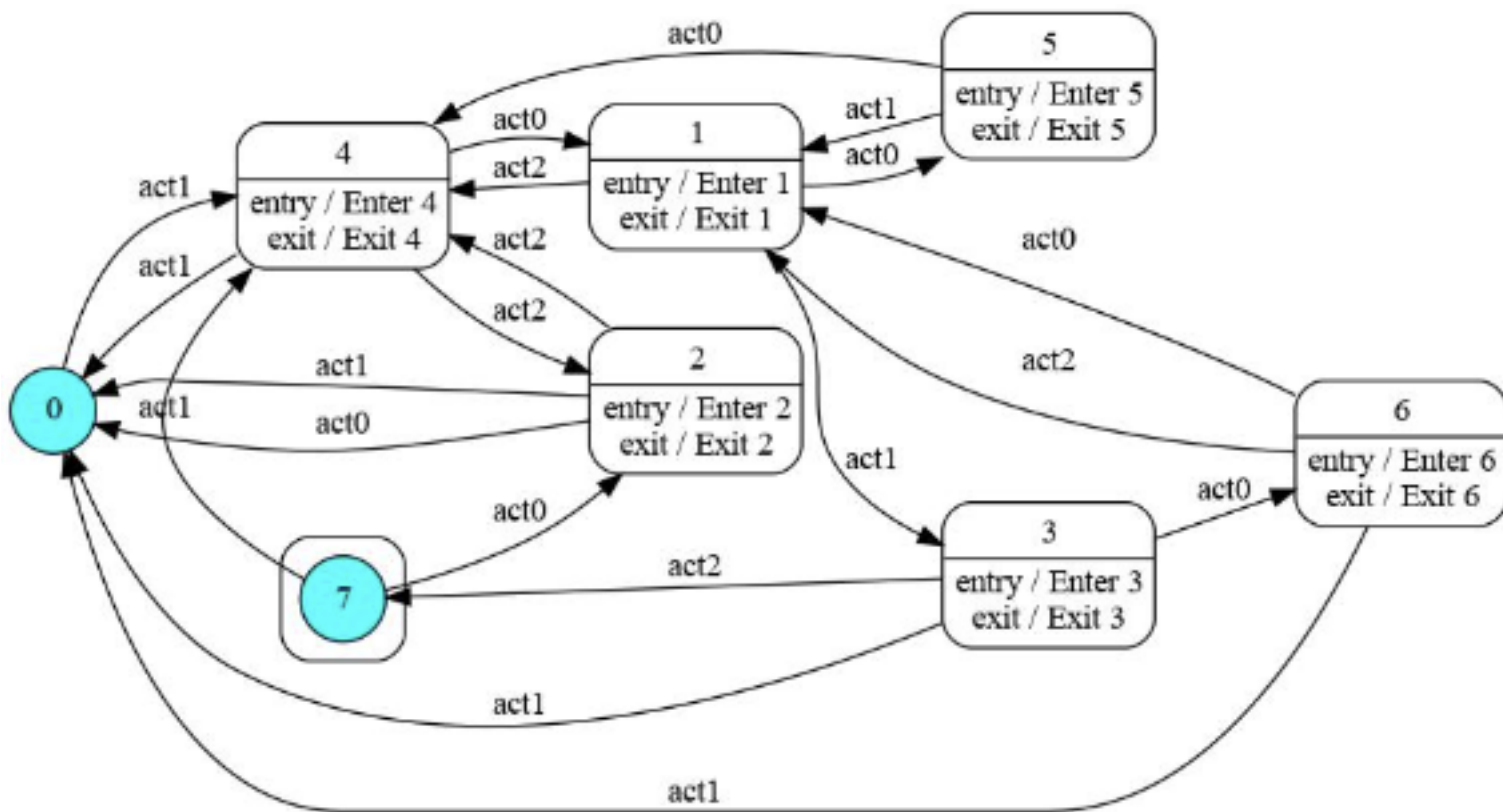
Test case 1: act1 act0 act2 act0 act0 act0 act2 act1 act1 act0 act2 act0 act2 act2 act1 act1 act0 act2 act2 act2 act1 act1 act2 act0 act1 act0 act1 act2

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

act2 act2 act0 act1 act2 act0 act1 act0 act1 act2

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

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



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

Risposta : 2

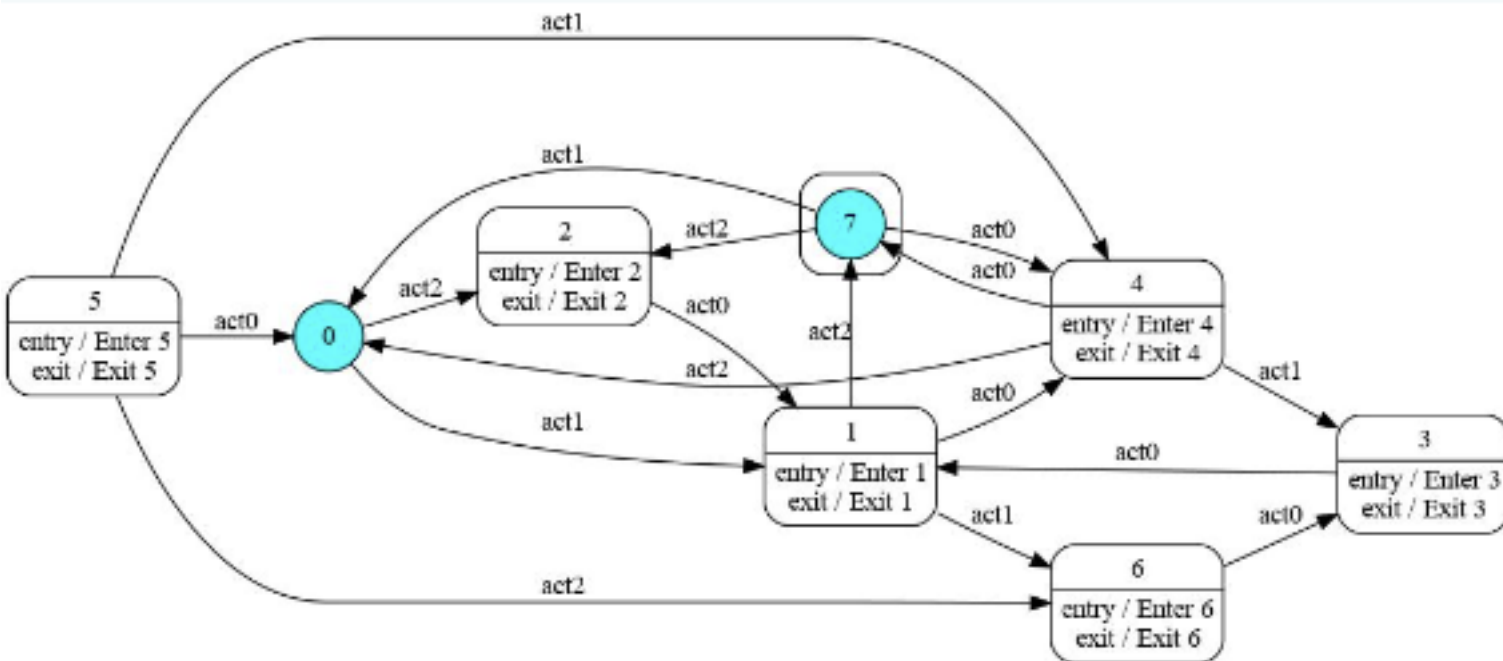
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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

Test case 1: act2 act1 act1 act2 act1 act1 act0

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

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



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.

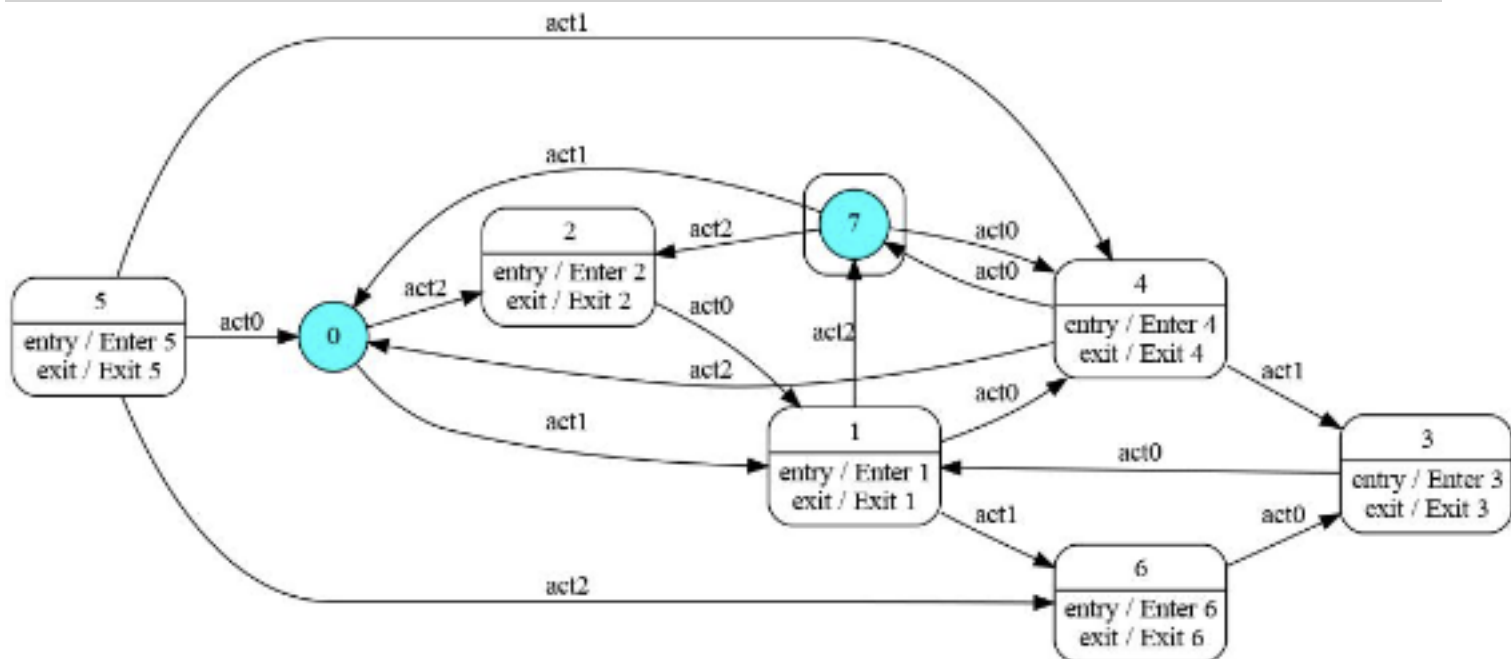
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

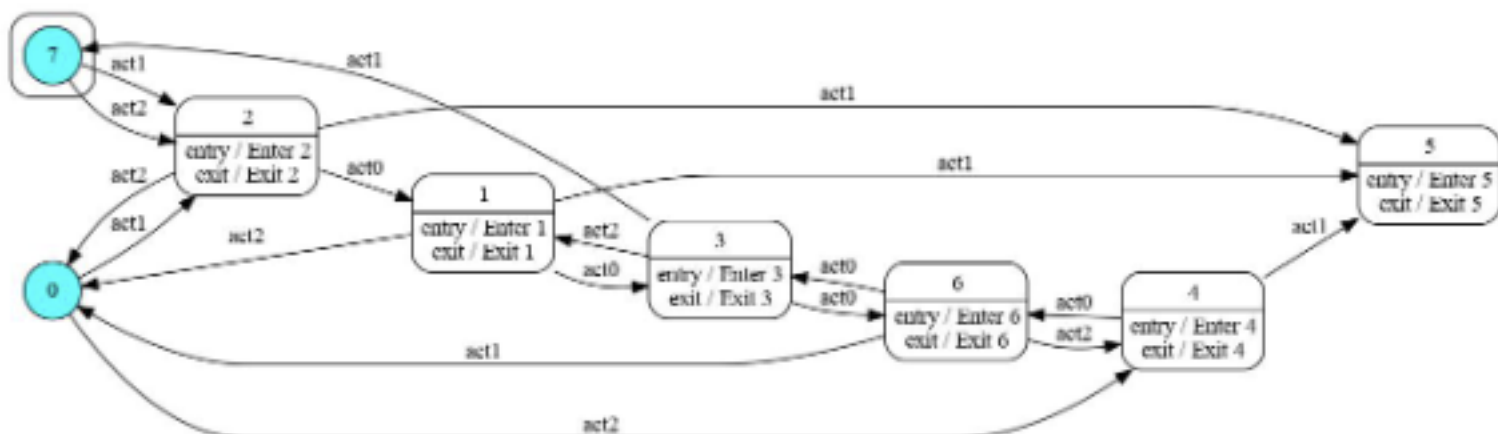
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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 act2 act0 act2 act2 act0 act0 act2 act2 act2 act2 act1

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

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



1. State coverage: 87%

2. State coverage: 100%

3. State coverage: 50%

Risposta : 1

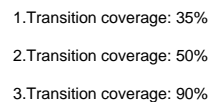
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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

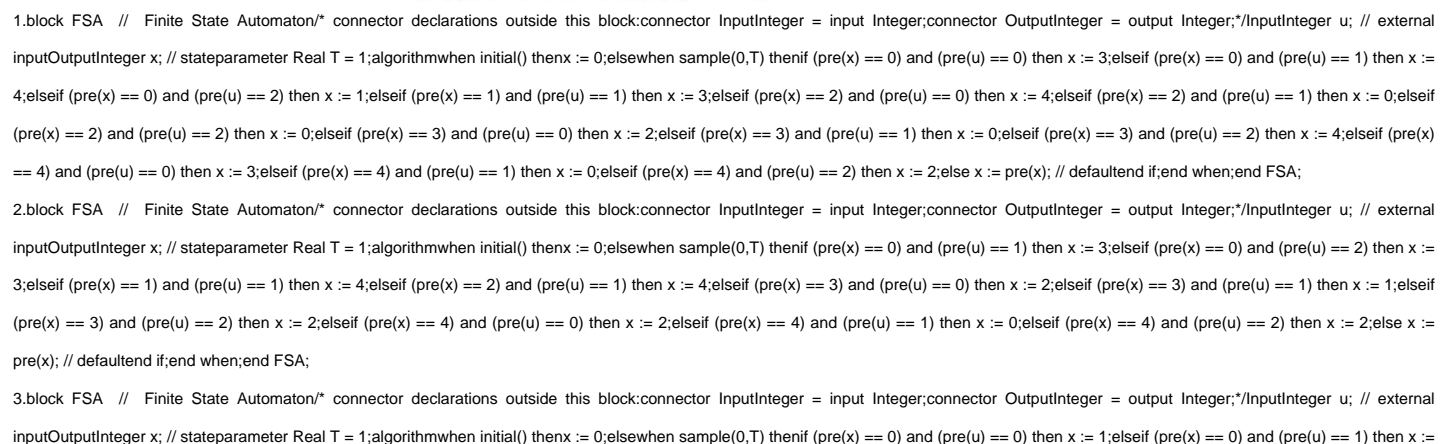
Test case 1: act1 act2 act0

Test case 2: act2 act2 act2 act2 act2 act2 act0

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



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



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

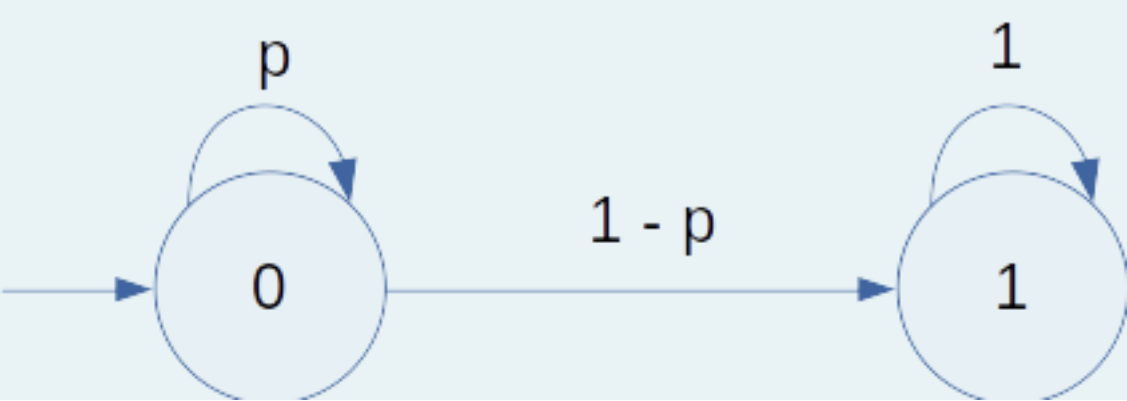
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il tempo necessario per completare la fase x è time(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappresenta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



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

Risposta : 3

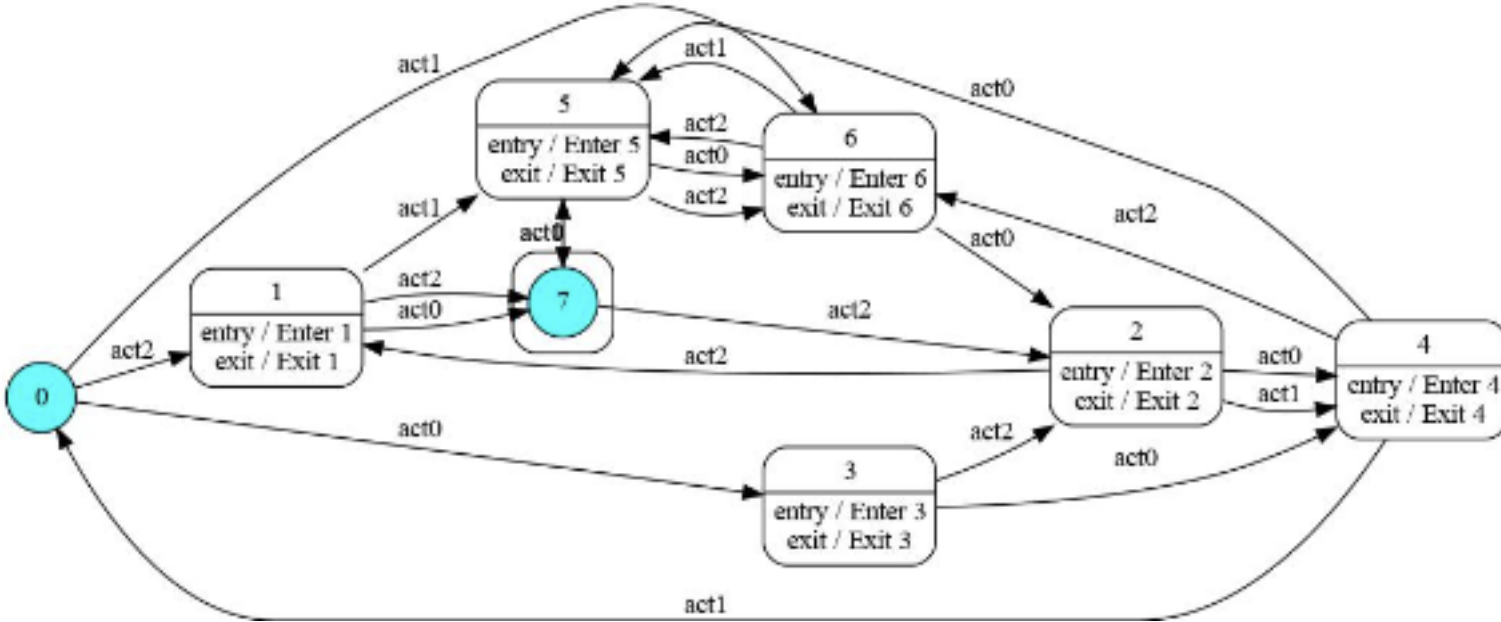
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il costo dello stato (fase) x è c(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappresenta il completamento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

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

Quindi il costo C(X) della sequenza di stati X = x(0), x(1), x(2), 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

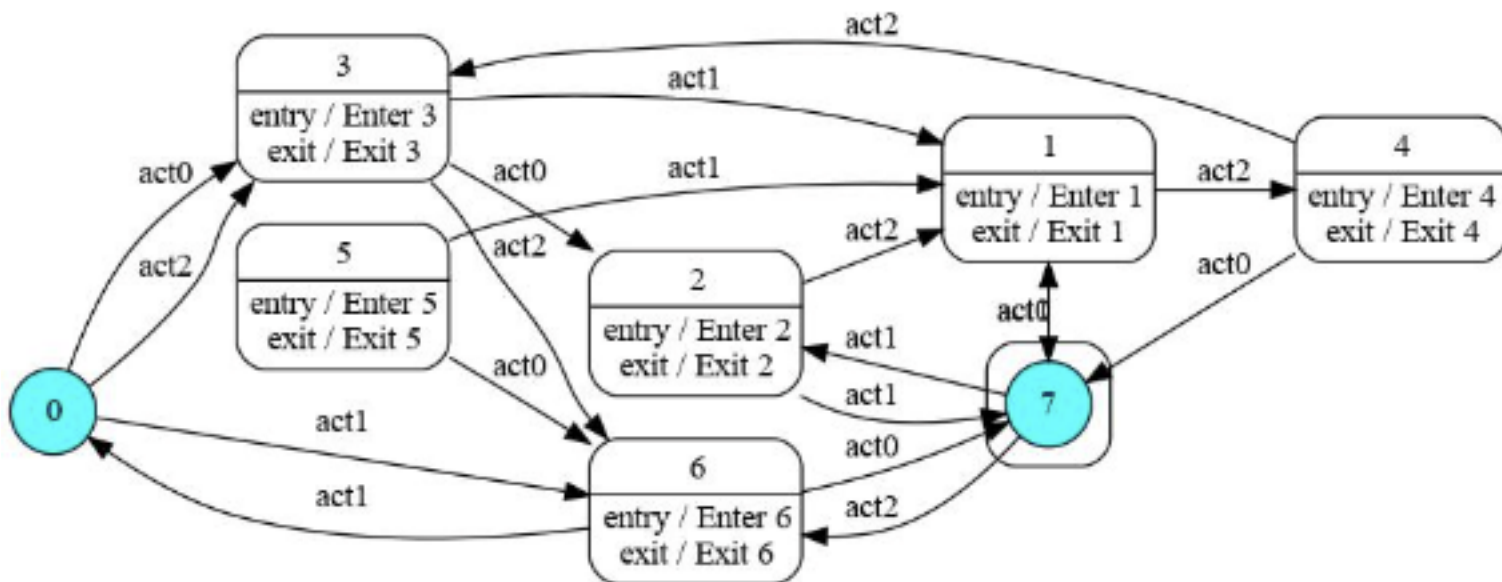
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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

Test case 1: act2 act1 act1

Test case 2: act0 act0 act2 act1

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



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

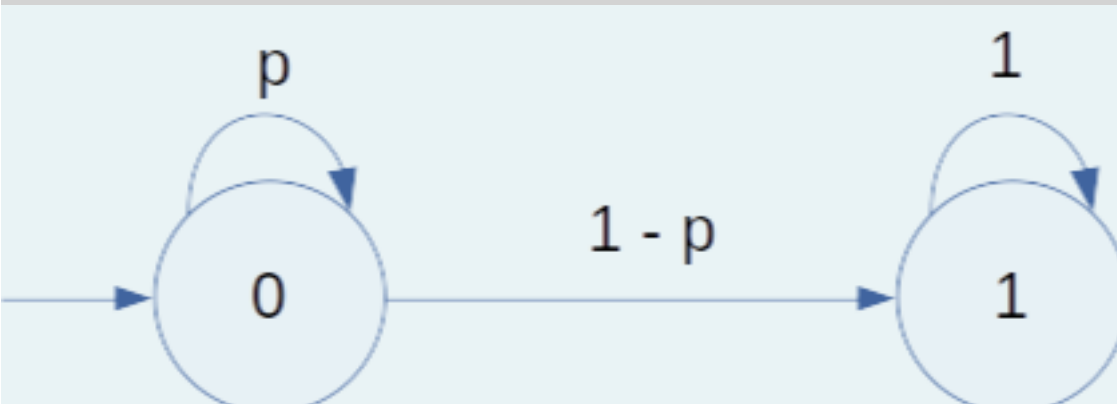
Risposta : 1

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

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

Uno scenario è una sequenza di stati.

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



- 1.0.28
- 2.0.12
- 3.0.42

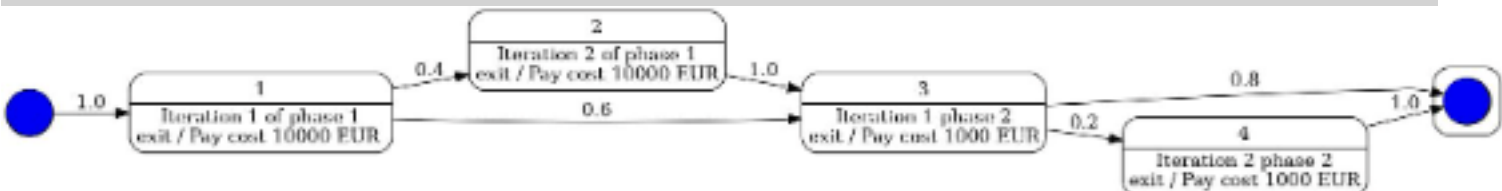
Risposta : 1

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

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

Uno scenario è una sequenza di stati.

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



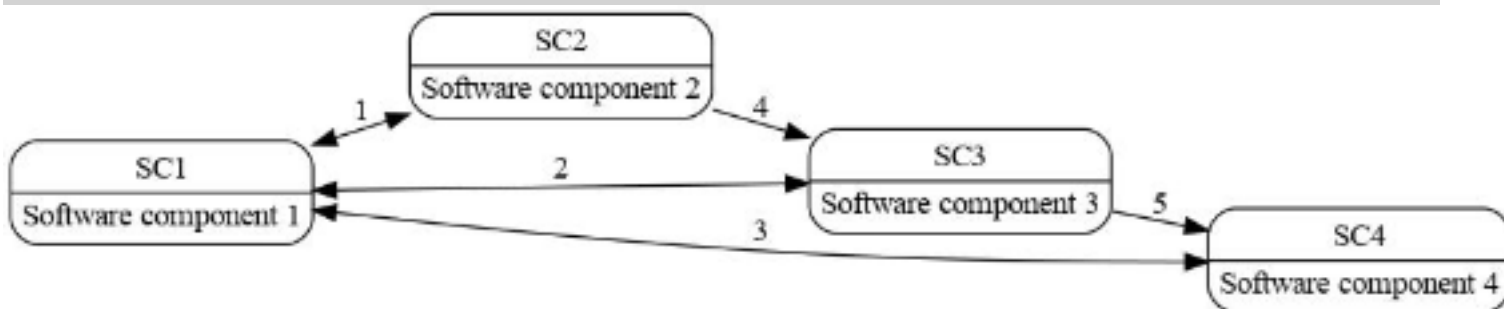
1.0.32

2.0.08

3.0.12

Risposta : 3

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

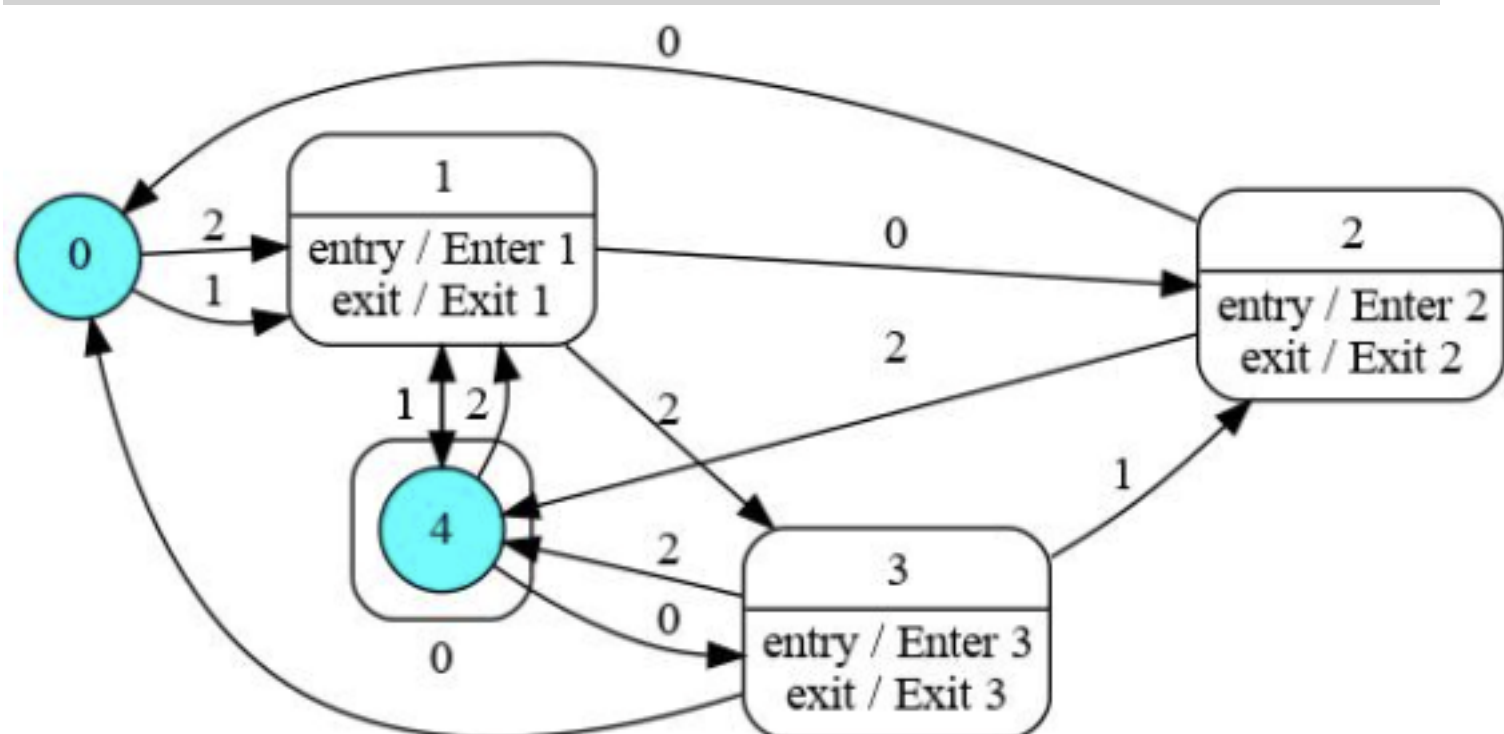


1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() then x := 0;elsewhen sample(0,T) then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

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

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

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

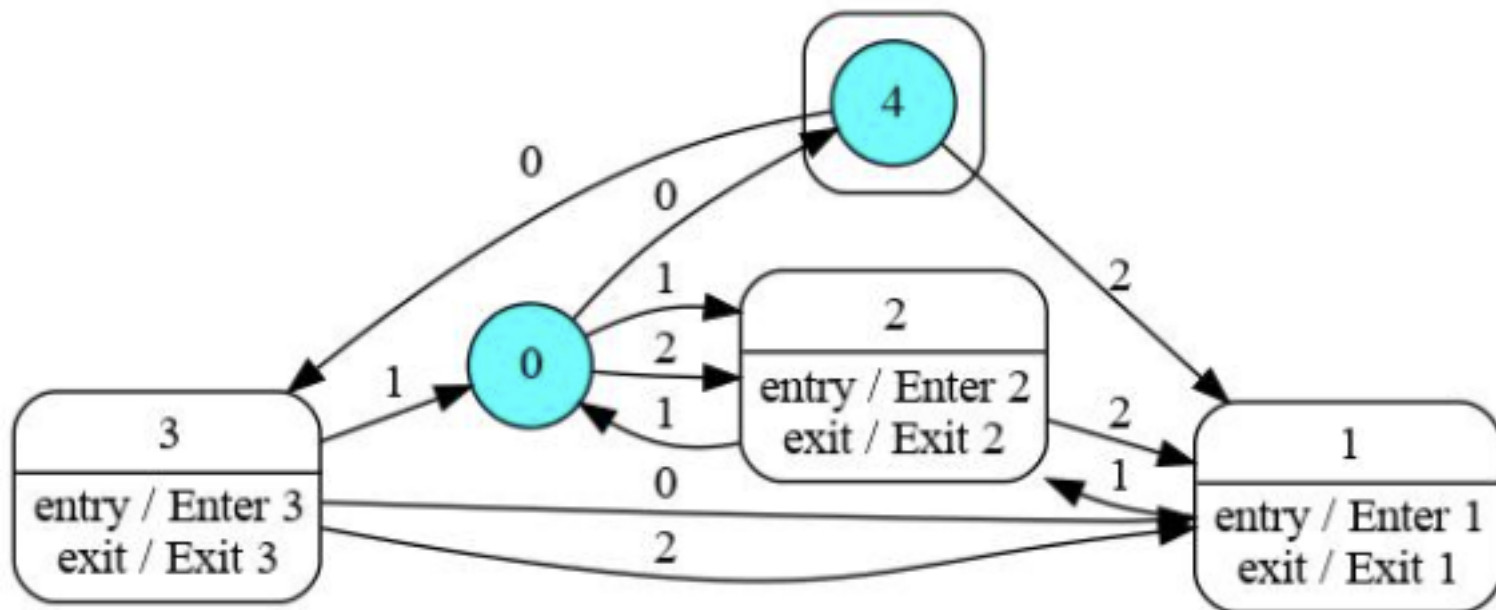
```
(pre(x) == 2) and (pre(u) == 2) then x := 0;elseif (pre(x) == 3) and (pre(u) == 1) then x := 2;elseif (pre(x) == 4) and (pre(u) == 0) then x := 1;elseif (pre(x) == 4) and (pre(u) == 1) then x := 1;else x := pre(x); // defaultend if;end when;end FSA;
```

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

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

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

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

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

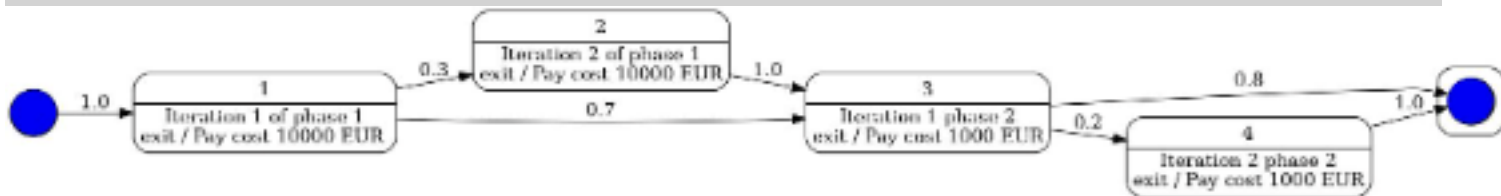
Risposta : 3

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

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

Uno scenario è una sequenza di stati.

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



1.0.24

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.

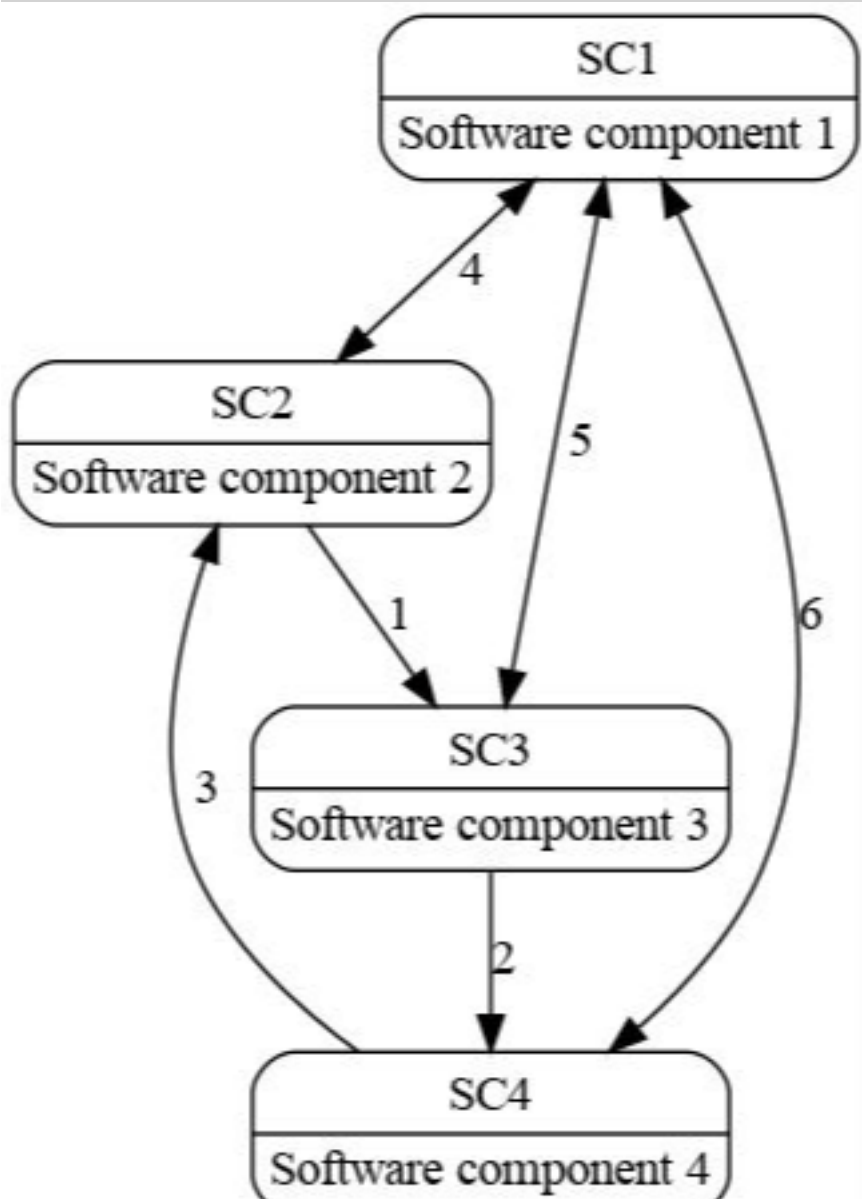
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

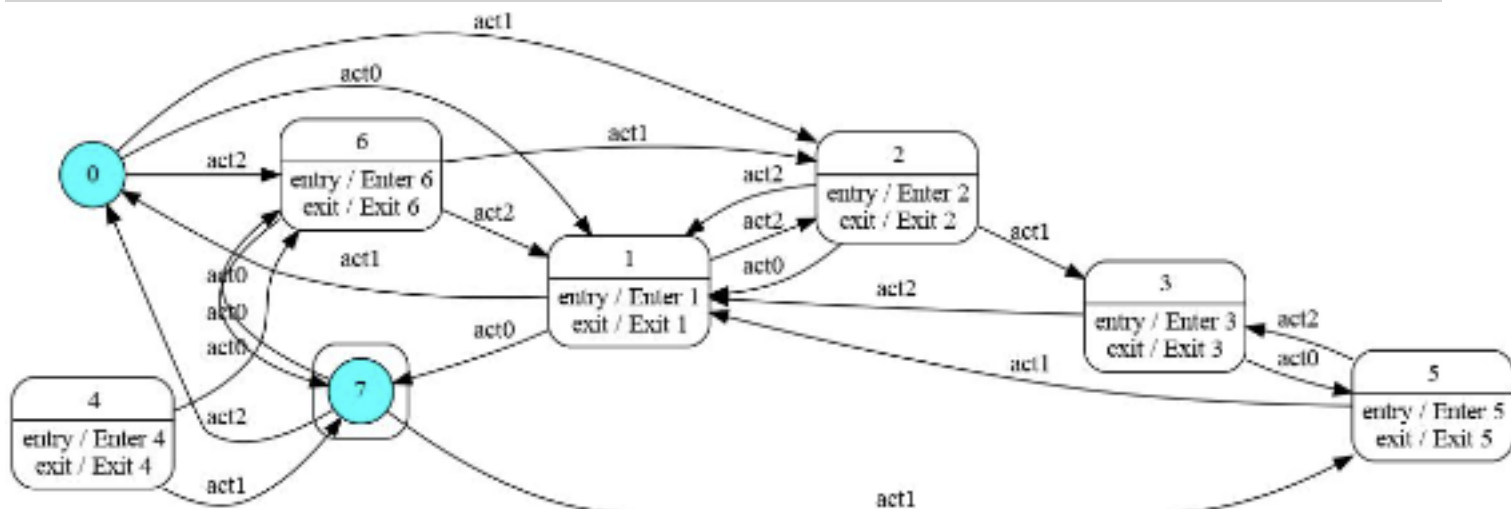
Si consideri il seguente insieme di test cases:

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



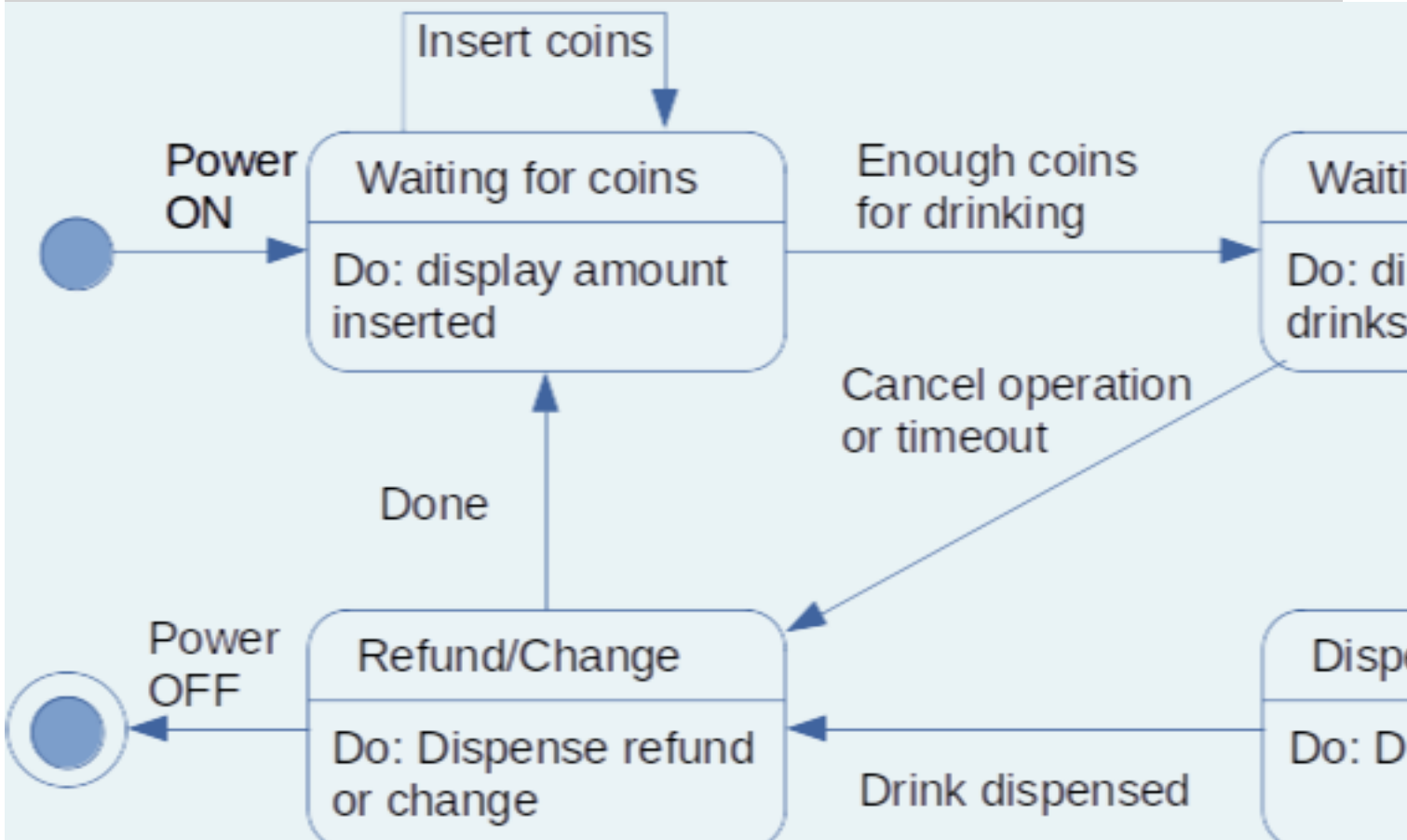
1.State coverage: 90%

2.State coverage: 80%

3.State coverage: 100%

Risposta : 3

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



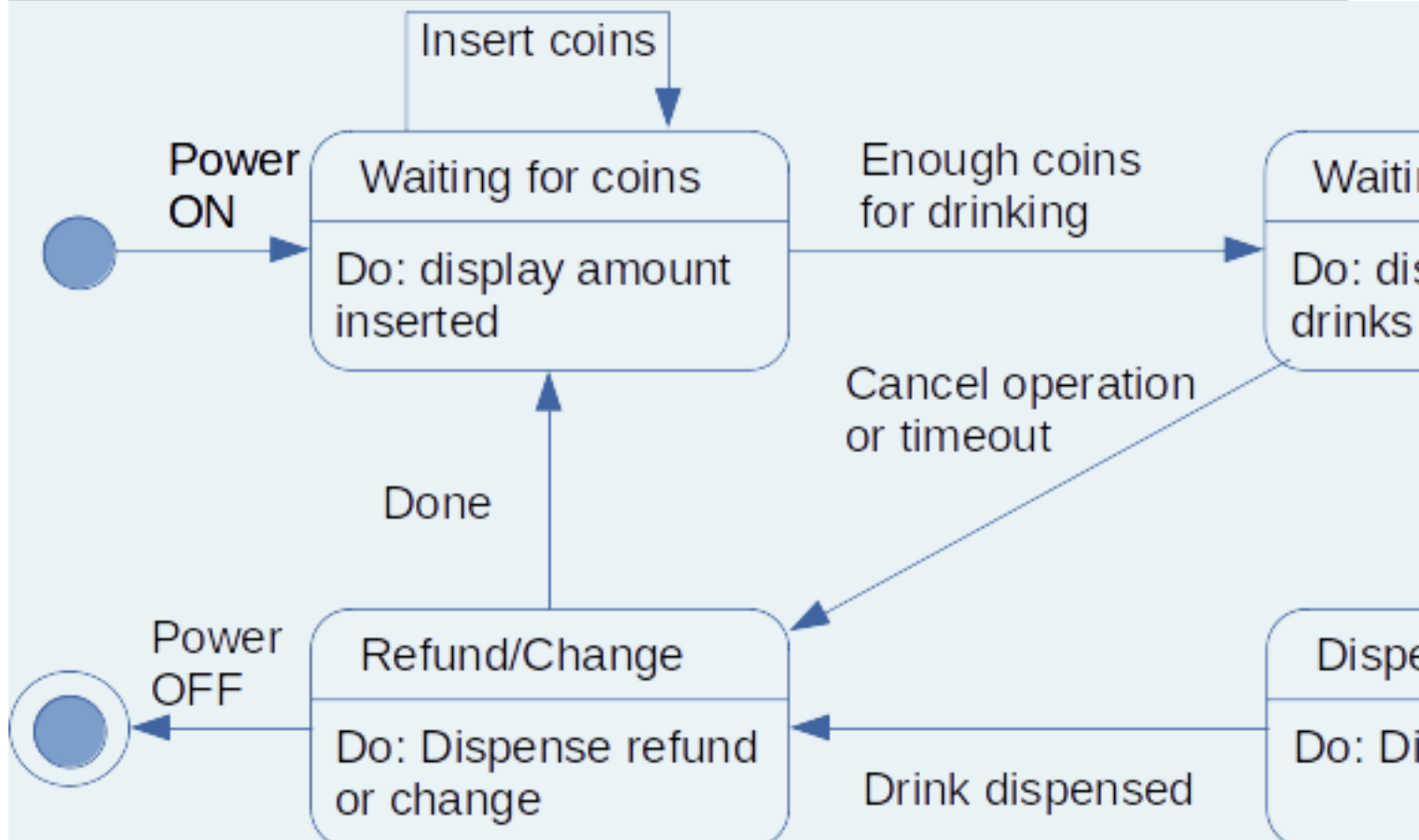
1.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in parallelo.

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

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 ?



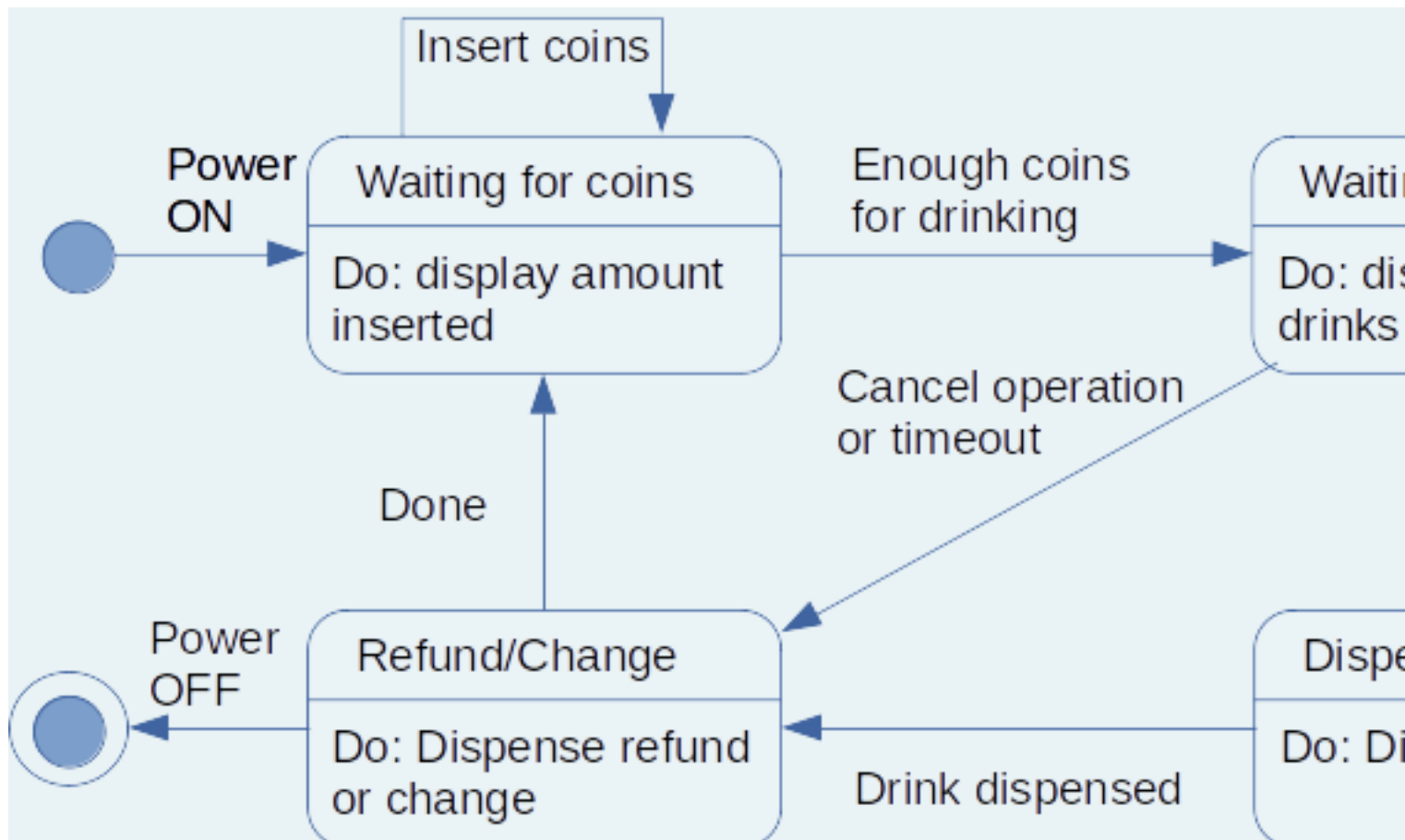
1. Anche se ci sono abbastanza monete potrebbe non essere possibile ottenere la bevanda selezionata.

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

Risposta : 1

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.

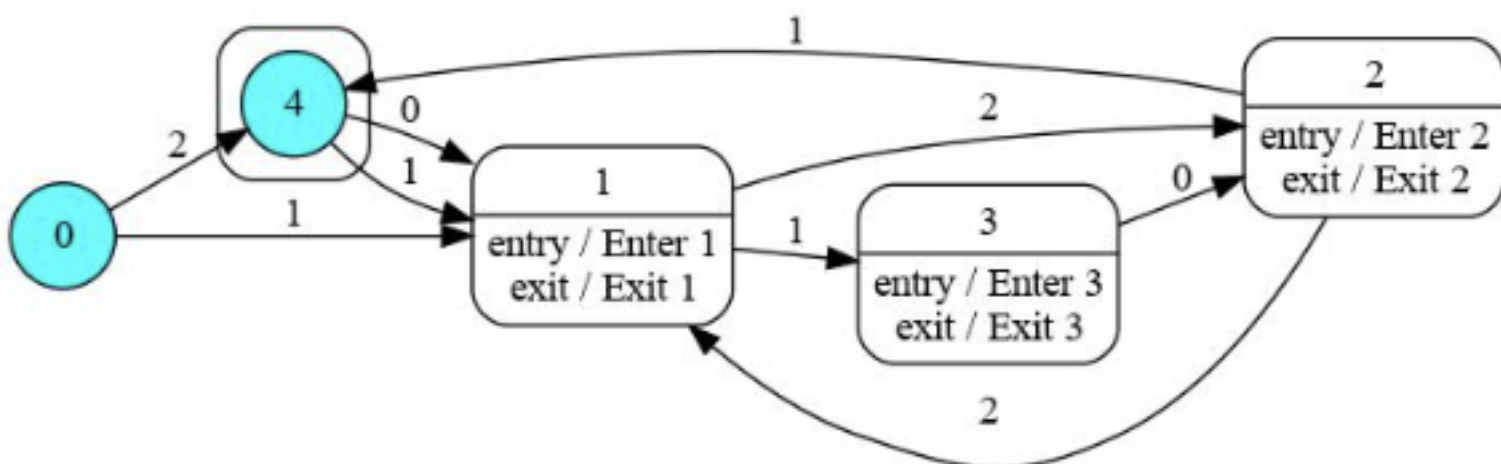
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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.

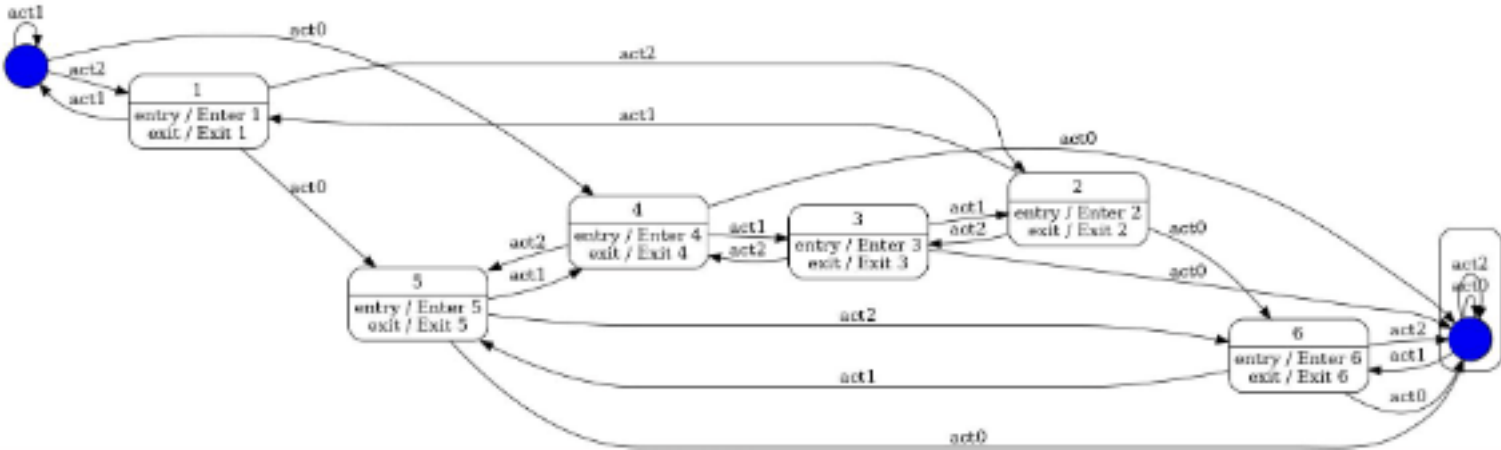
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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

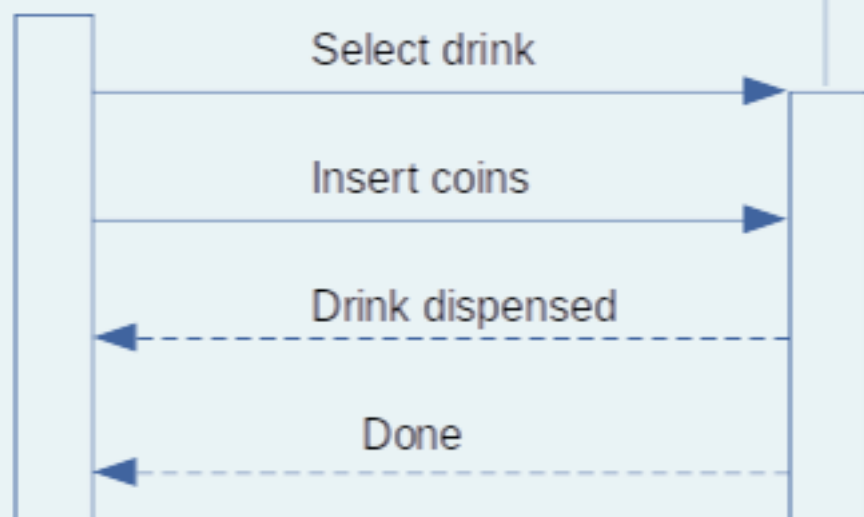
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?



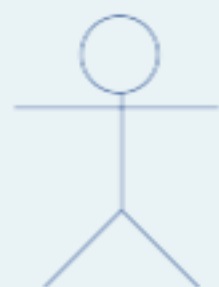
Customer



Vending Machine



Customer



Vending
Machine

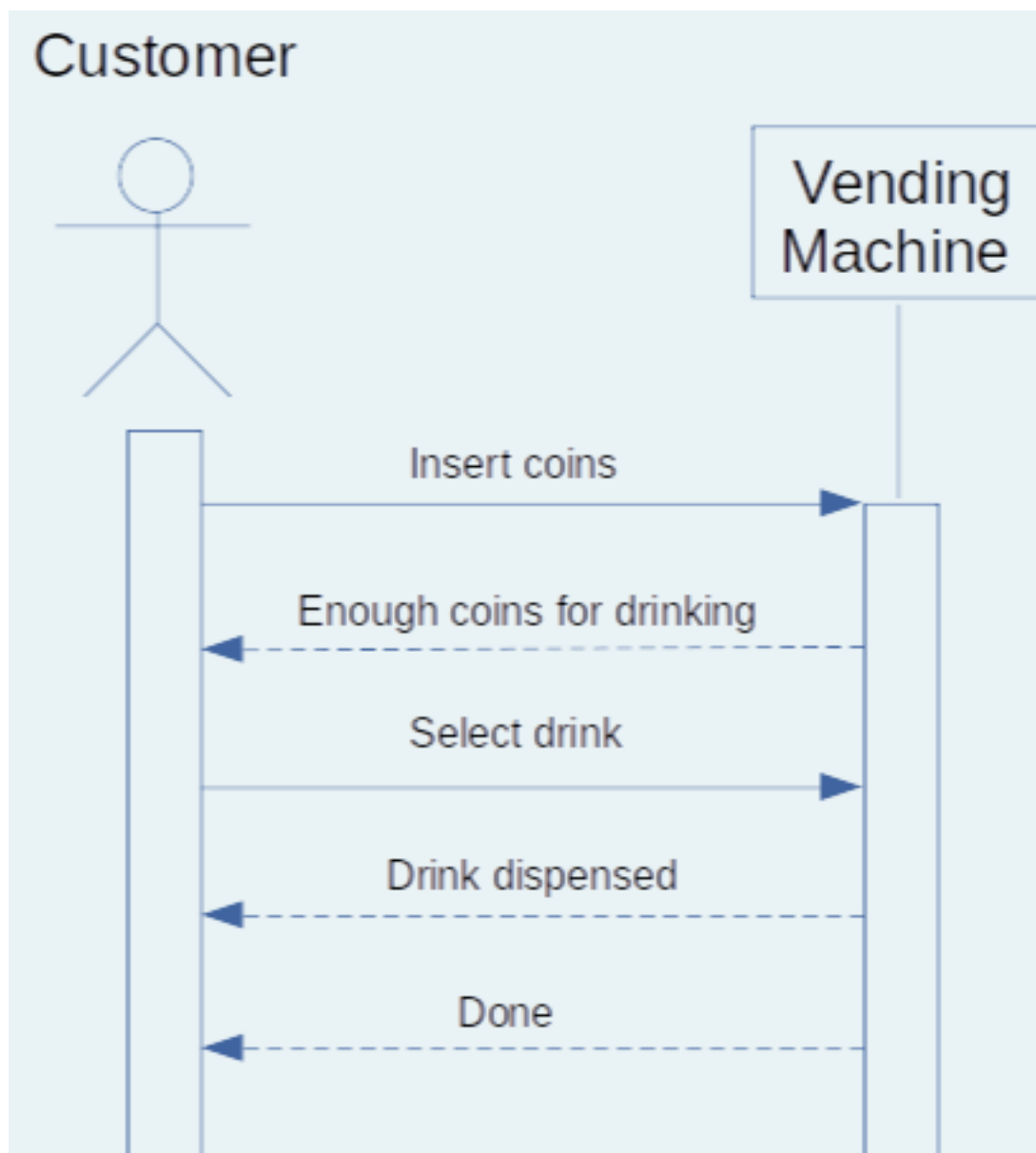
Insert coins

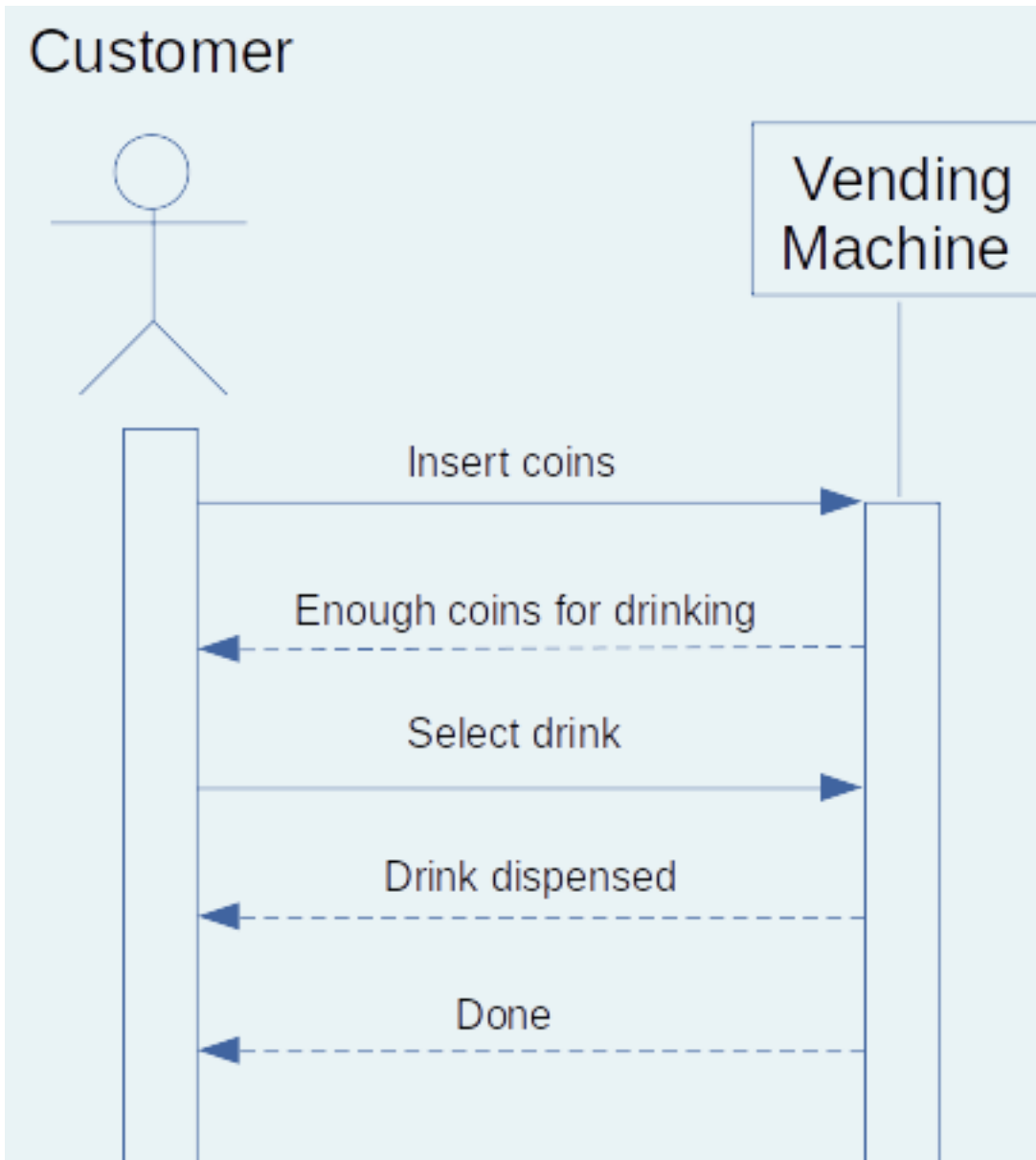
Enough coins for drinking

Select drink

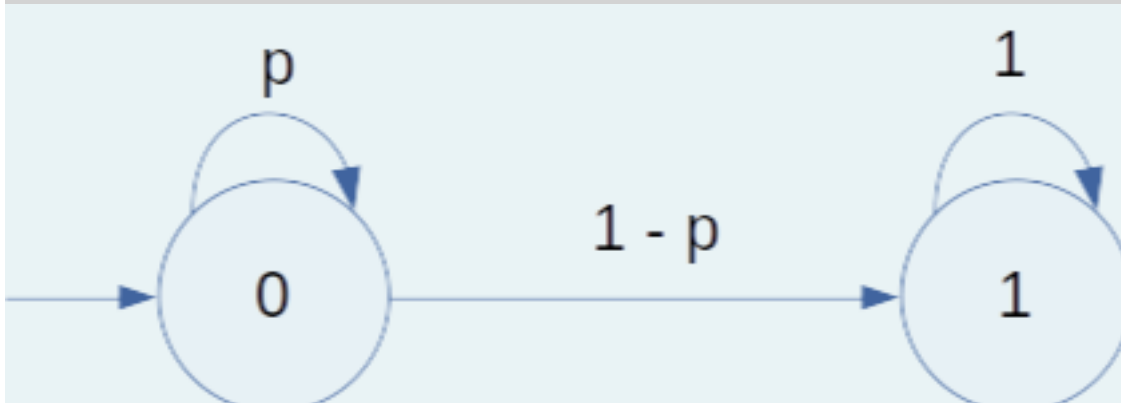
Drink dispensed

Done





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; elseif sample(0, T) thenif (pre(state) == 0) and (Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state := 0; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 3; Machine2Customer := 2;elseif (pre(state) == 3) // refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end CoffeeMachine;

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; elseif sample(0, T) thenif (pre(state) == 0) and
  
```

```
(Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected
then // drink selected state := 2; // dispensing drink Machine2Customer := 0; elseif (pre(state) == 1) and (Customer2Machine == 3) // cancel transaction then // refund state :=
3; // refund/change Machine2Customer := 0; elseif (pre(state) == 2) // drink dispensed then // drink dispensed state := 3; Machine2Customer := 2;elseif (pre(state) == 3) //
refund/change then // refund state := 0; Machine2Customer := 3; // done else state := pre(state); Machine2Customer := pre(Machine2Customer); end if;end when;end
CoffeeMachine;

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

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.

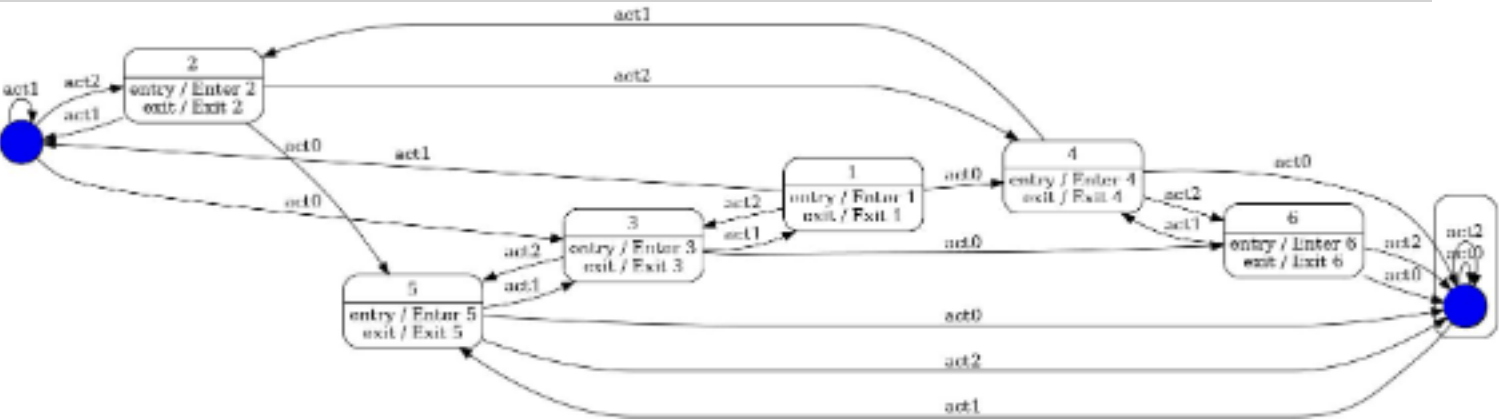
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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

Test case 2: act0 act0 act2

Test case 3: act2 act0 act2

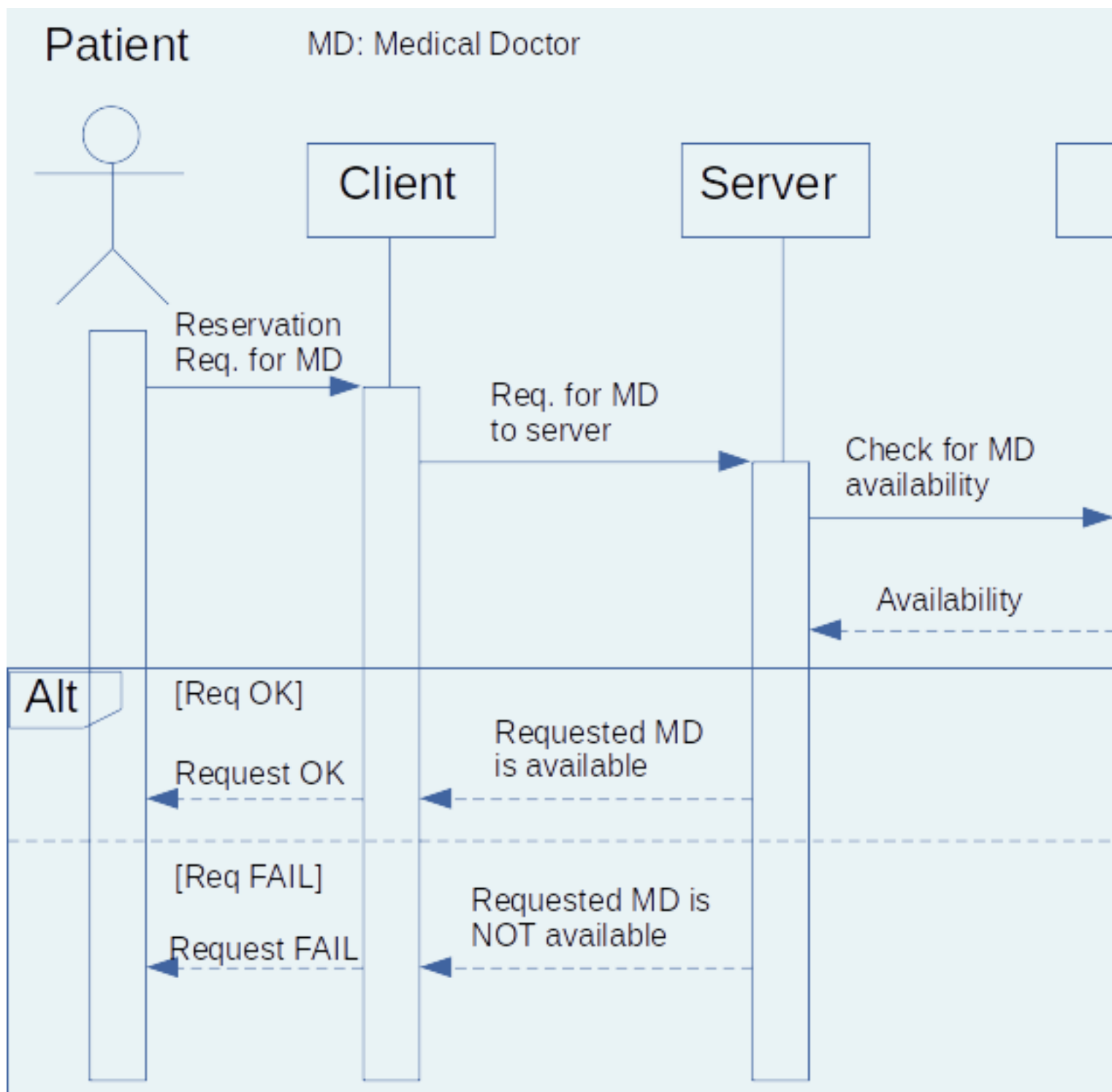
Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



- 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.

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.

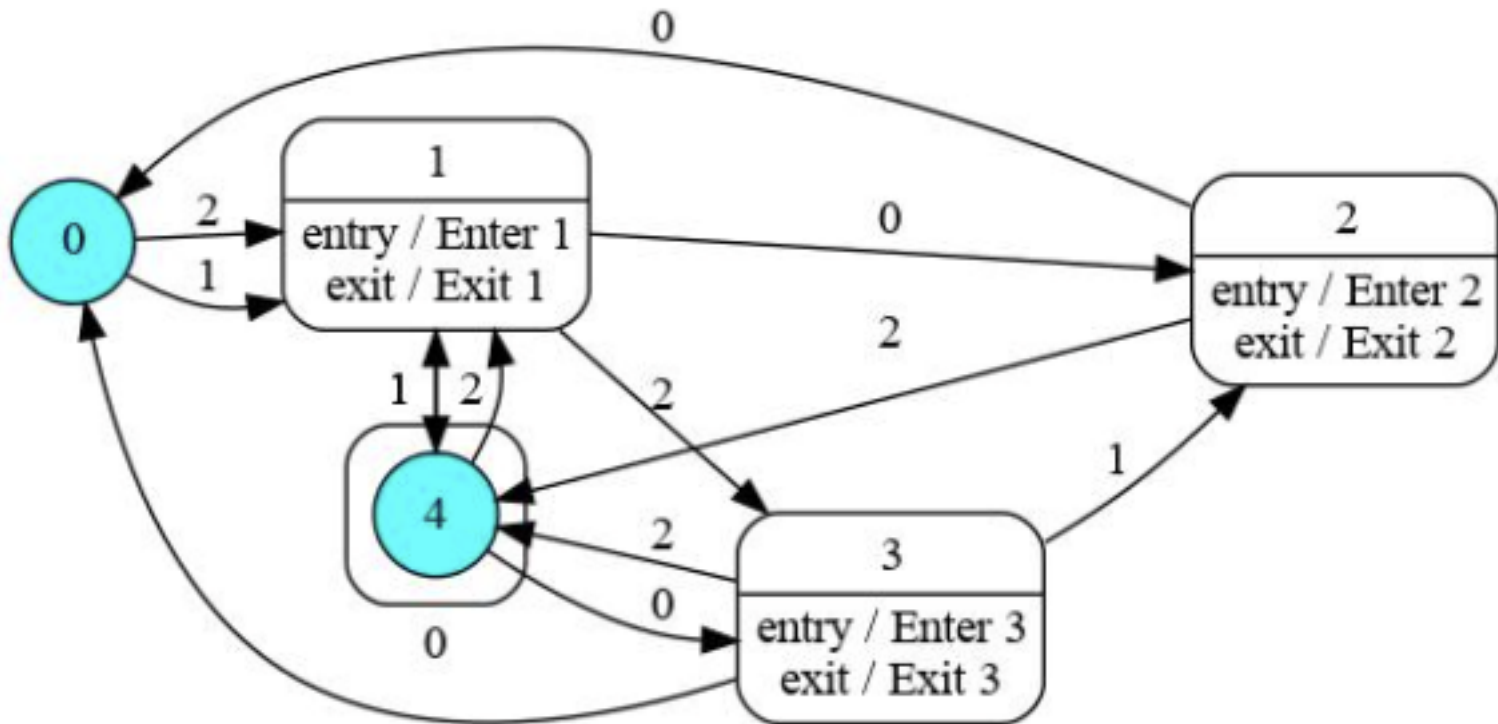
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

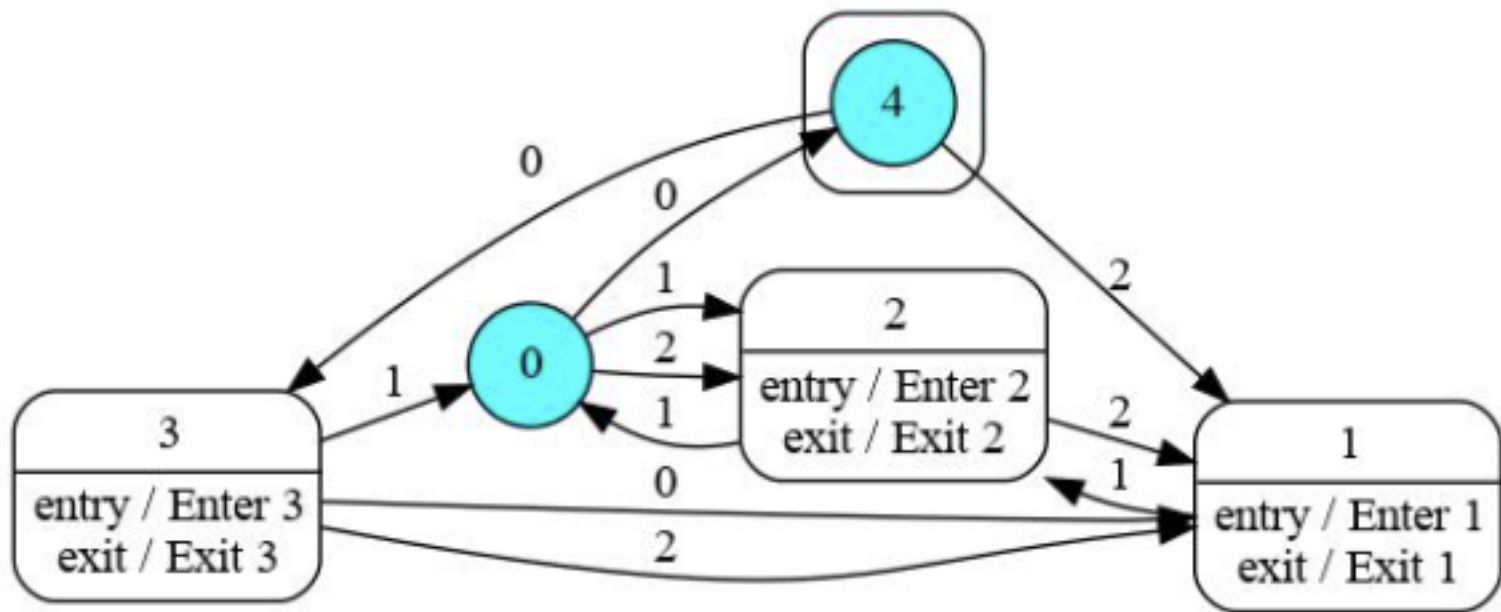
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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

Test case 1: act2 act0 act2 act2 act2

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

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



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.

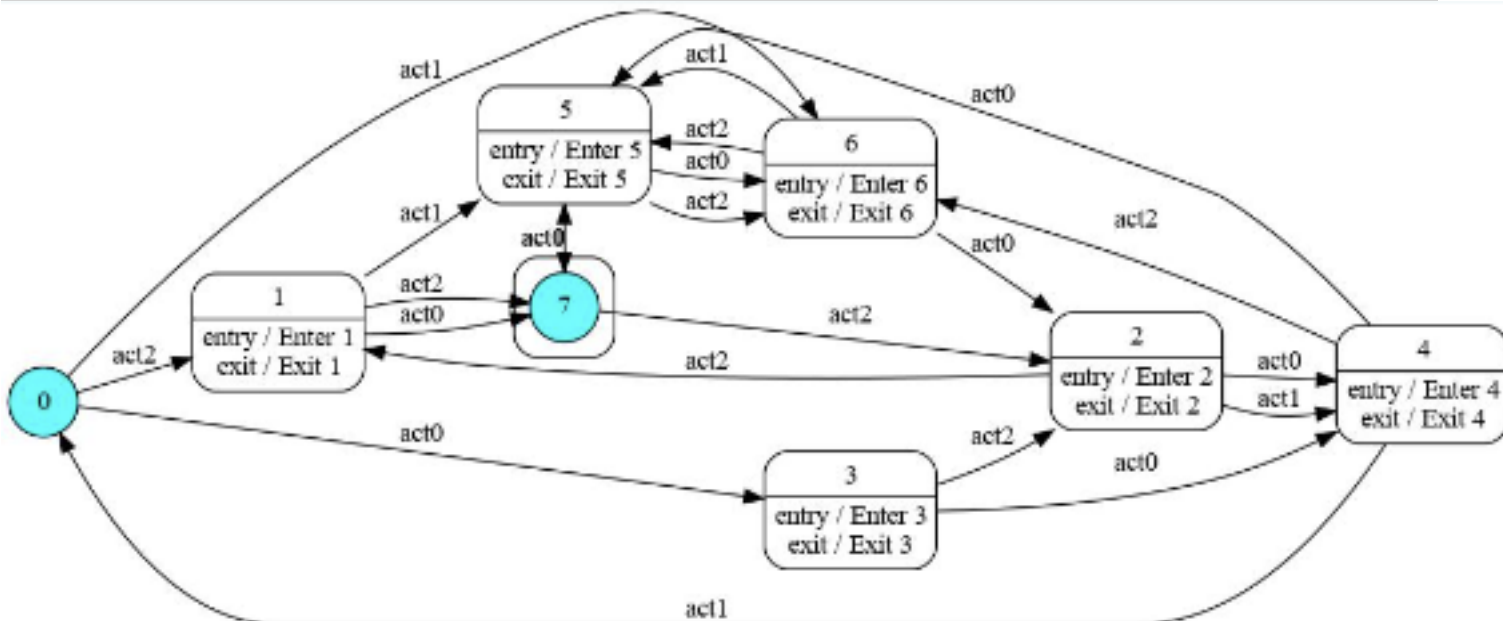
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

Test case 2: act0 act1 act2 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?



1.60%

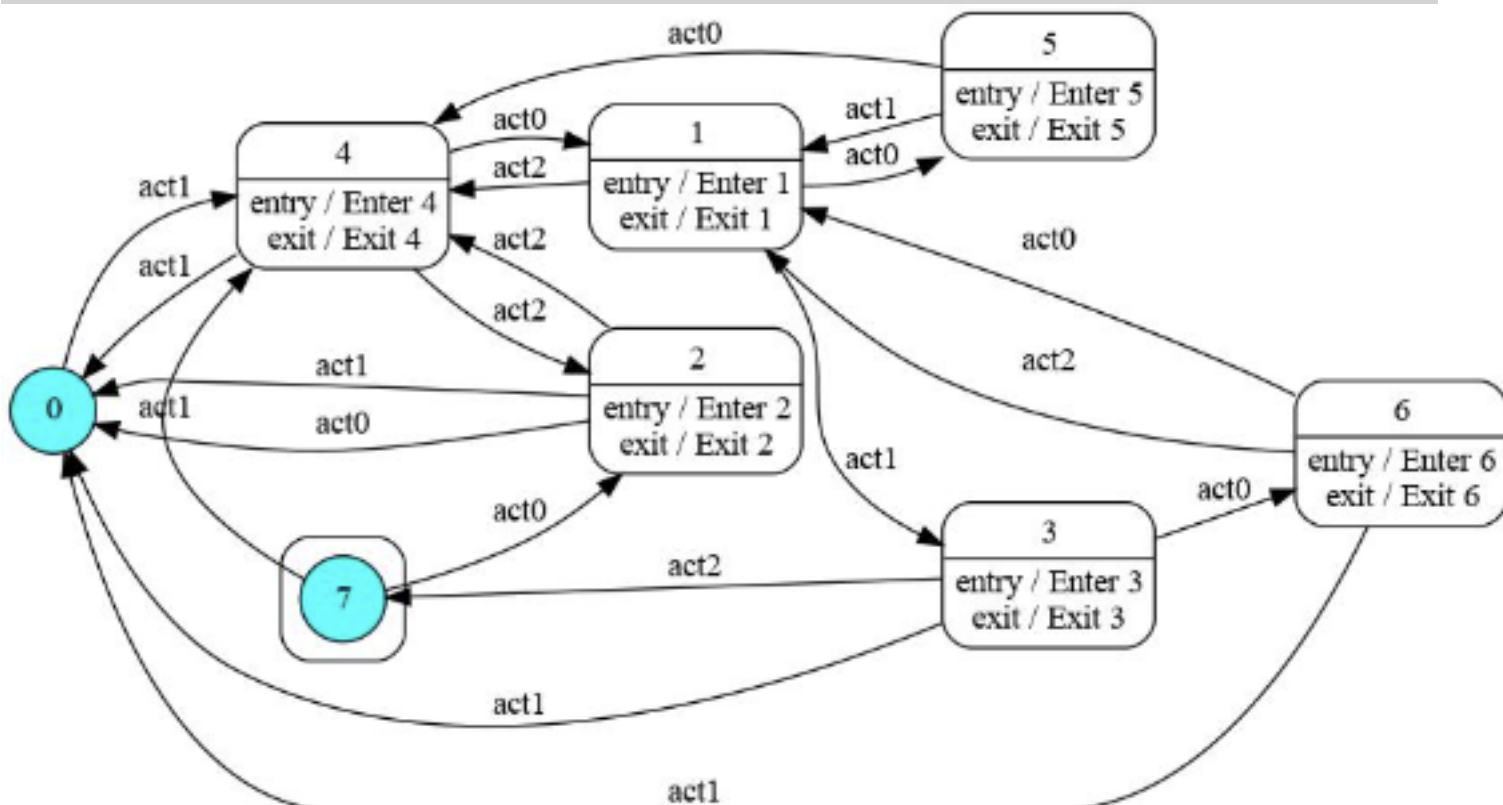
2.80%

3.100%

Risposta : 2

Si consideri l'automa seguente:

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



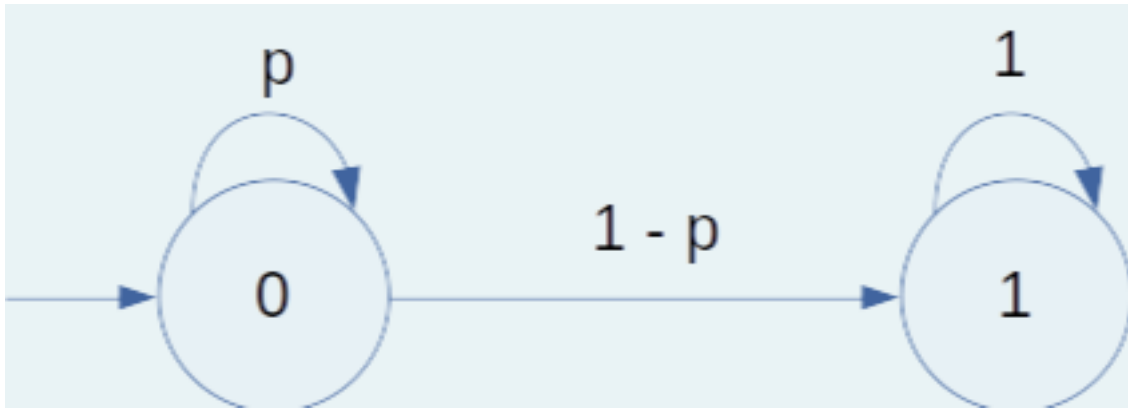
1.function nextinput Integer x;output Integer y;algorithm y := x;end next;class SystemInteger x;initial equationx = 0;equationwhen sample(0, 1) then x = next(pre(x));end when;end System;

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;

Si consideri la seguente Markov Chain:

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

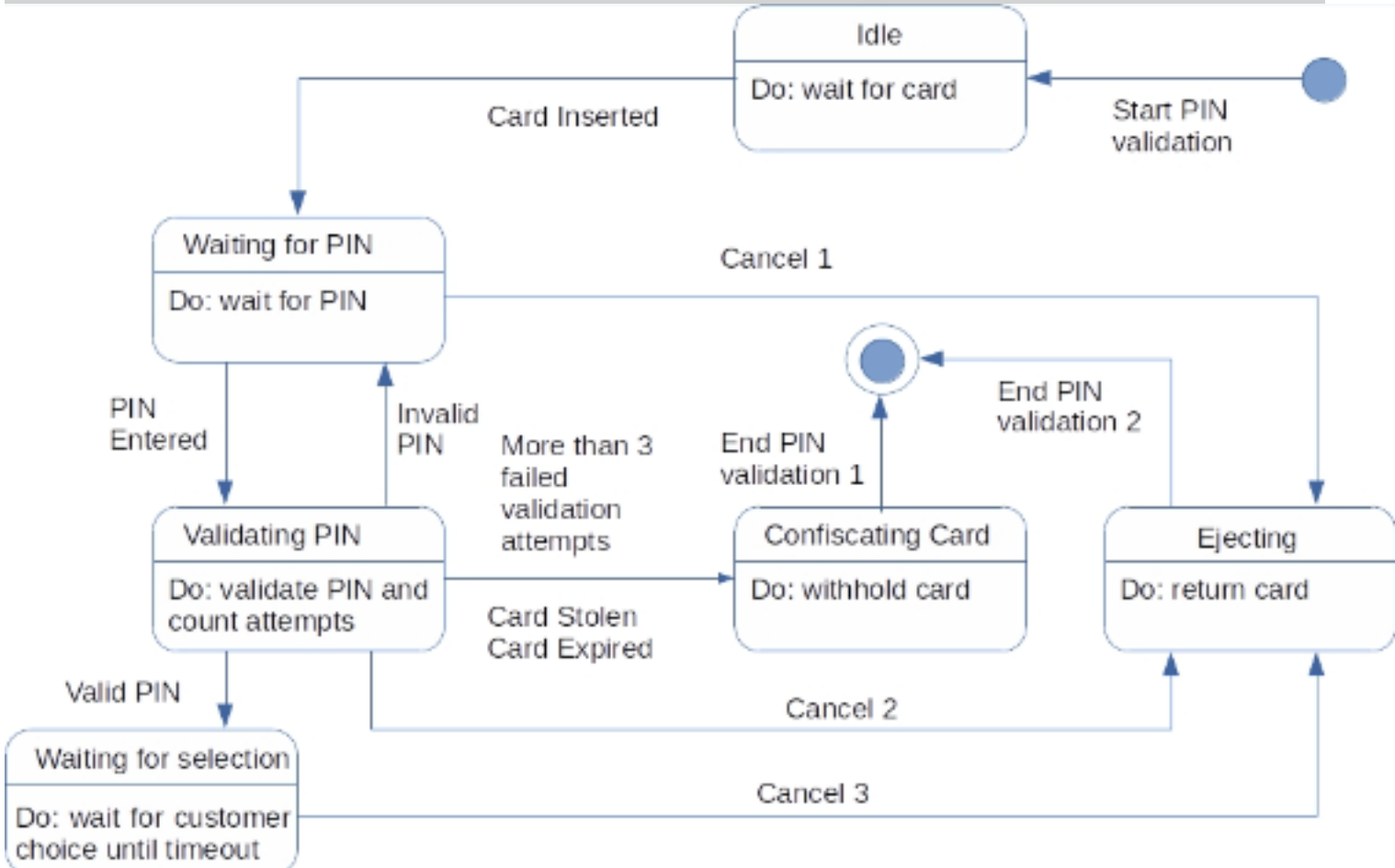


1.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 0, 1-p, 0;0, p, 1-p, 0;p, 0, 0, 1];Integer x; Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen 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] + 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, 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;

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, 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] + 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?

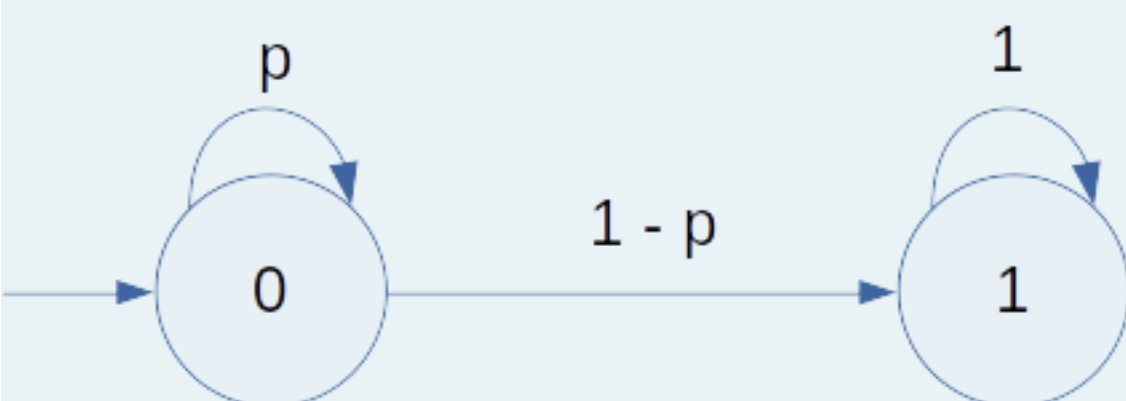


```
1.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [0, 1, 0, 0; p, 0, 1-p, 0; 0, p, 1-p, 0; 0, 0, 0, 1];Integer x; Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen 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] + 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] = [0, 1, 0, 0; p, 0, 1-p, 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] + 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] = [0, 1, 0, 0; p, 0, 0, 1-p; 0, 0, 1-p, 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] + A[x, F3]) then x := F3; else x := End;end if;end when;end System;
```

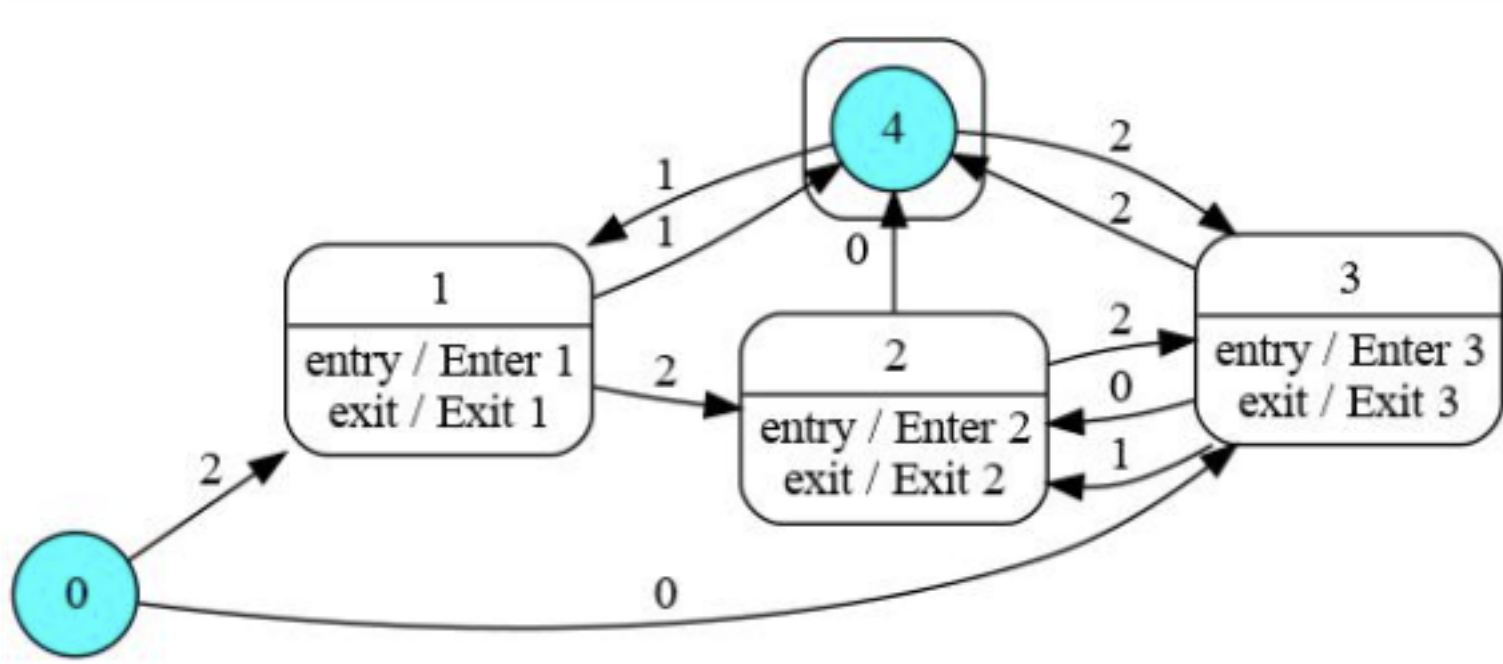
Quale delle seguenti frasi è corretta riguardo all'ctivity diagram in figura ?



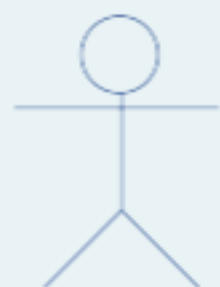
- 1.Una volta selezionato il piatto di mare da preparare, la stessa persona prepara prima il pesce e poi il contorno.
- 2.Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in parallelo.
- 3.Una volta selezionato il piatto di mare da preparare, la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

Risposta : 2

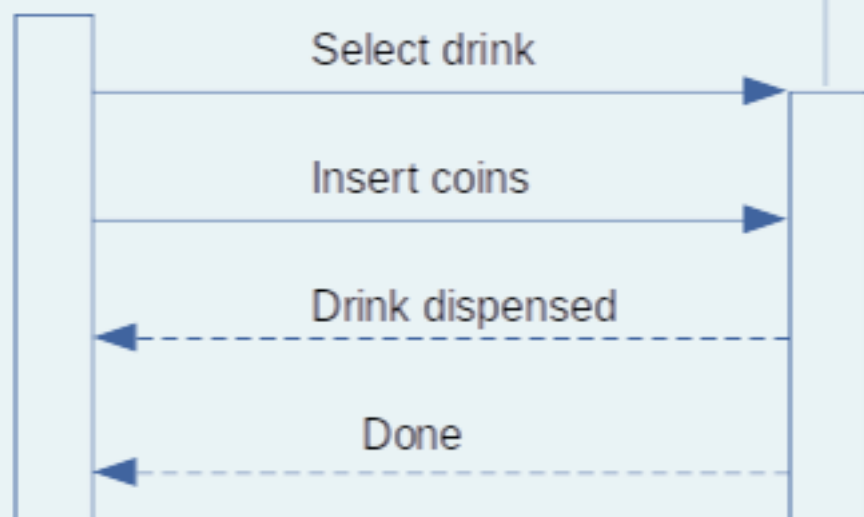
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?



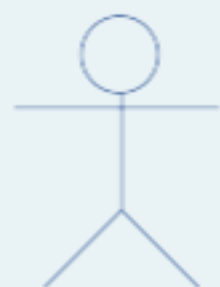
Customer



Vending Machine



Customer



Vending
Machine

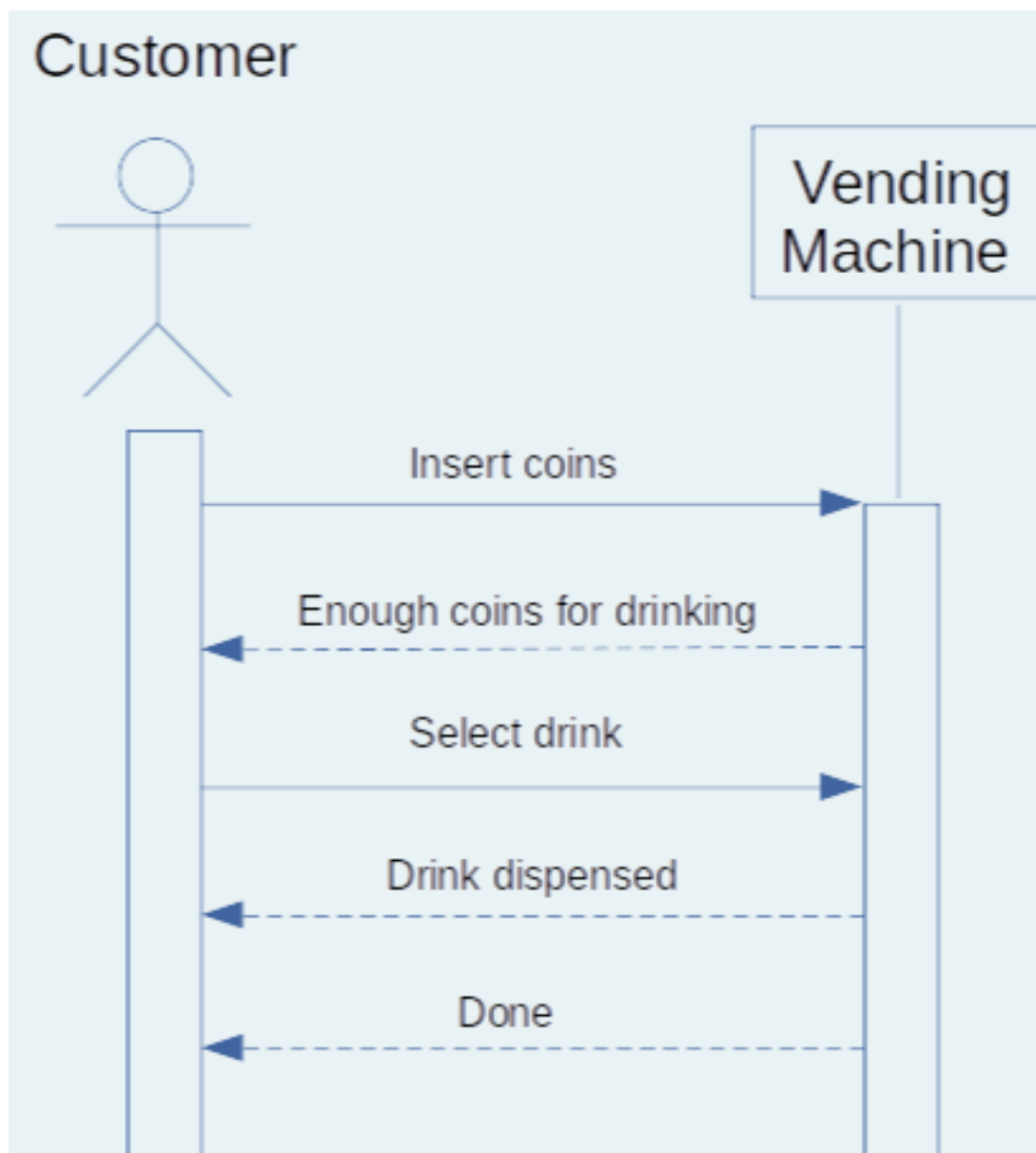
Insert coins

Enough coins for drinking

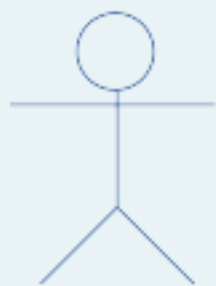
Select drink

Drink dispensed

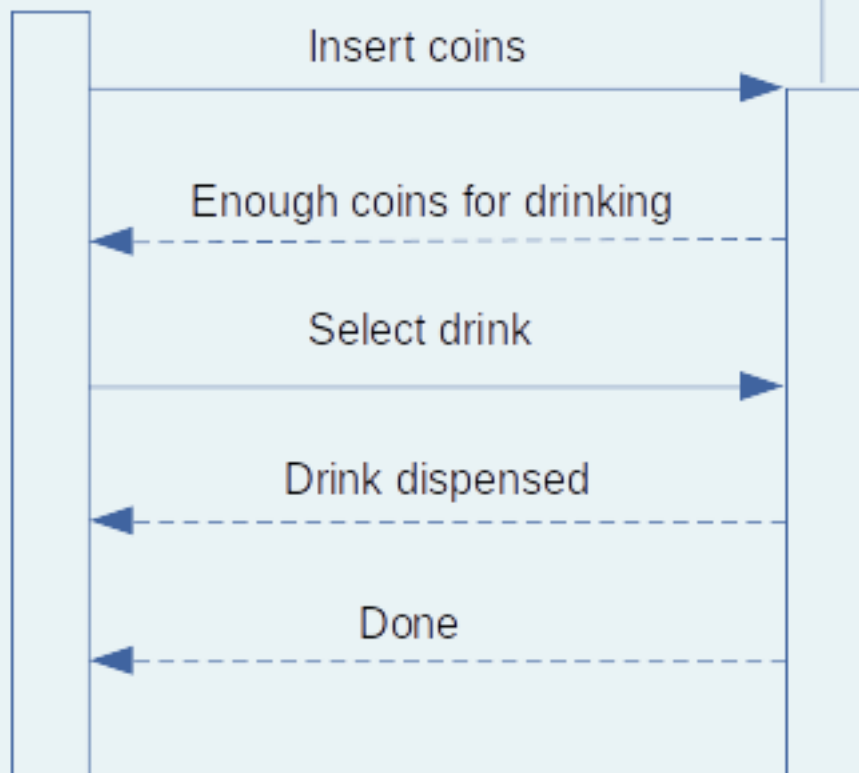
Done



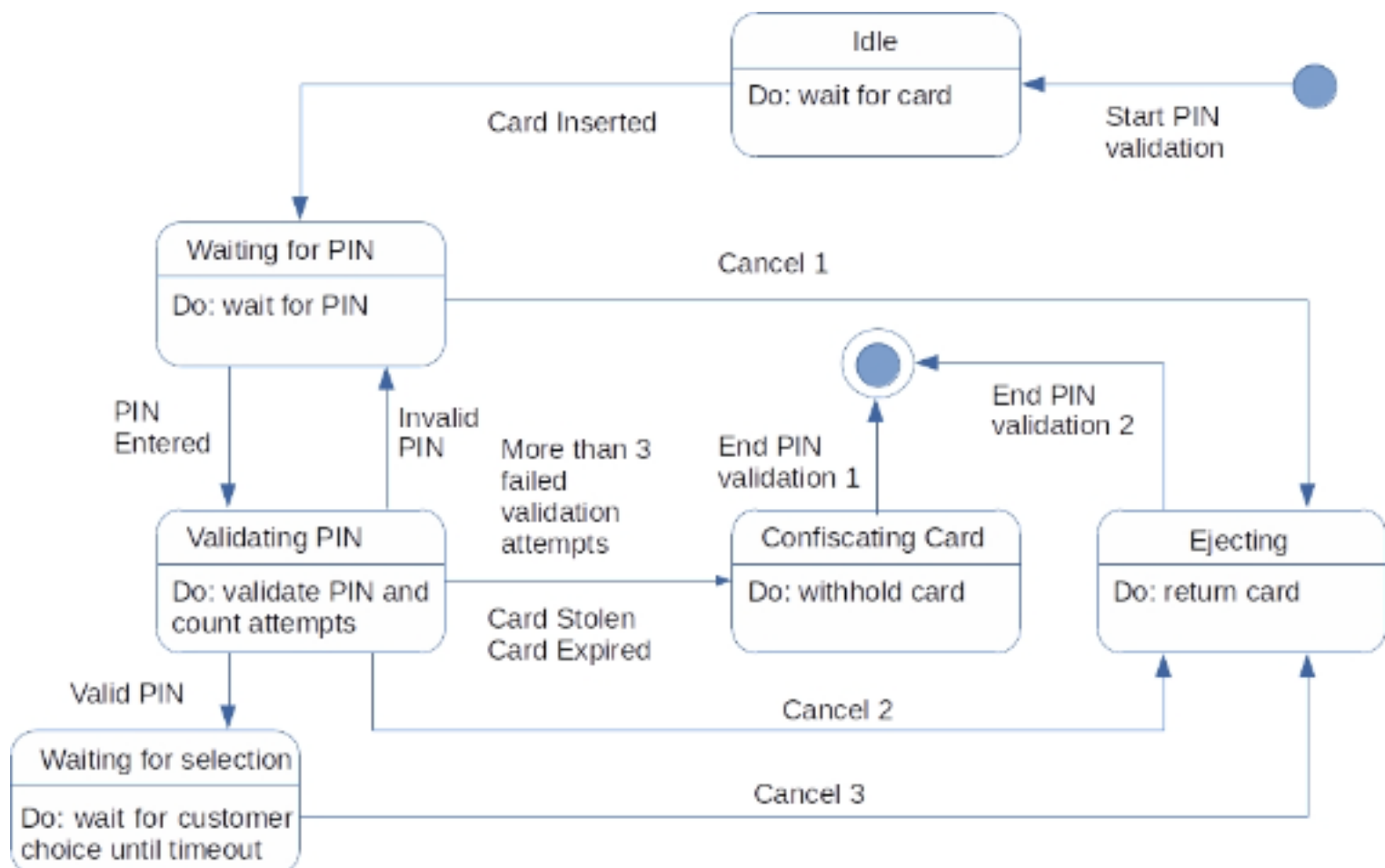
Customer



Vending Machine



Quale pattern architetturale meglio descrive l'architettura in figura ?



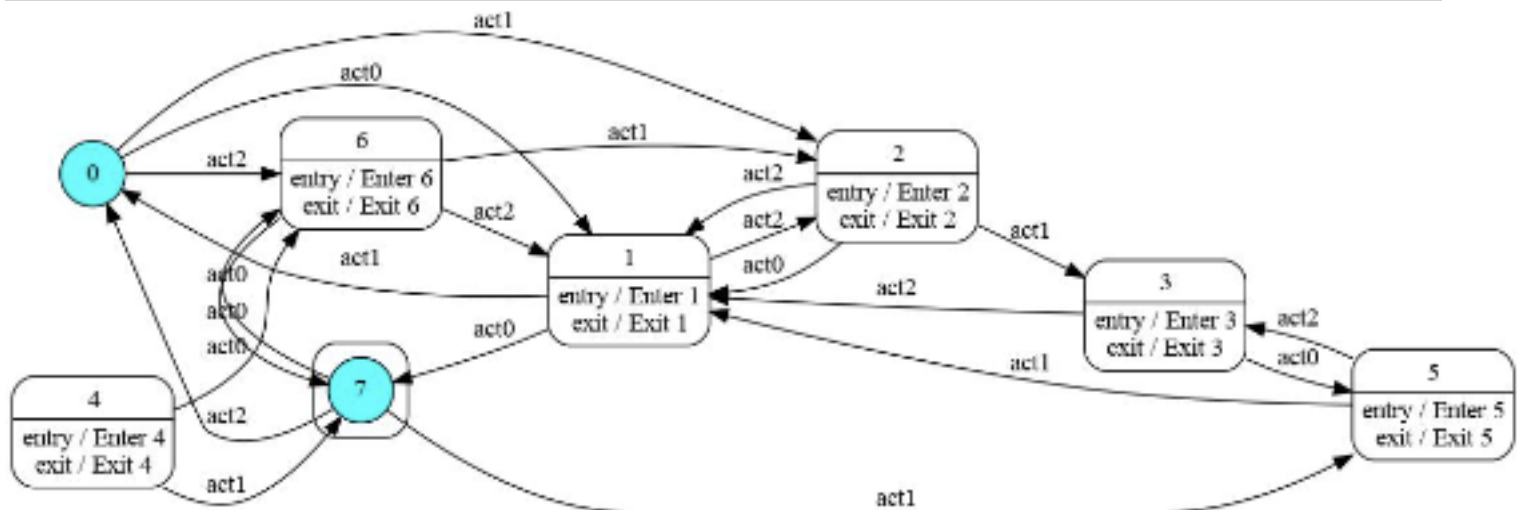
1.Pipe and filter architecture.

2.Model View Controller.

3.Layred architecture.

Risposta : 3

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; elseif (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; elseif (pre(state) == 0) and

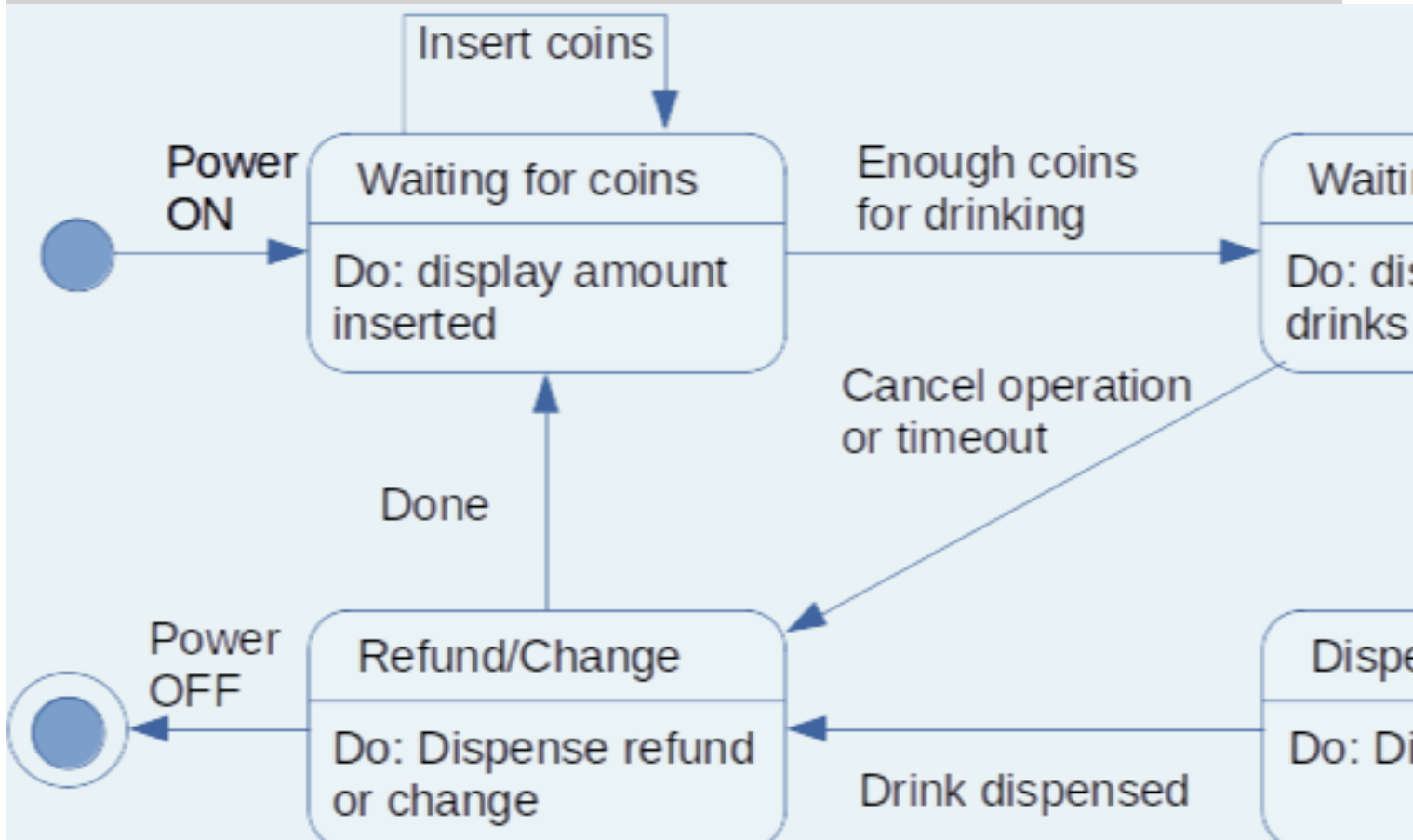

```

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

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

```

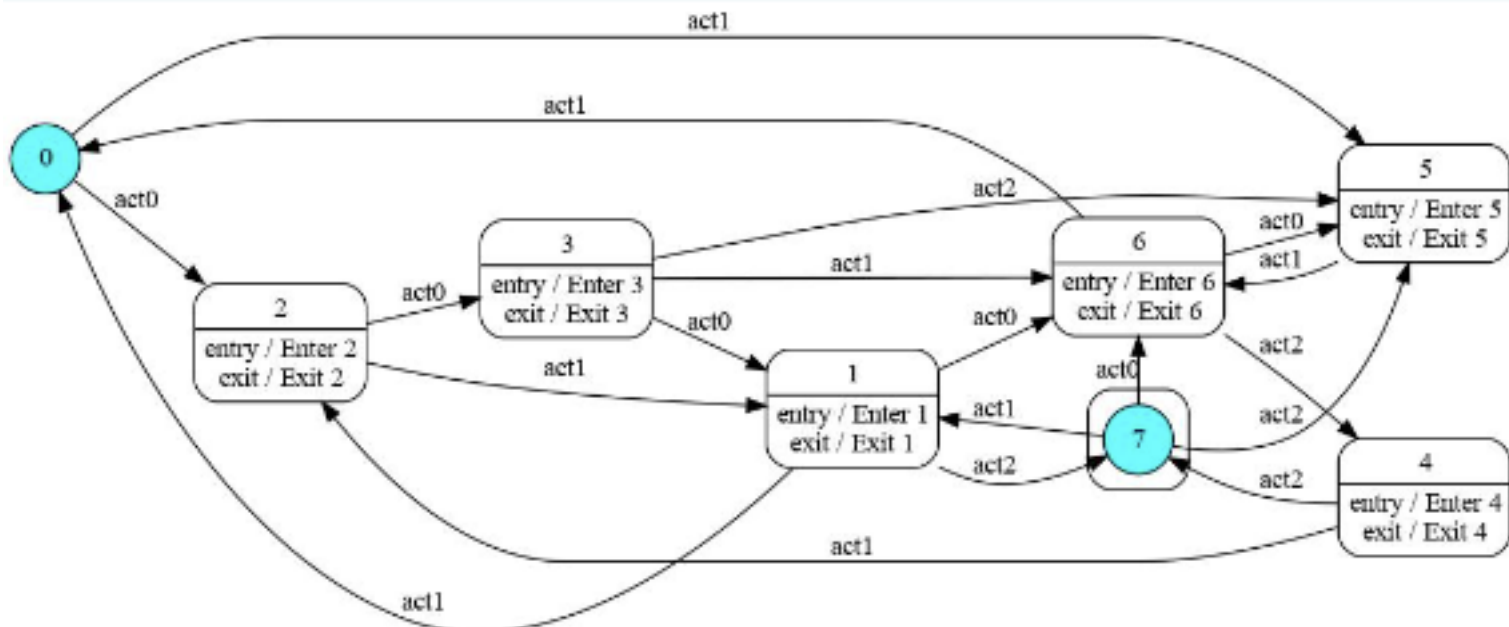
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?



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

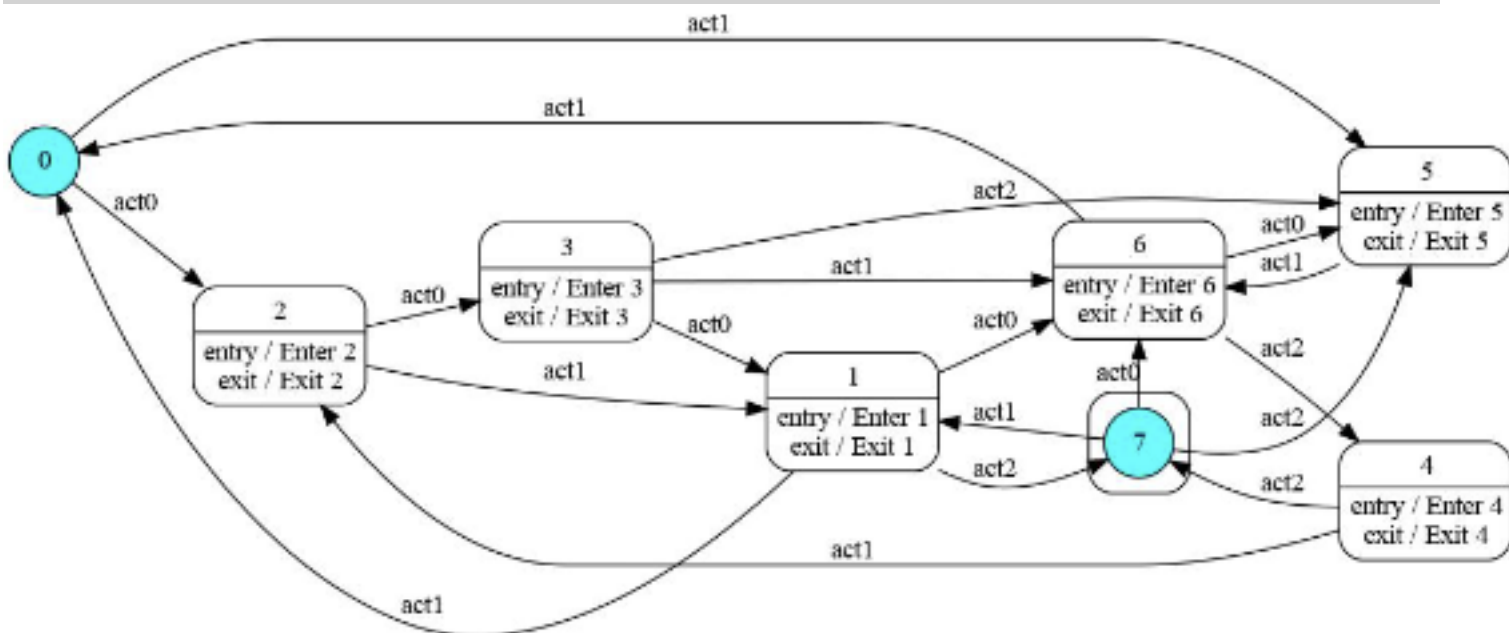
Quale delle seguenti frasi è corretta riguardo al Sequence Diagram in figura ?



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.

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 ?

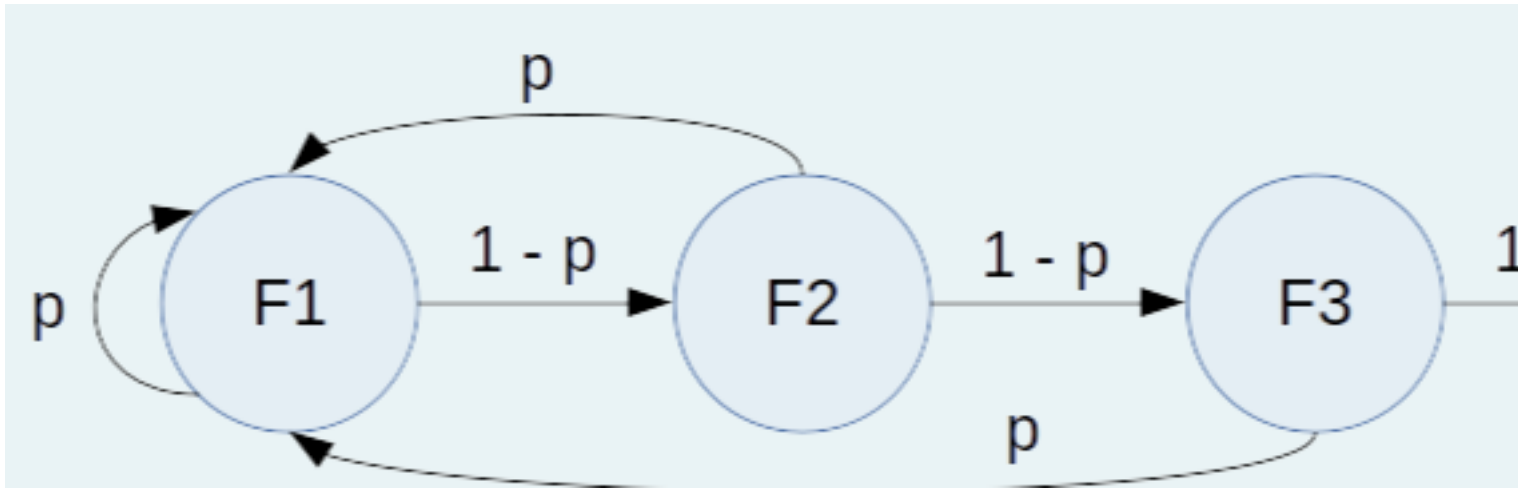


1. Una volta selezionata la bevanda non è possibile cancellare l'operazione.
2. La macchina non dà resto.
3. Una volta inserite monete per due bevande è possibile ottenerle senza reinserire le monete.

Risposta : 1

Si consideri la seguente Markov Chain:

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



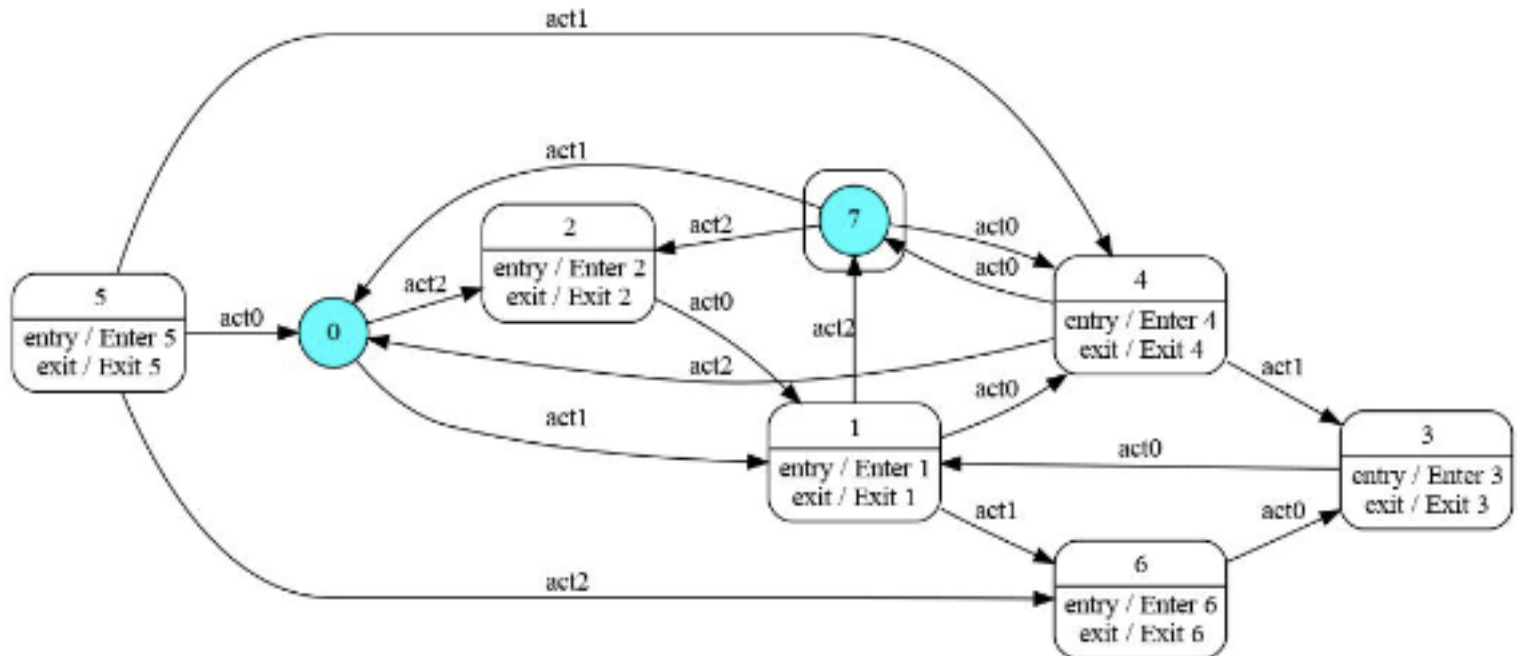
1.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 0, 1-p, 0; 0, p, 1-p, 0; 0, 0, 1-p, 0; 0, 0, 0, 1];Integer x; Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen 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] + 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; 0, p, 1-p, 0; 0, 0, 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] + 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; 0, p, 1-p, 0; 0, 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] + 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?



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] = [0, 1, 0, 0; 0, p, 1-p, 0; 0, 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] + 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] = [0, 1, 0, 0; 0, p, 1-p, 0; 0, 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] + A[x, F3]) then x := F3; else x := End;end if;end when;end System;

```
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] + 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] = [0, 1, 0, 0;p, 0, 0, 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] + A[x, F3]) then x := F3; else x := End;end if;end when;end System;
```

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

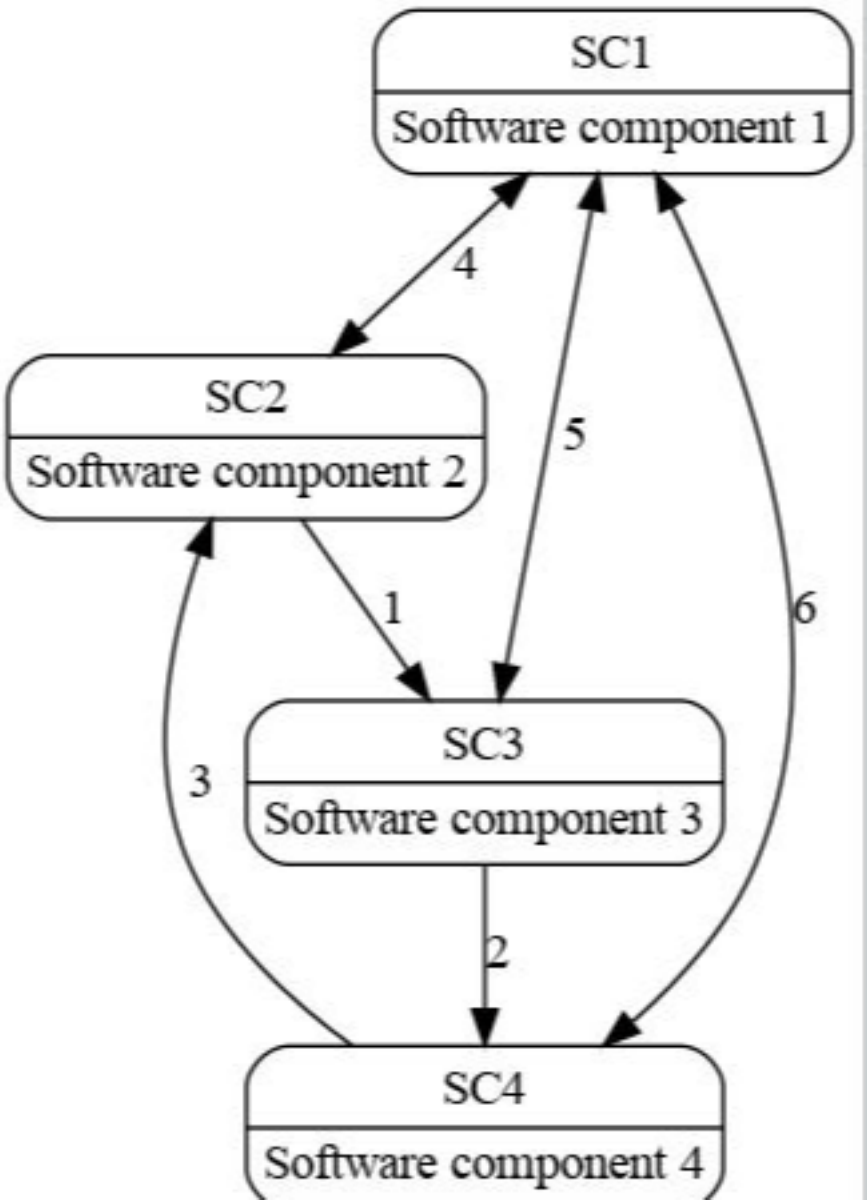
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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

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

Test case 3: act1 act1 act2 act0 act1 act0

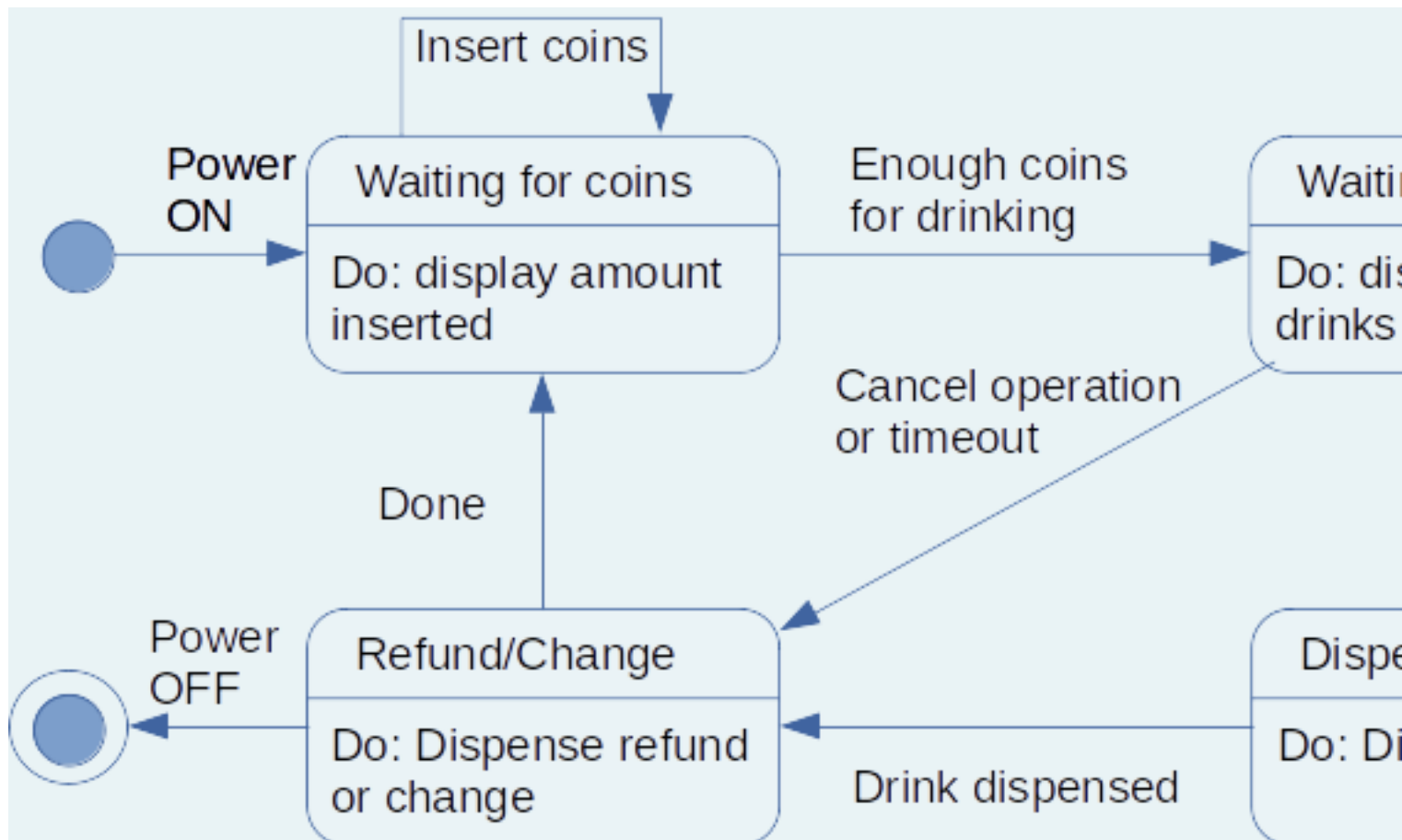
Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



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

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

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

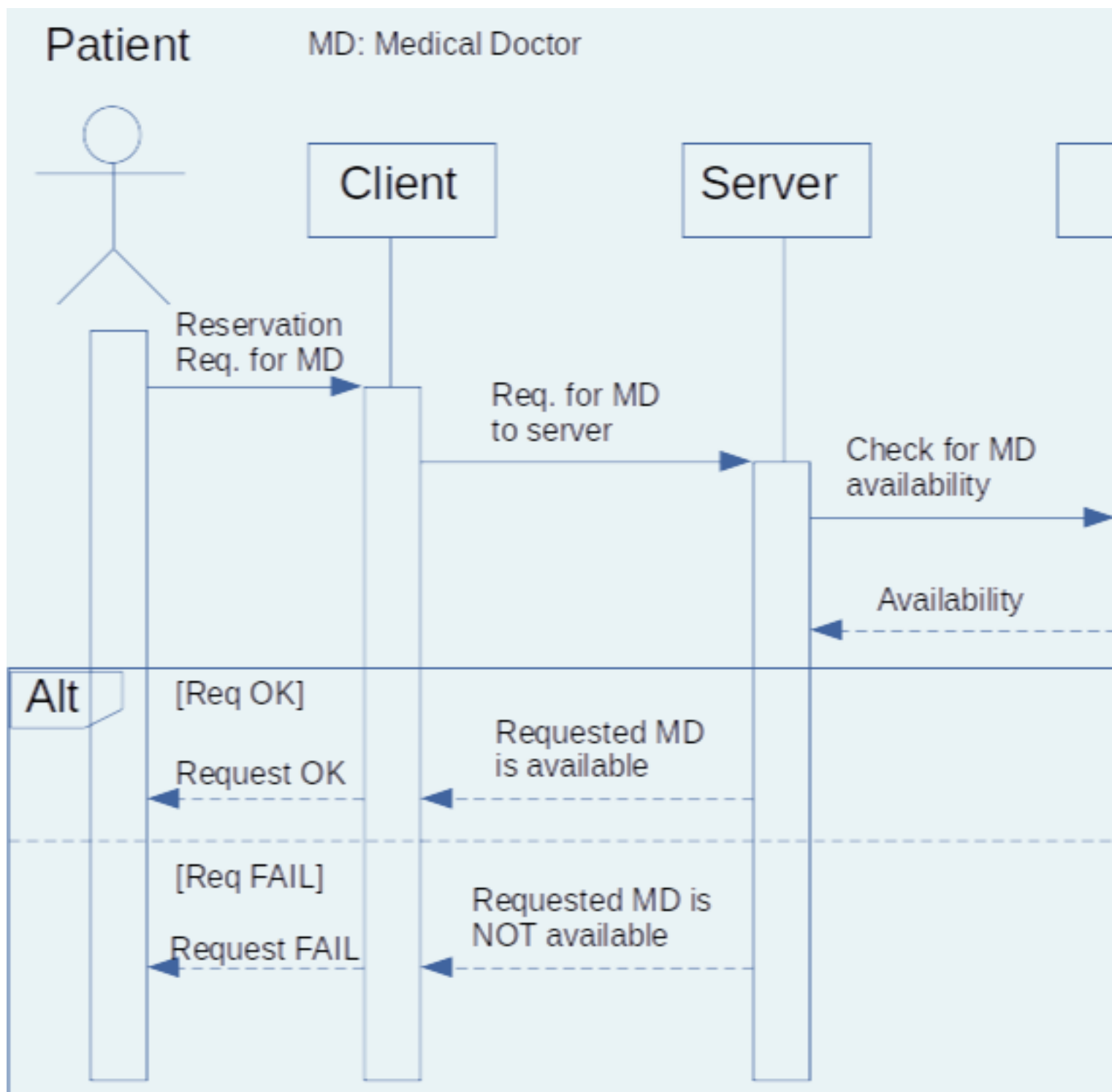
Risposta : 3

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

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

Uno scenario è una sequenza di stati.

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



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.

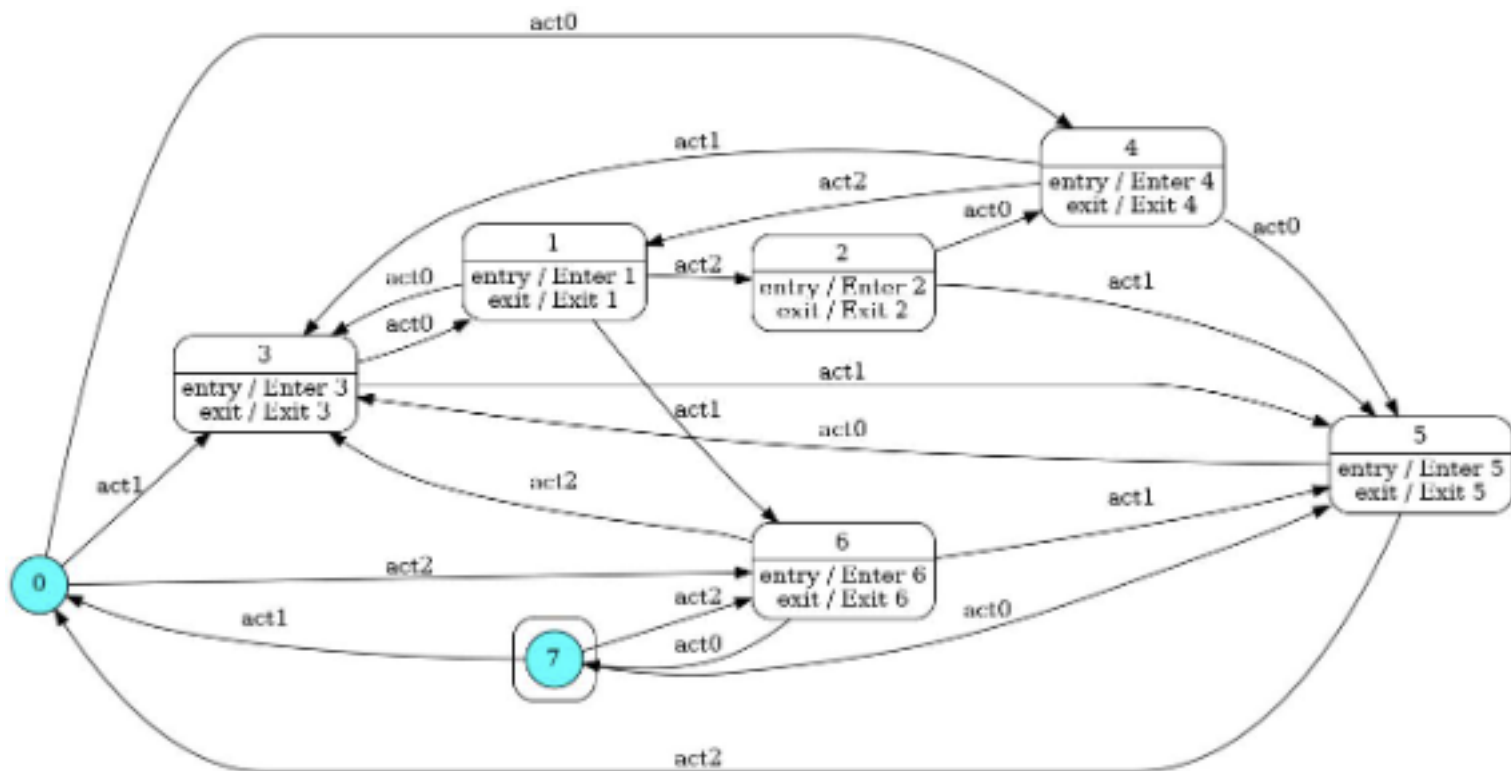
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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?



- 1.100%
- 2.35%
- 3.50%

Risposta : 3

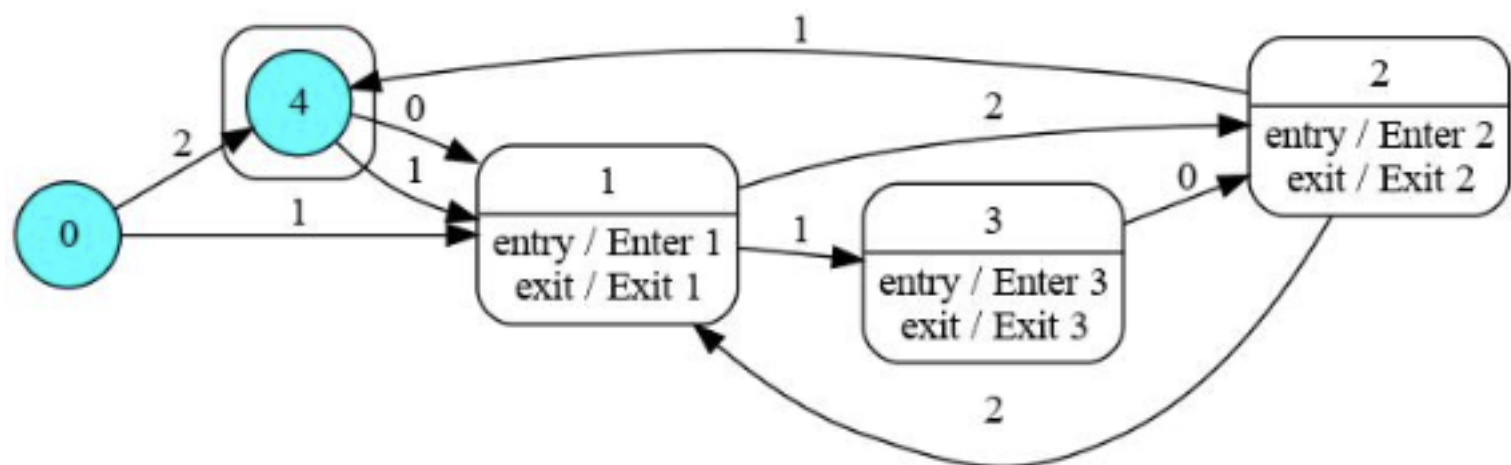
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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

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

Test case 2: act0

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



- 1.50%
- 2.90%
- 3.100%

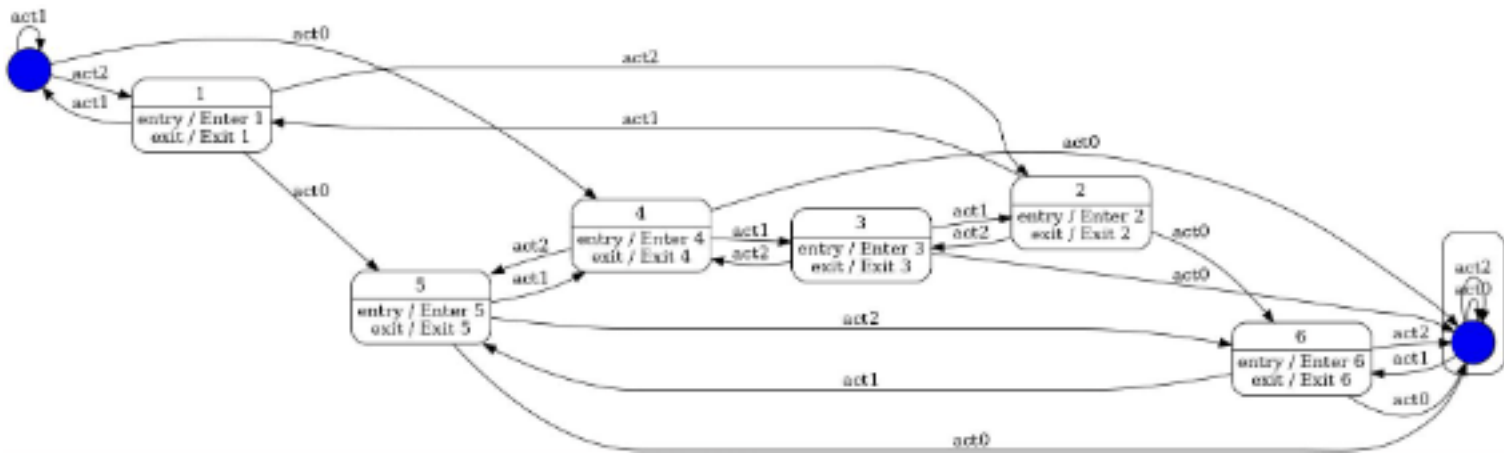
Risposta : 2

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 act1 act2 act1 act2 act2 act2

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



1.60%

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.

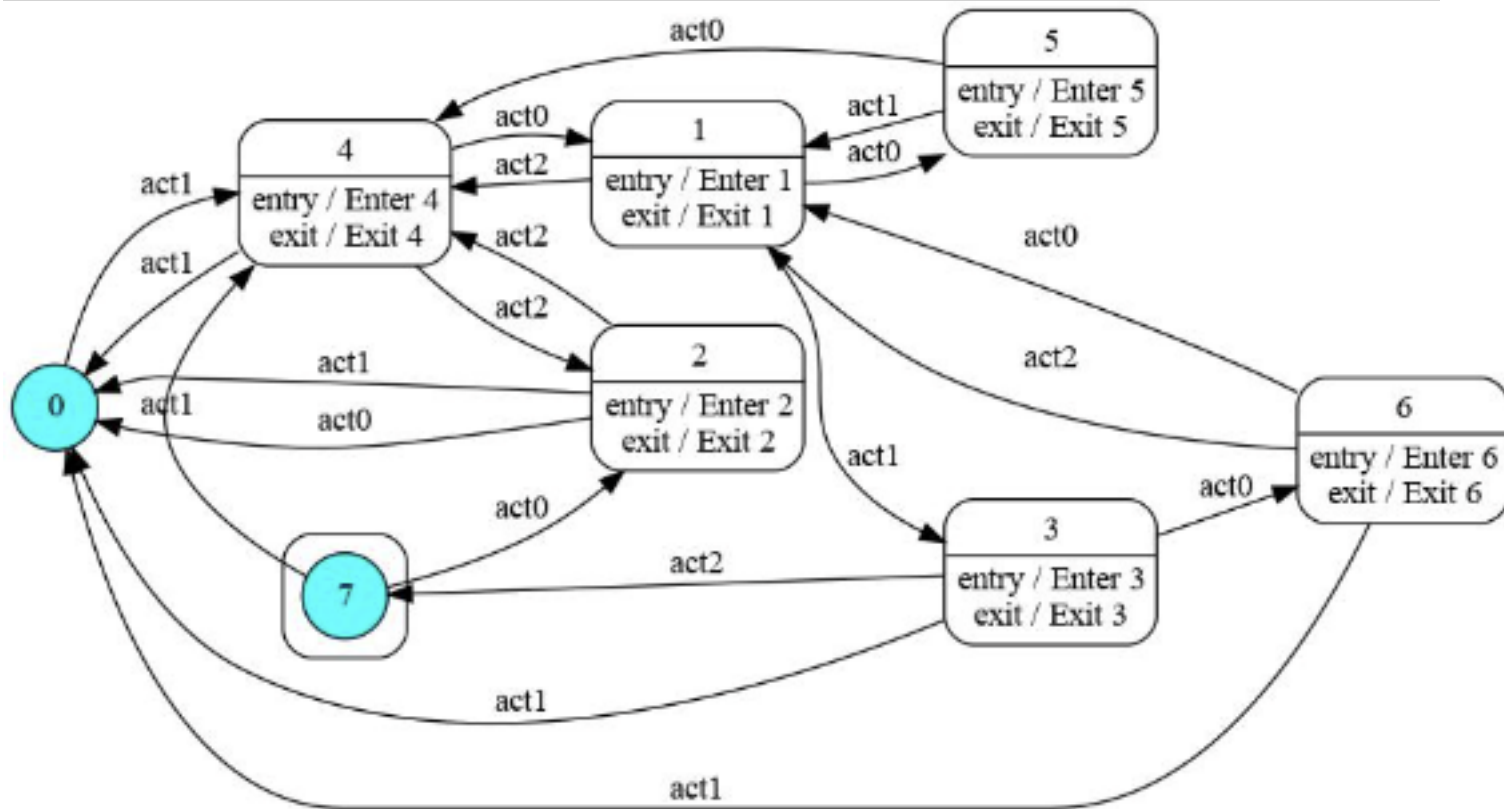
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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

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

Test case 3: act1 act1

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



1.State coverage: 60%

2.State coverage: 85%

3.State coverage: 100%

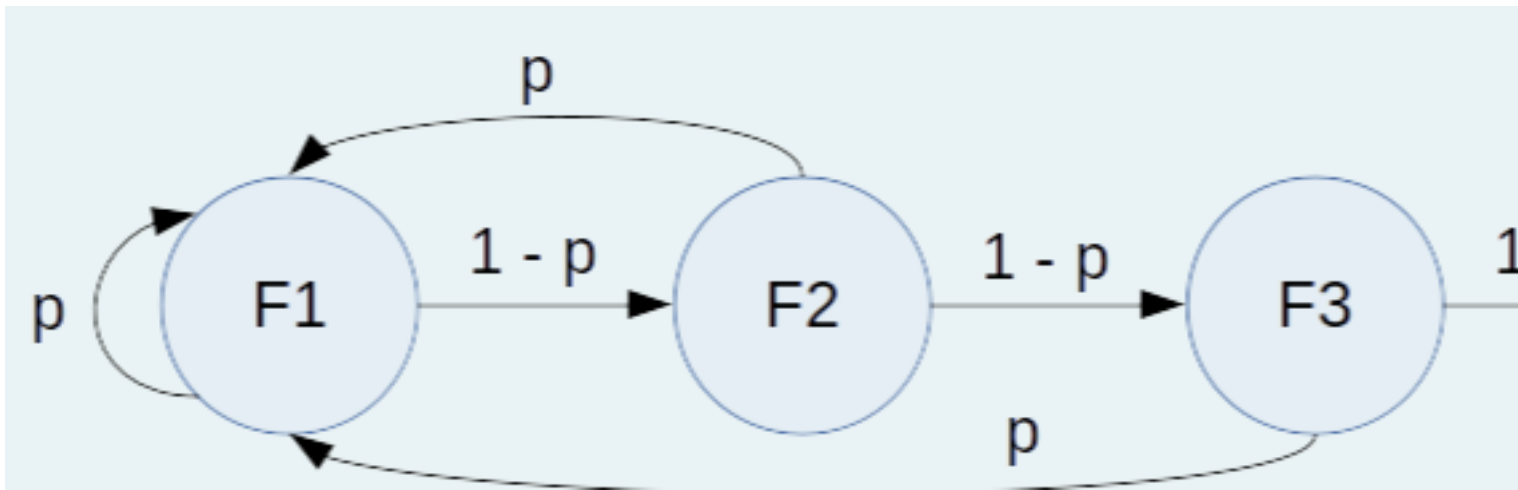
Risposta : 2

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

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

Uno scenario è una sequenza di stati.

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



1.0.08

2.0.12

3.0.32

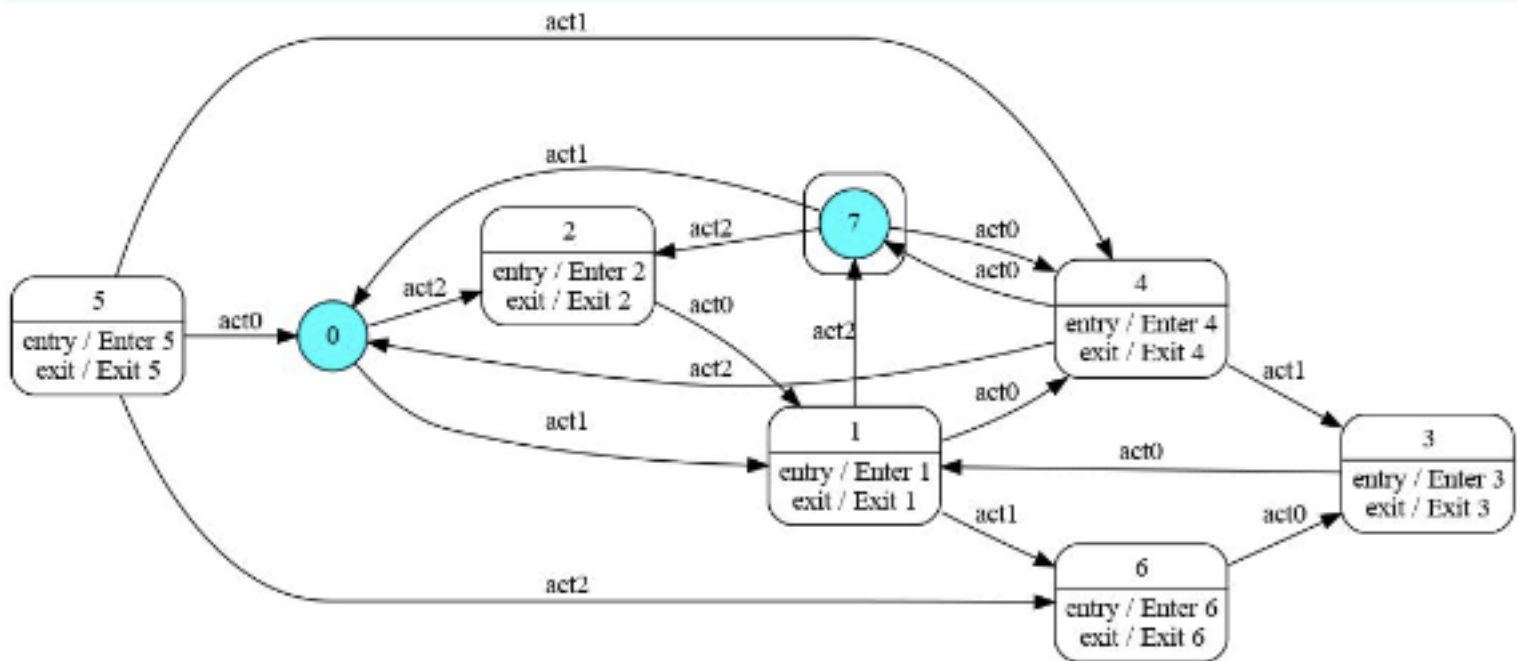
Risposta : 2

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

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

Uno scenario è una sequenza di stati.

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



1.0.12

2.0.28

3.0.42

Risposta : 2

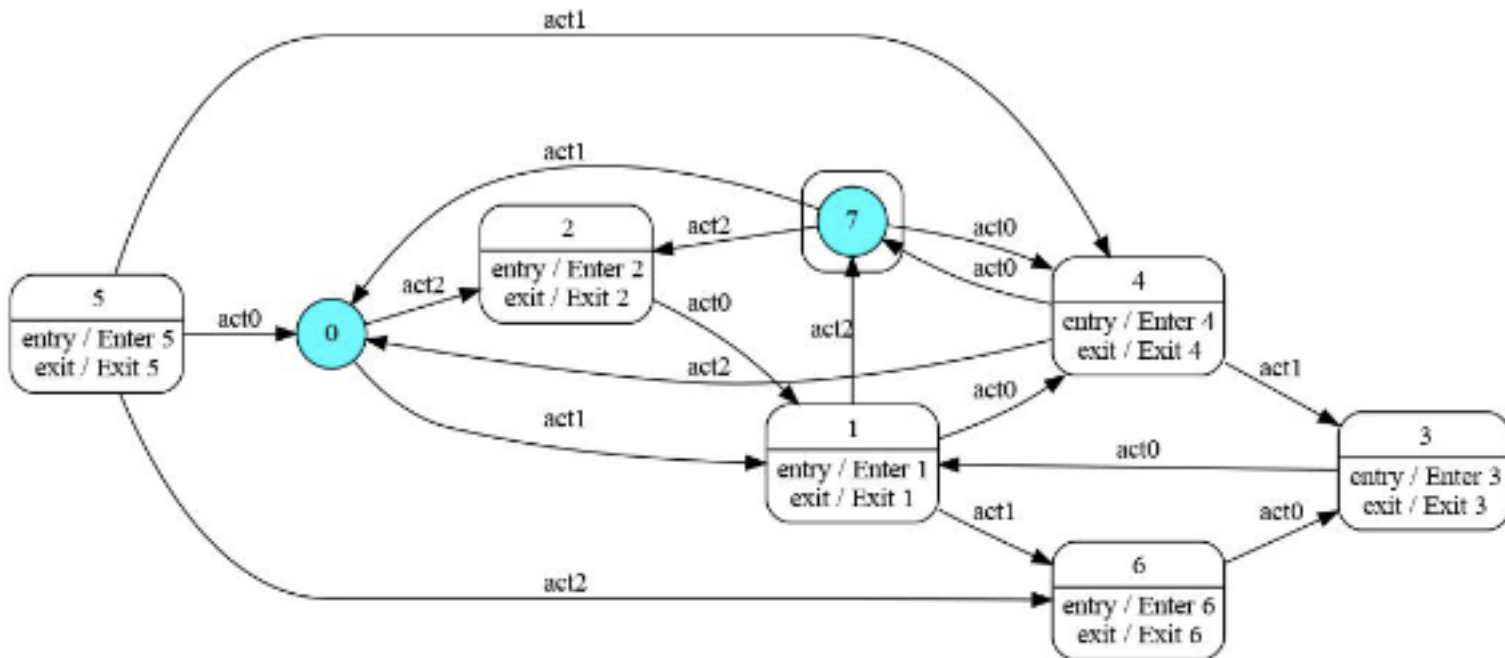
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il costo dello stato (fase) x è $c(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappresenta il completamento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

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

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

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

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



1. $c(0) \cdot (1 - p) / p$
2. $c(0) / (1 - p)$
3. $c(0) / (p \cdot (1 - p))$

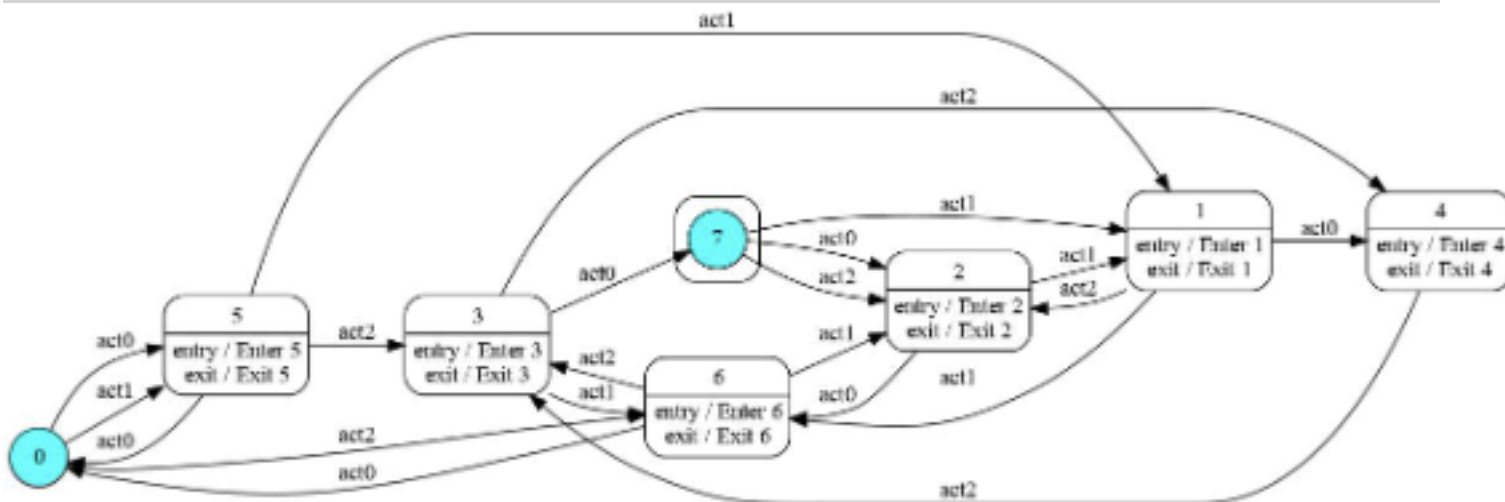
Risposta : 2

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

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

Uno scenario è una sequenza di stati.

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



1. 0.07
2. 0.27
3. 0.03

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.

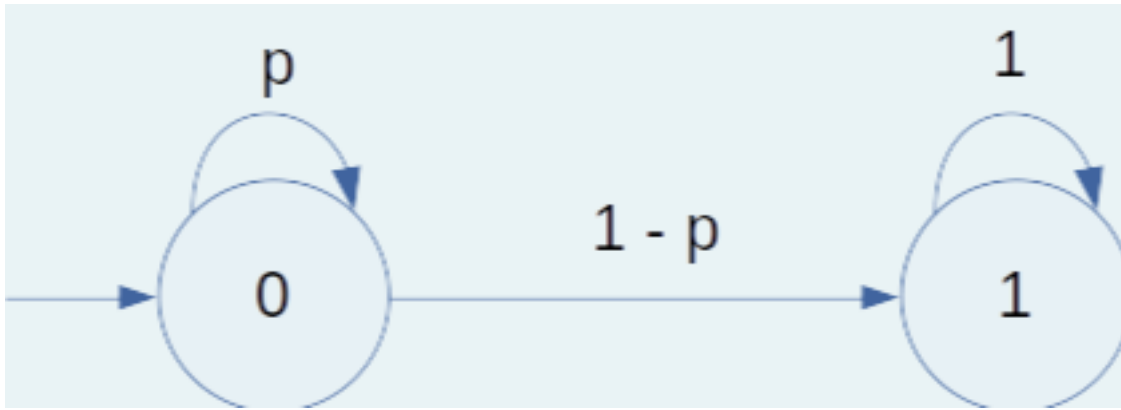
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act0

Test case 2: act1 act0 act2 act2 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?



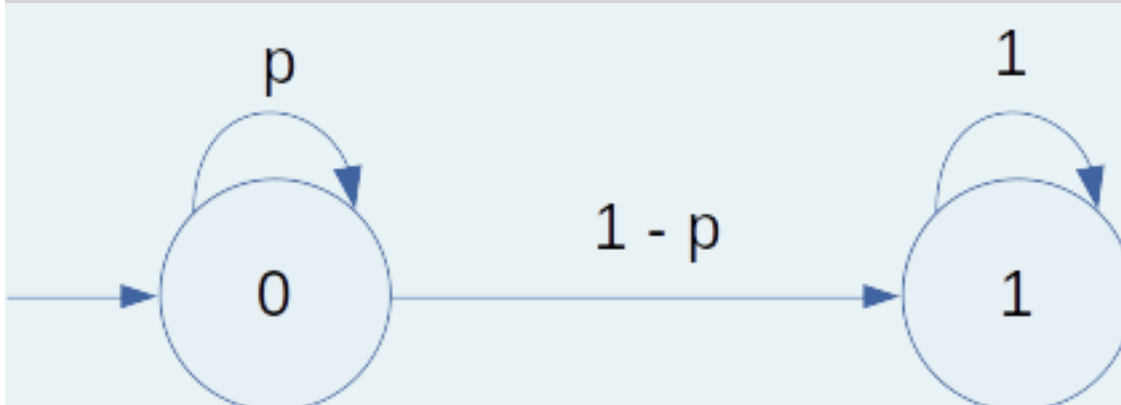
1.60%

2.90%

3.75%

Risposta : 1

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



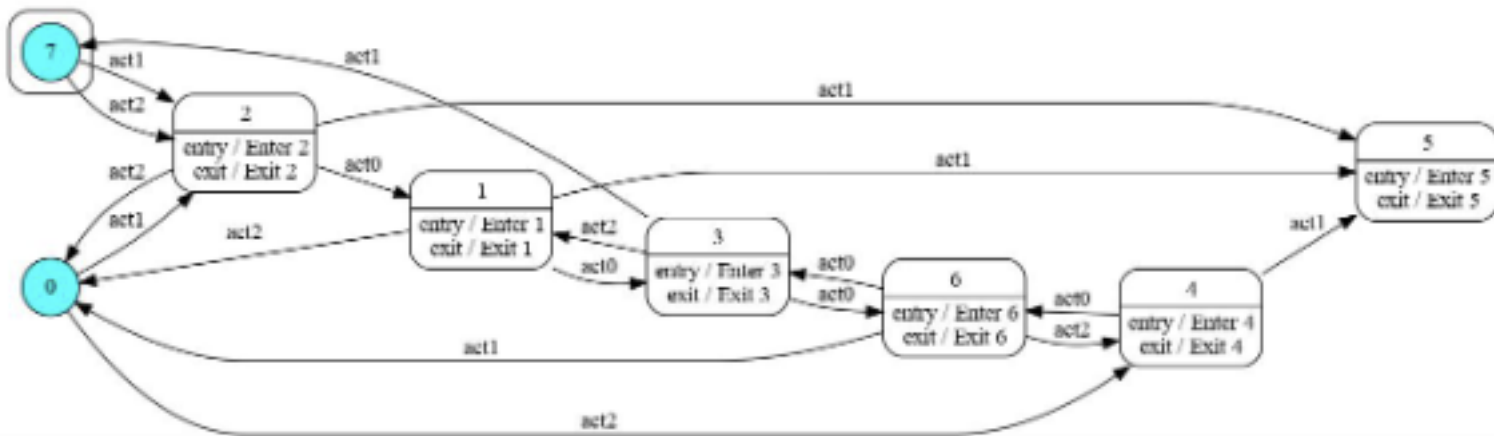
1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif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

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

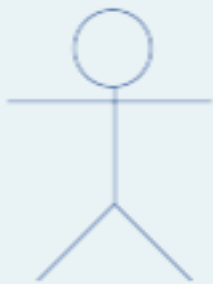
3.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() thenx := 0;elsewhen sample(0,T) thenif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

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 ?



Customer



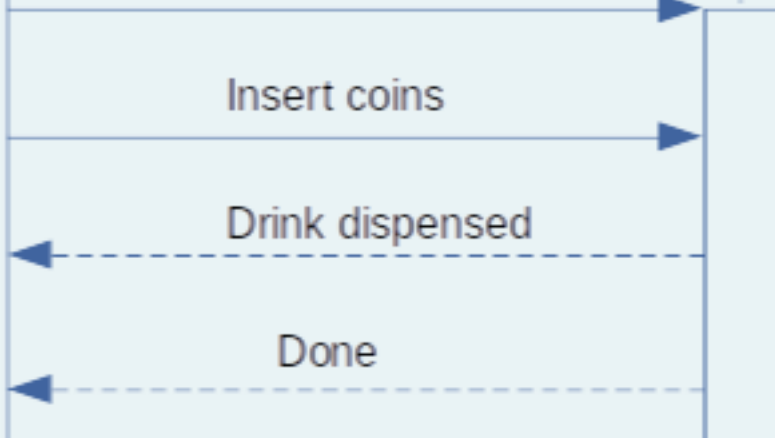
Vending Machine

Select drink

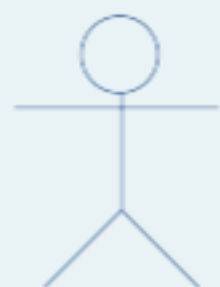
Insert coins

Drink dispensed

Done



Customer



Vending
Machine

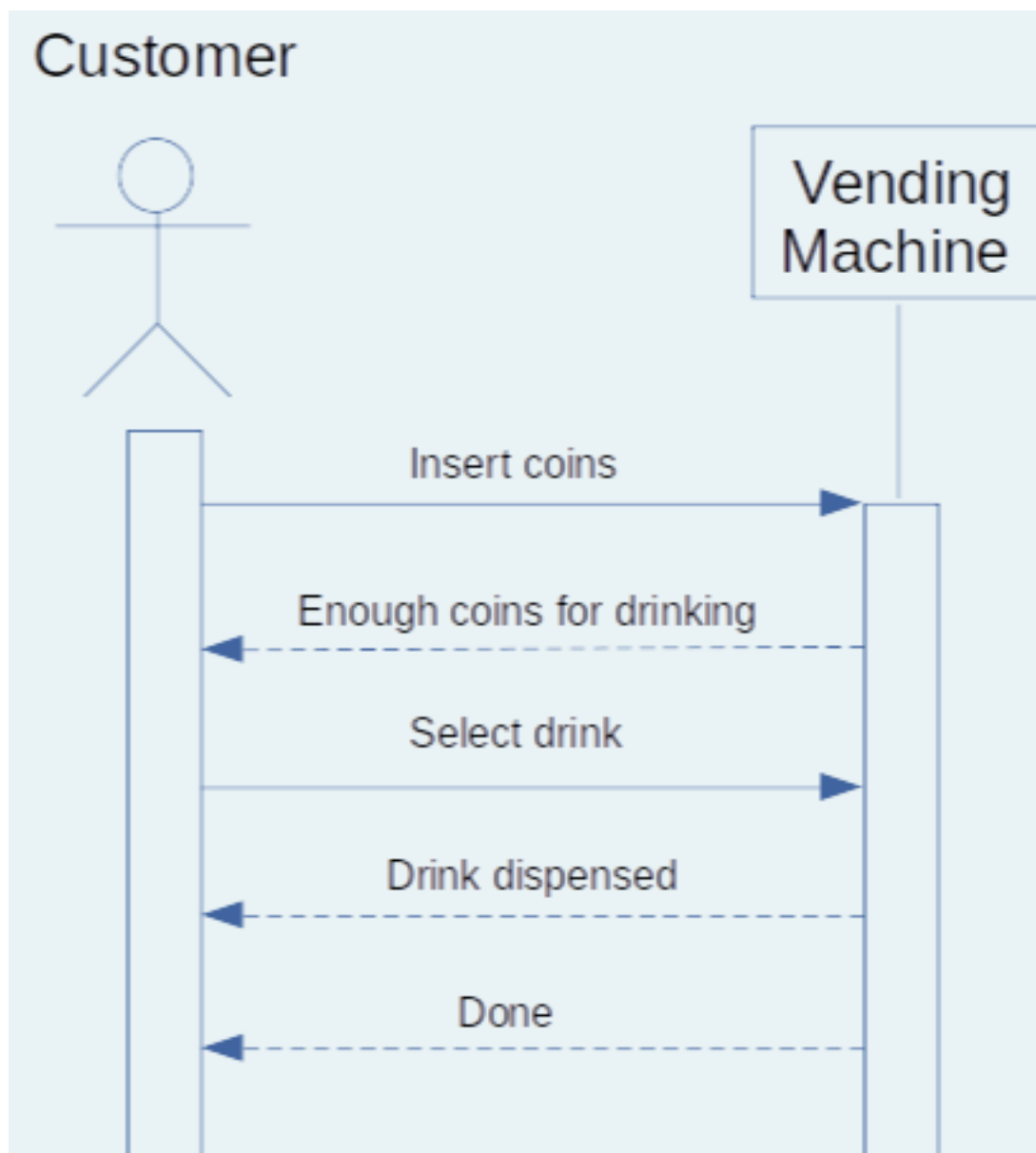
Insert coins

Enough coins for drinking

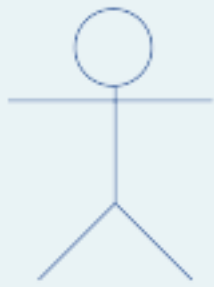
Select drink

Drink dispensed

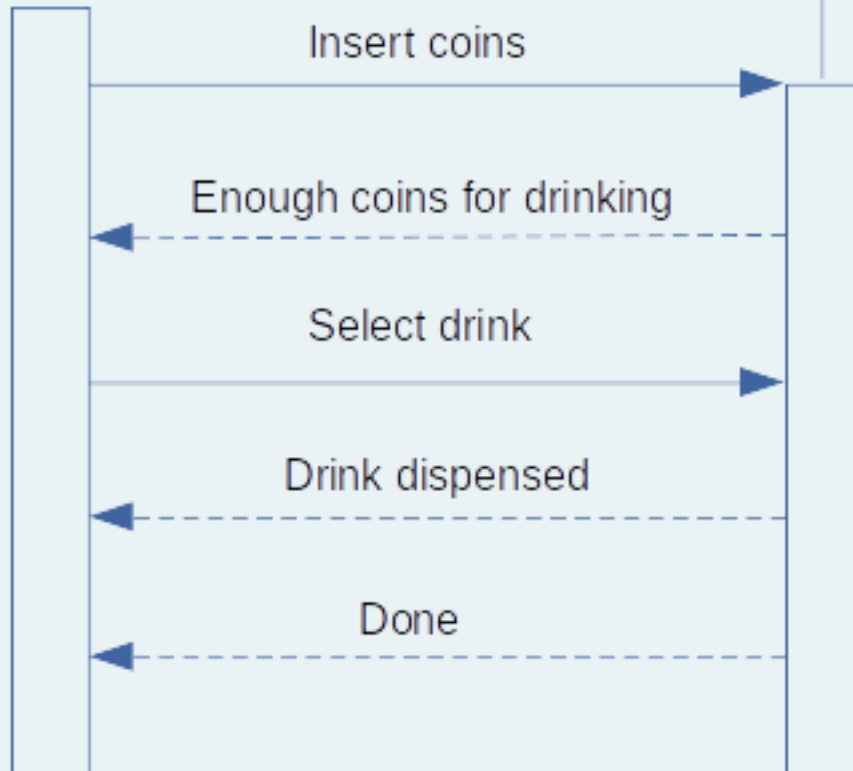
Done



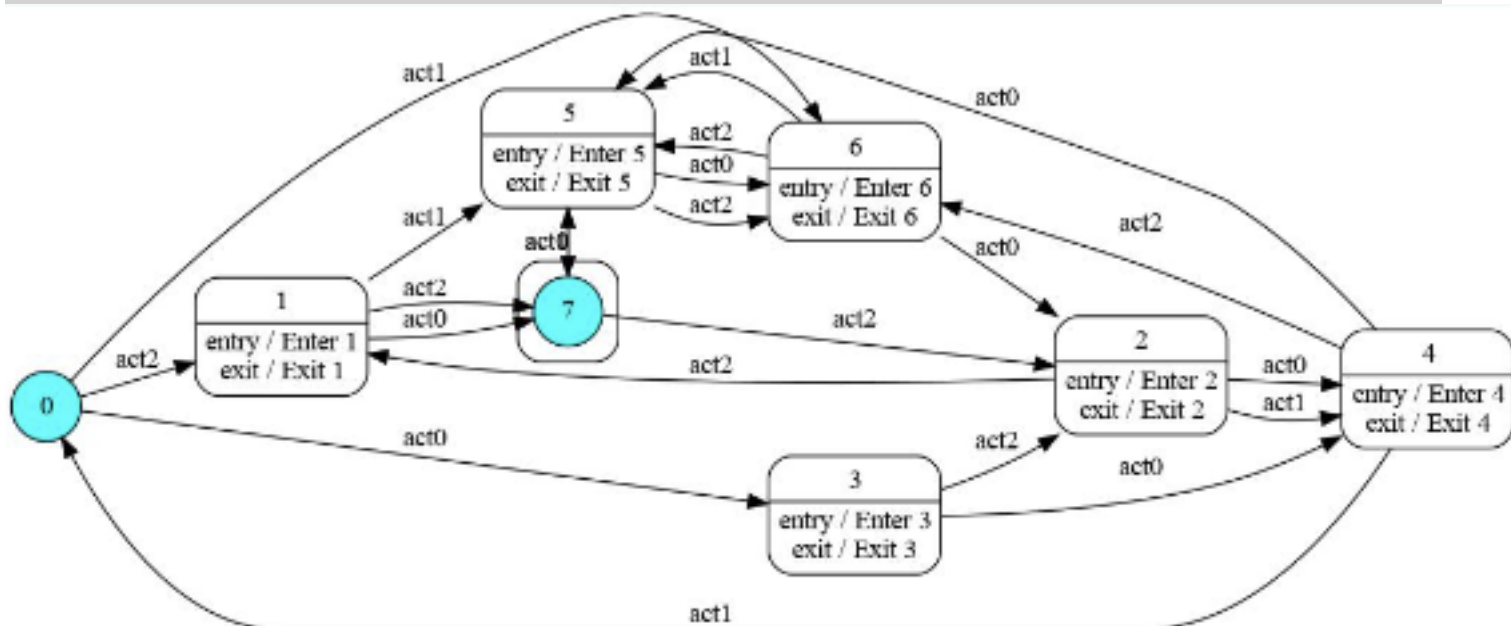
Customer



Vending Machine



Quale delle seguenti frasi è corretta riguardo all'activity 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.

Risposta : 3

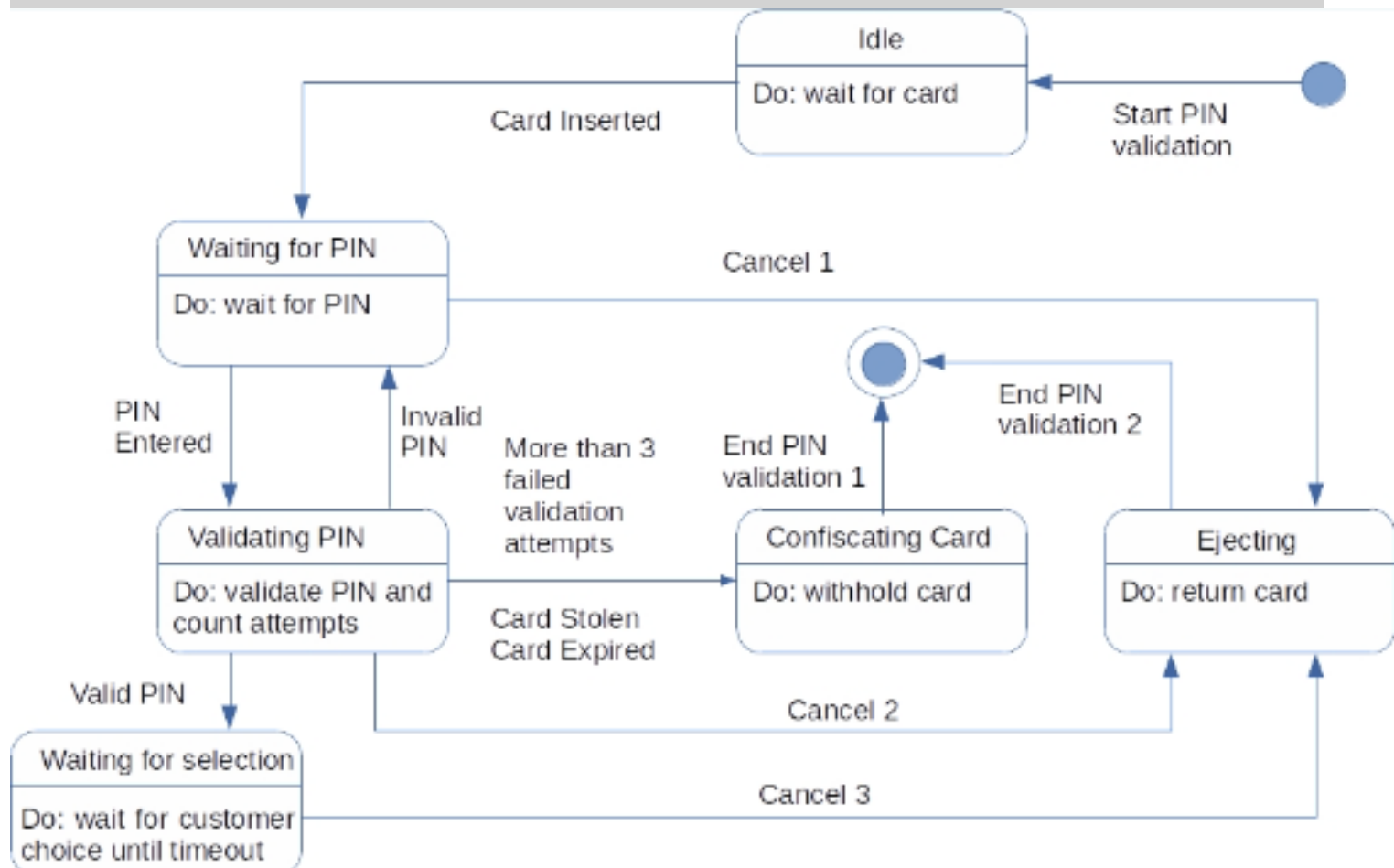
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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 act0 act1 act1 act2 act1 act2 act0 act0 act0 act2

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



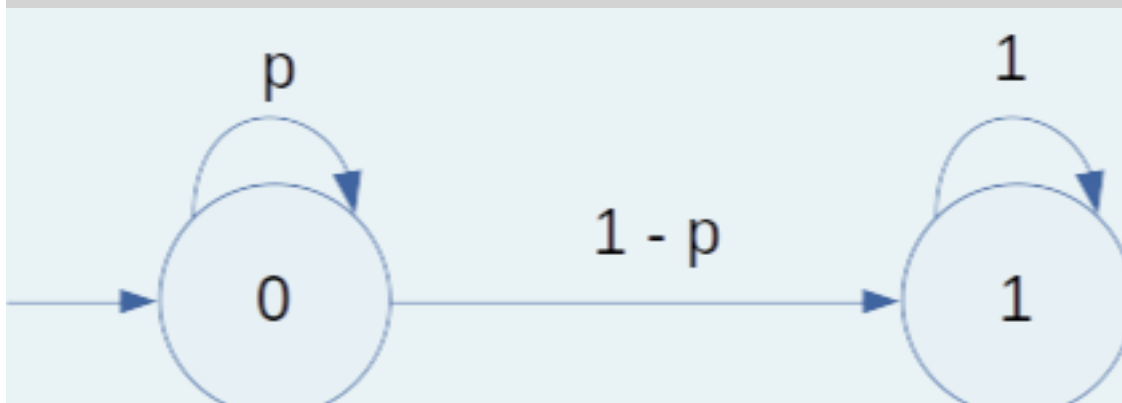
1.State coverage: 60%

2.State coverage: 75%

3.State coverage: 90%

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

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

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

Risposta : 3

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

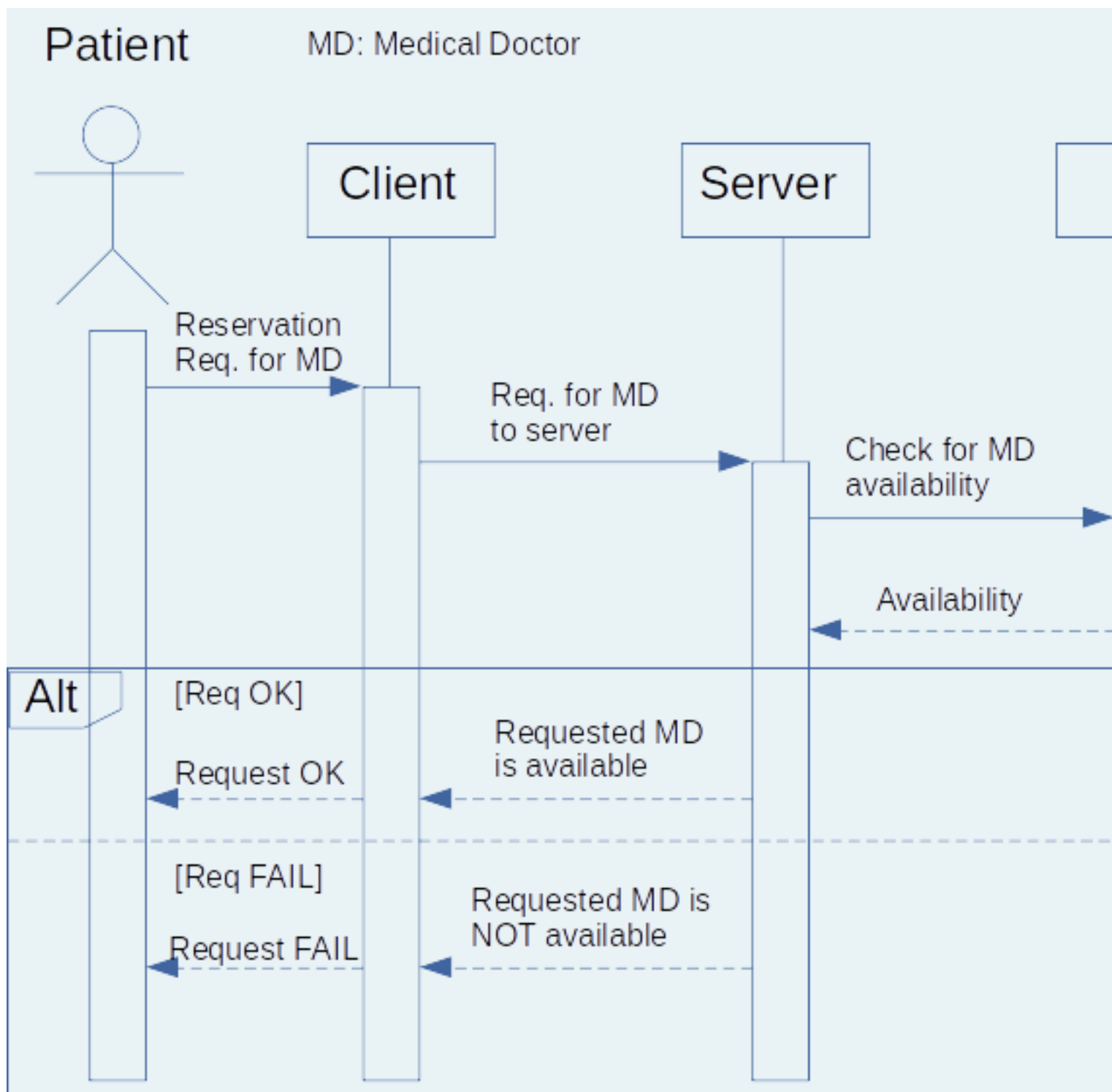
Test case 3: act2 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



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

Risposta : 3

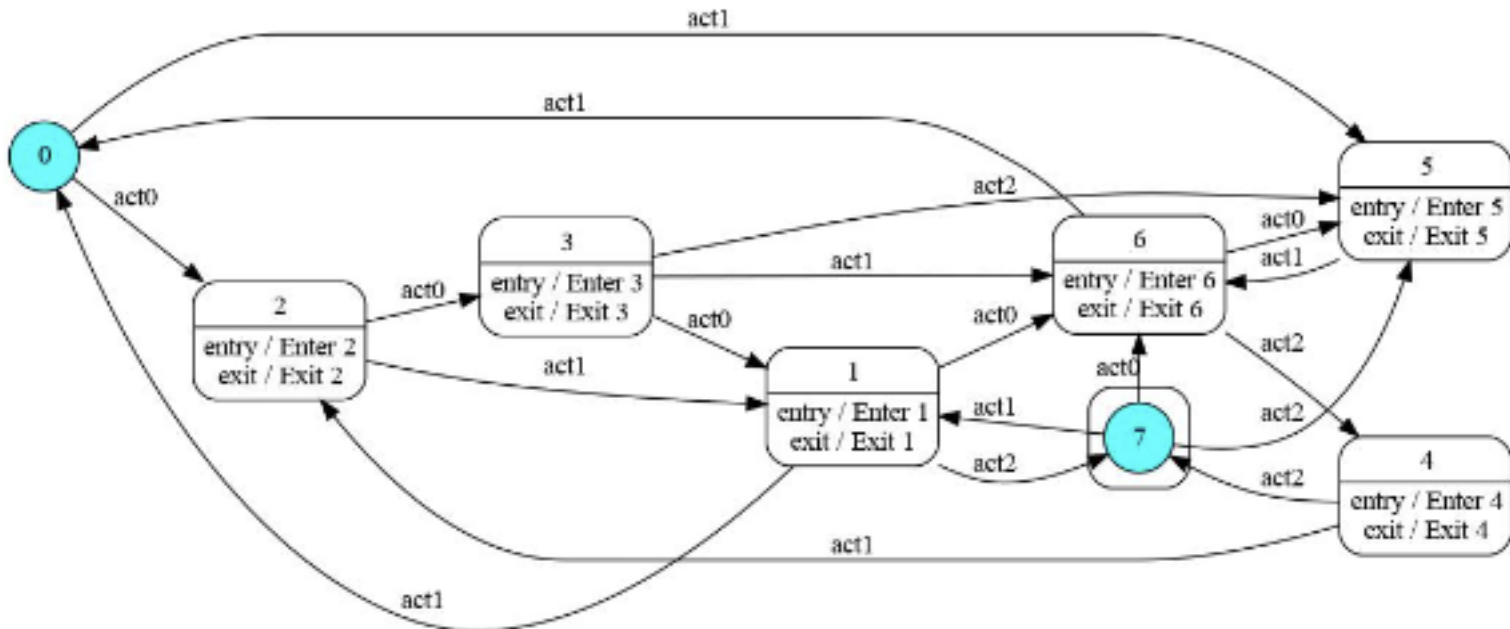
Quale delle seguenti frasi è corretta riguardo al Sequence Diagram in figura ?



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.
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.

Risposta : 1

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



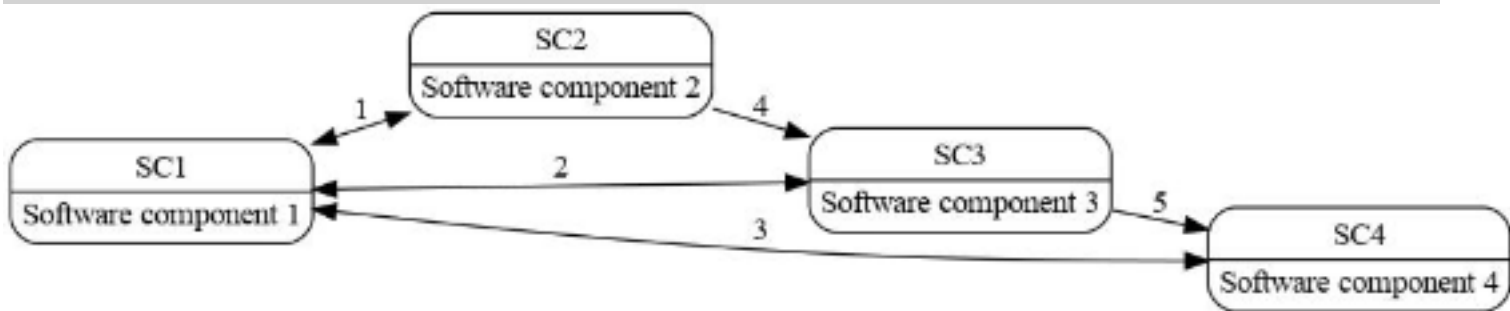
1.block FSA // Finite State Automaton/* connector declarations outside this block:connector InputInteger = input Integer;connector OutputInteger = output Integer;*/InputInteger u; // external inputOutputInteger x; // stateparameter Real T = 1;algorithmwhen initial() then x := 0;elsewhen sample(0,T) then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

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

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

Risposta : 2

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. $1/(p \cdot (1 - p))$
3. $(1 - p)/p$

Risposta : 1

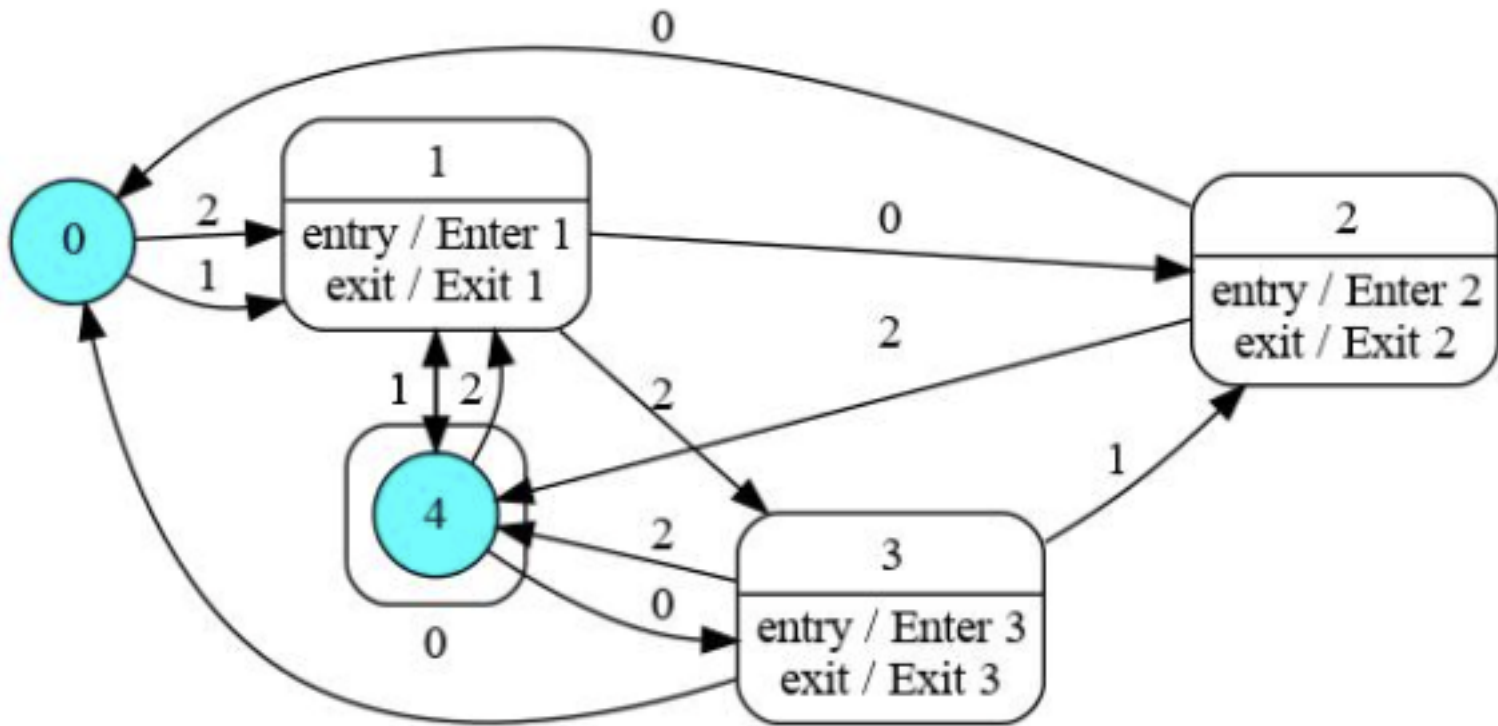
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il tempo necessario per completare la fase x è $\text{time}(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase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

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

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

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

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



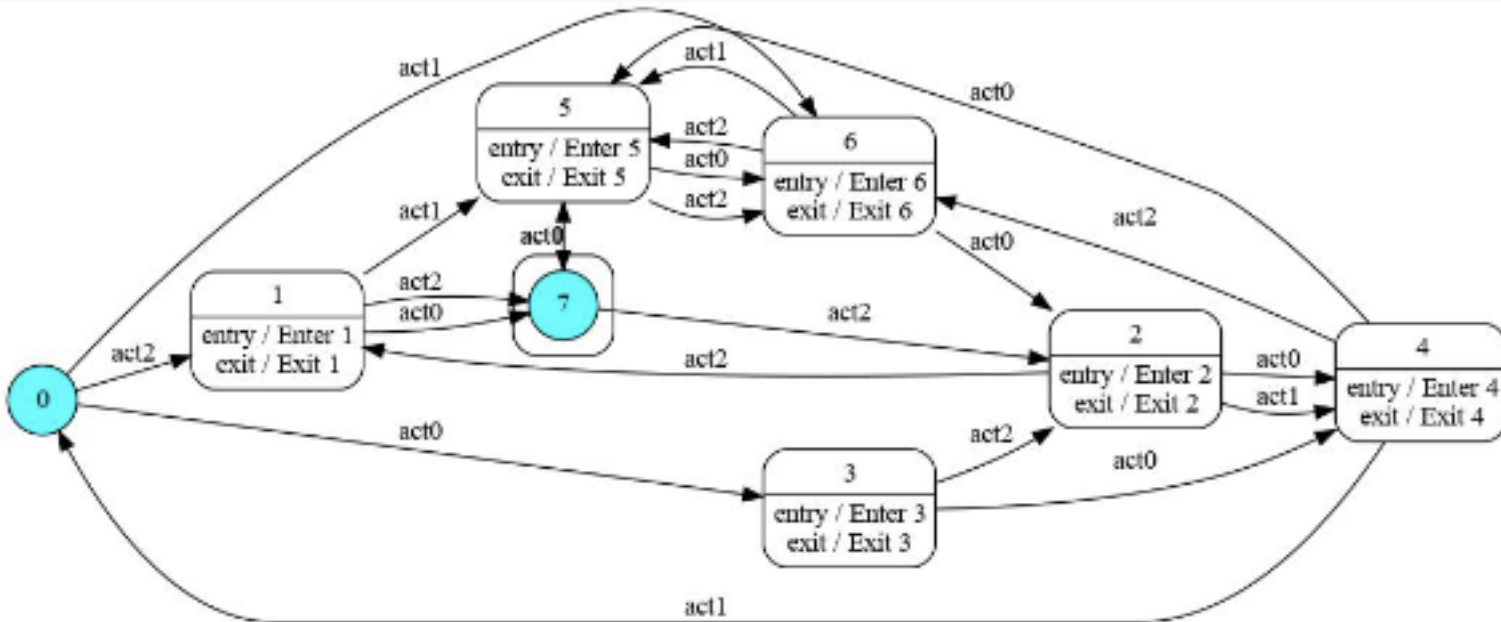
1. $\text{time}(0)/(p \cdot (1 - p))$

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

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

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.

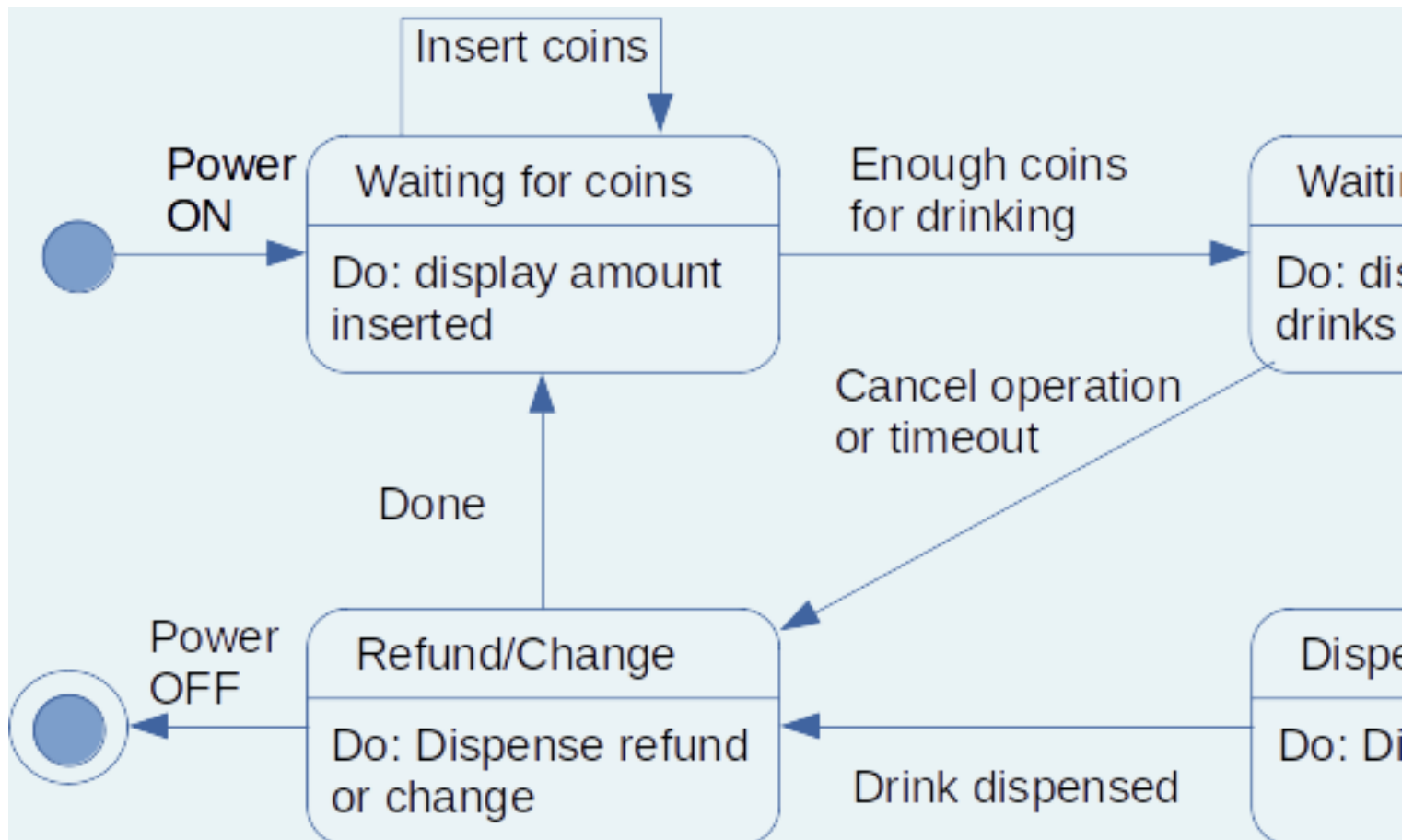
2. Anche se ci sono abbastanza monete potrebbe non essere possibile ottenere la bevanda selezionata.

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?



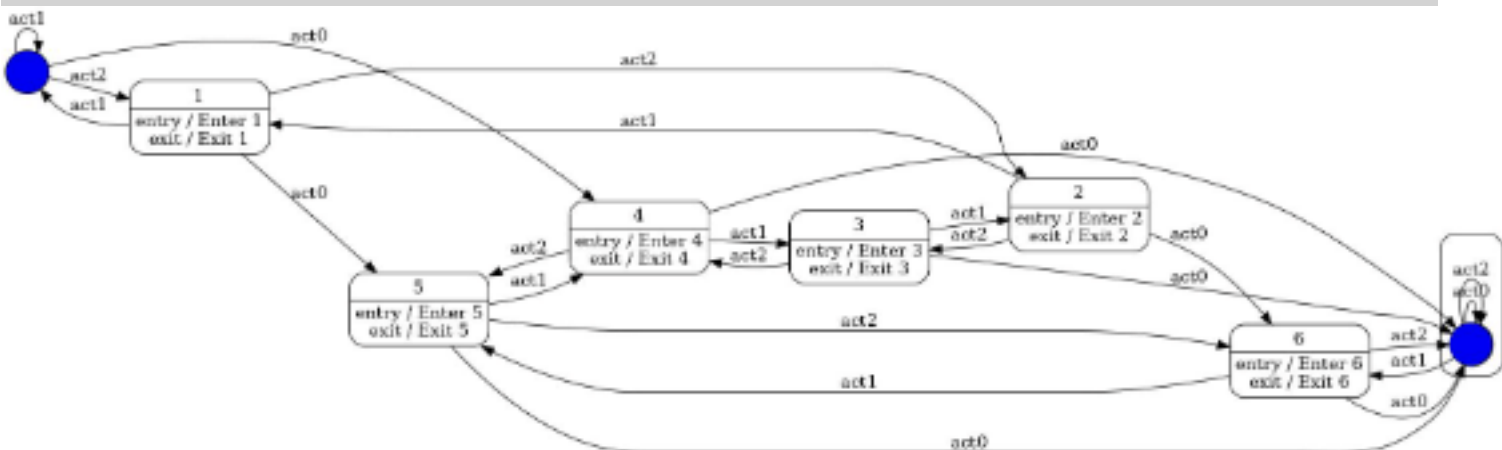
```
1.model System
parameter Integer F1 = 1;
parameter Integer F2 = 2;
parameter Integer F3 = 3;
parameter Integer End = 4;
parameter Real p = 0.3;
parameter Real A[4, 4] = [0, 1, 0, 0; p, 0, 0, 1-p; 0, 0, p, 0; 1-p, 0, 0, 1];
Integer x;
Real r1024;
Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];
algorithm
when 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;
```

```
2.model System
parameter 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; 1-p, 0, 0, 1];
Integer x;
Real r1024;
Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];
algorithm
when 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;
```

```
3.model System
parameter 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; 1-p, 0, 0, 1];
Integer x;
Real r1024;
Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];
algorithm
when 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?



```

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, 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;

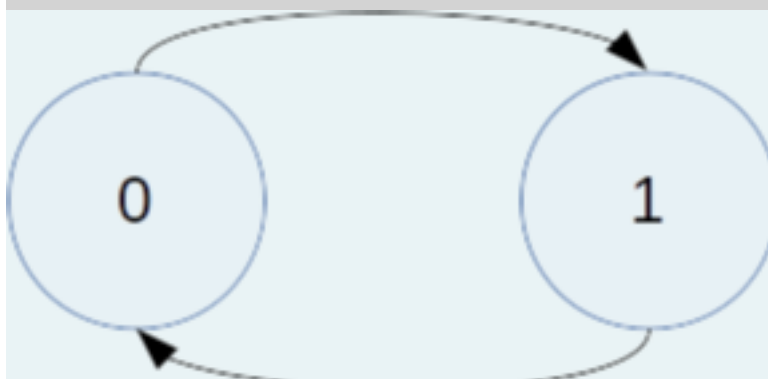
2.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 0, 1-p, 0; p, 1-p, 0, 0; p, 0, 0, 1-p; 0, 0, 0, 1];Integer x; Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen 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] + 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, 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] + A[x, F3]) then x := F3; else x := End;end if;end when;end System;

```

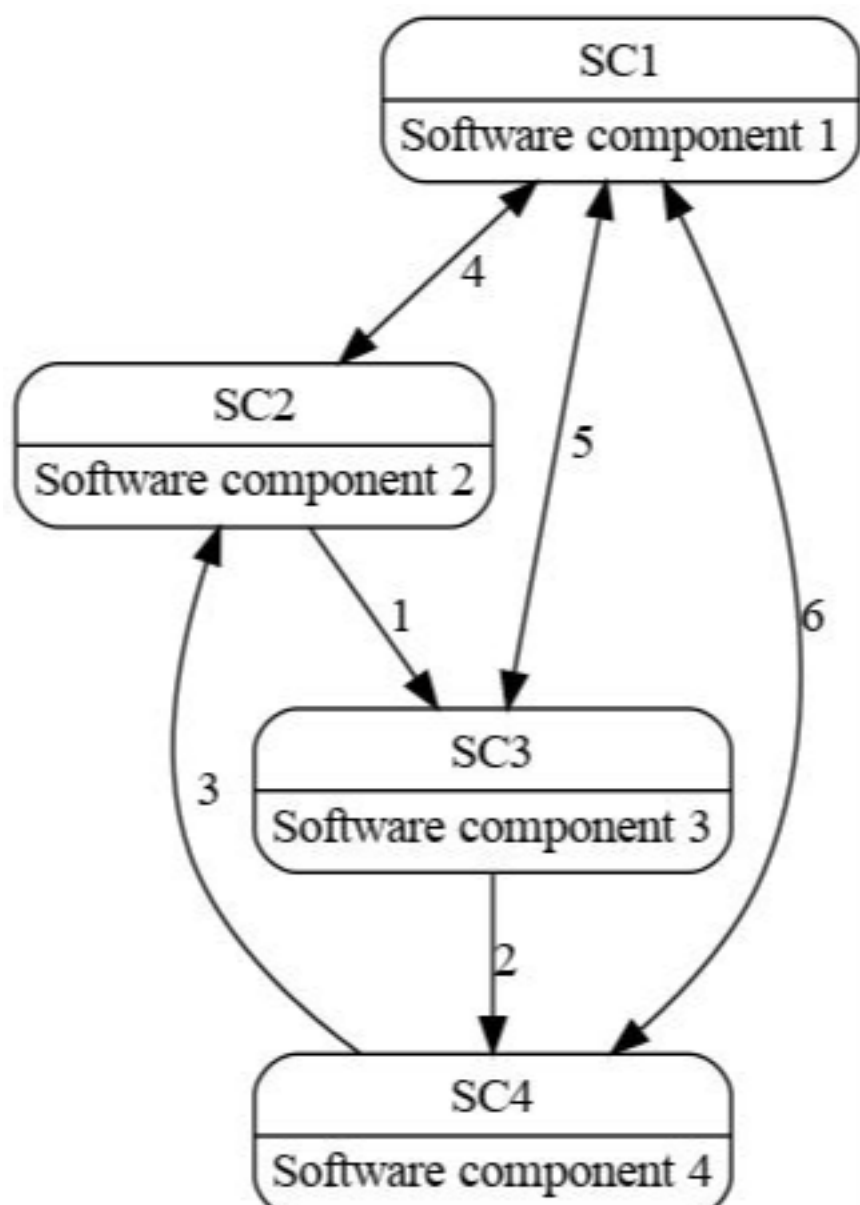
Si consideri l'automa seguente:

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



- 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;

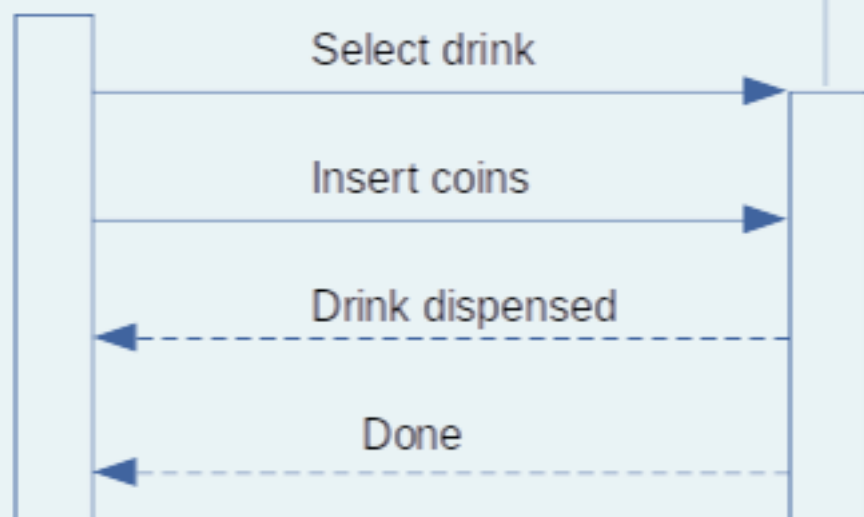
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?

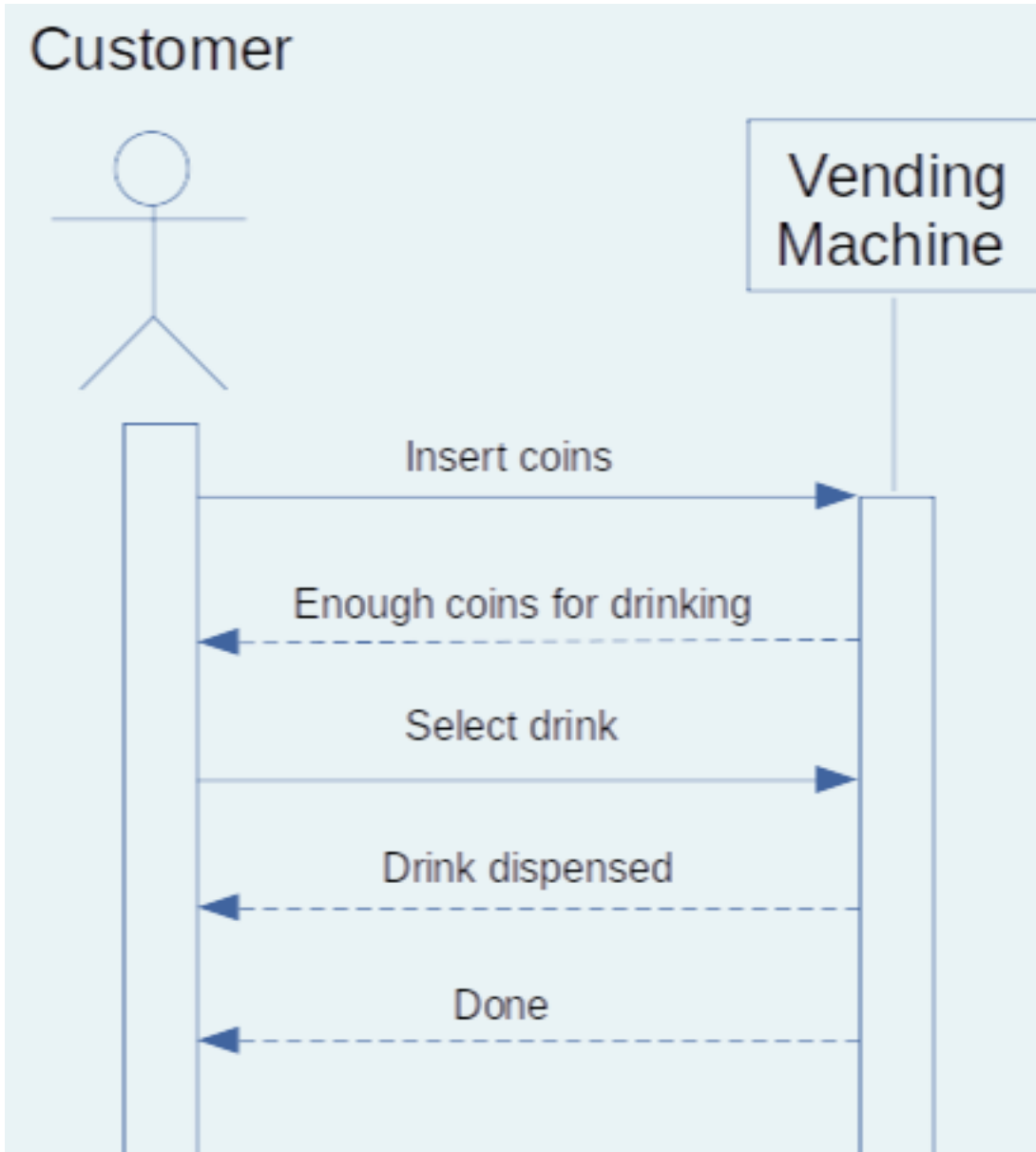


Customer



Vending Machine





iVBORw0KGgoAAAANSUgAAAACAAACAYAAACUpFJAAAAKEIEQVRliWN8+fnr4ZRMpGwSgYBaOAioBpoB0wCkbBKBgFo2D4AQCKdAPUYeTFMQAAAABJRU5ErkJggg==

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

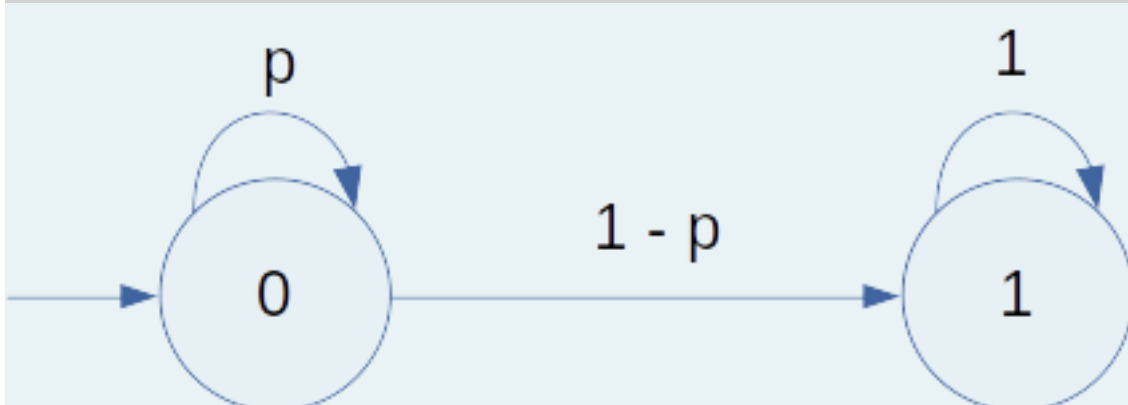
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

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

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

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



1.80%

2.60%

3.100%

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.

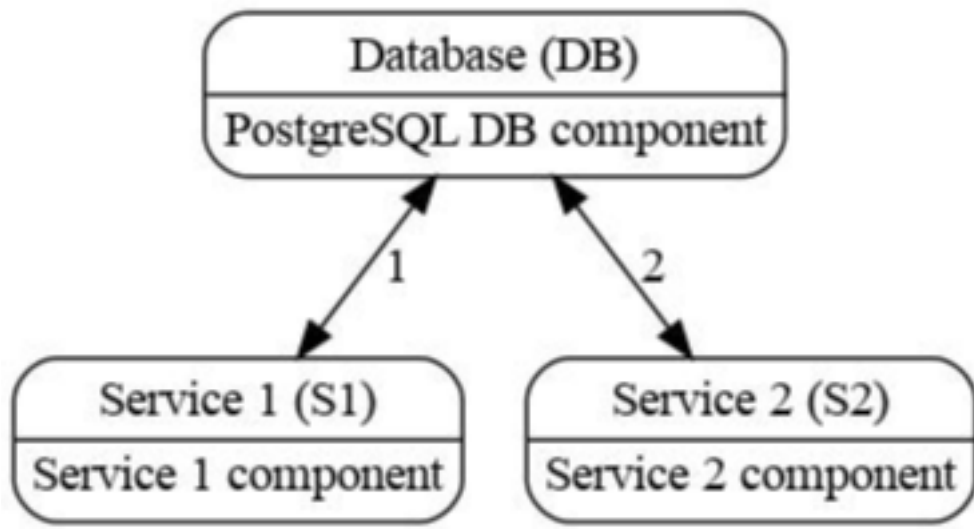
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: 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?



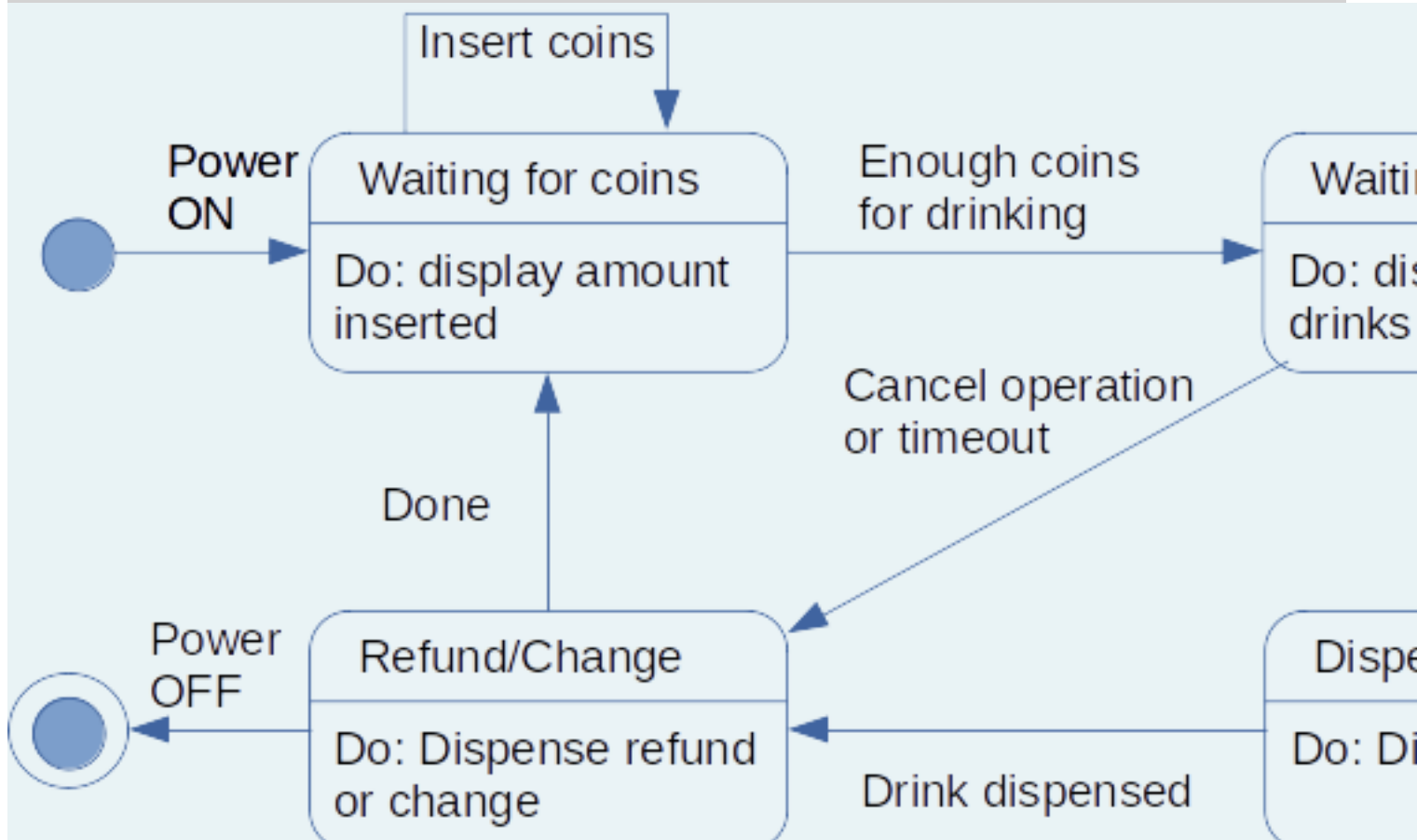
1. Transition coverage: 100%

2. Transition coverage: 70%

3. Transition coverage: 40%

Risposta : 3

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; elseif (pre(state) == 0) and
(Customer2Machine == 1) then // customer has inserted enough coins state := 1; Machine2Customer := 1; elseif (pre(state) == 1) and (Customer2Machine == 2) // drink selected
```

```

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

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

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

```

Risposta : 3

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.

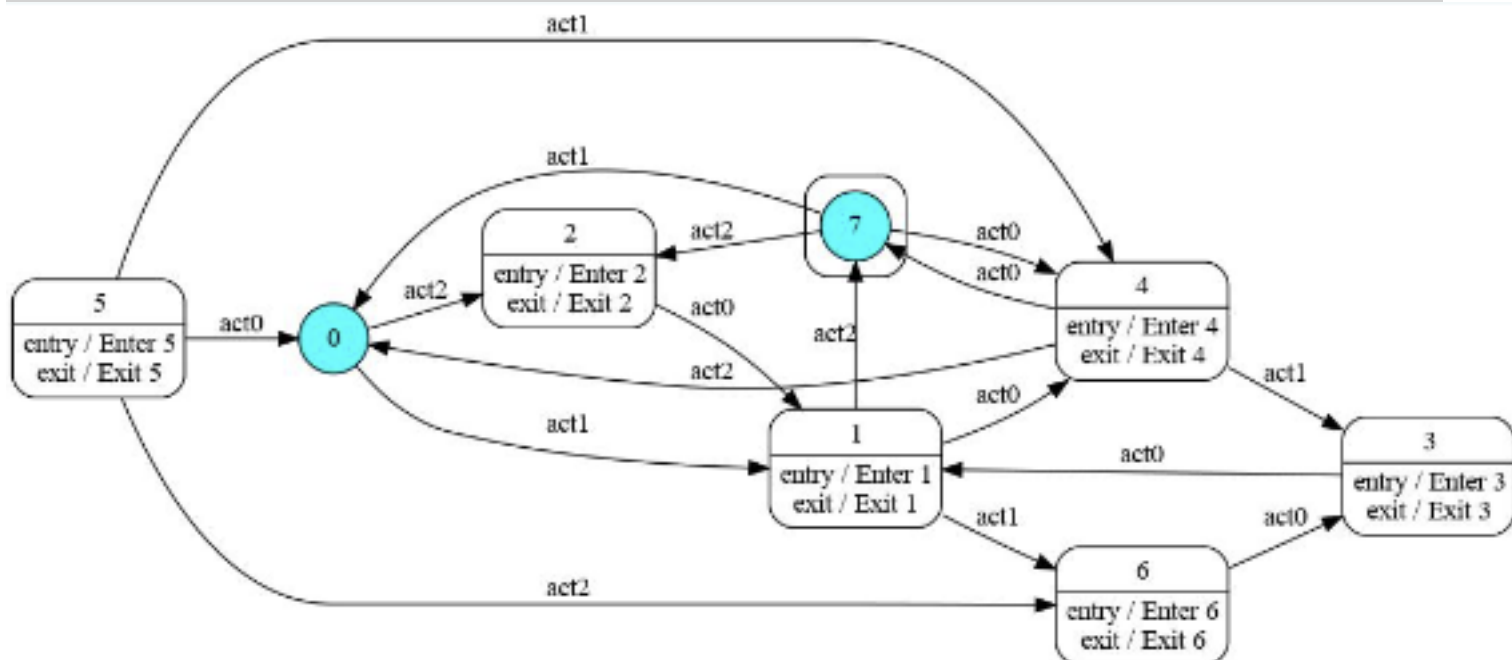
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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.

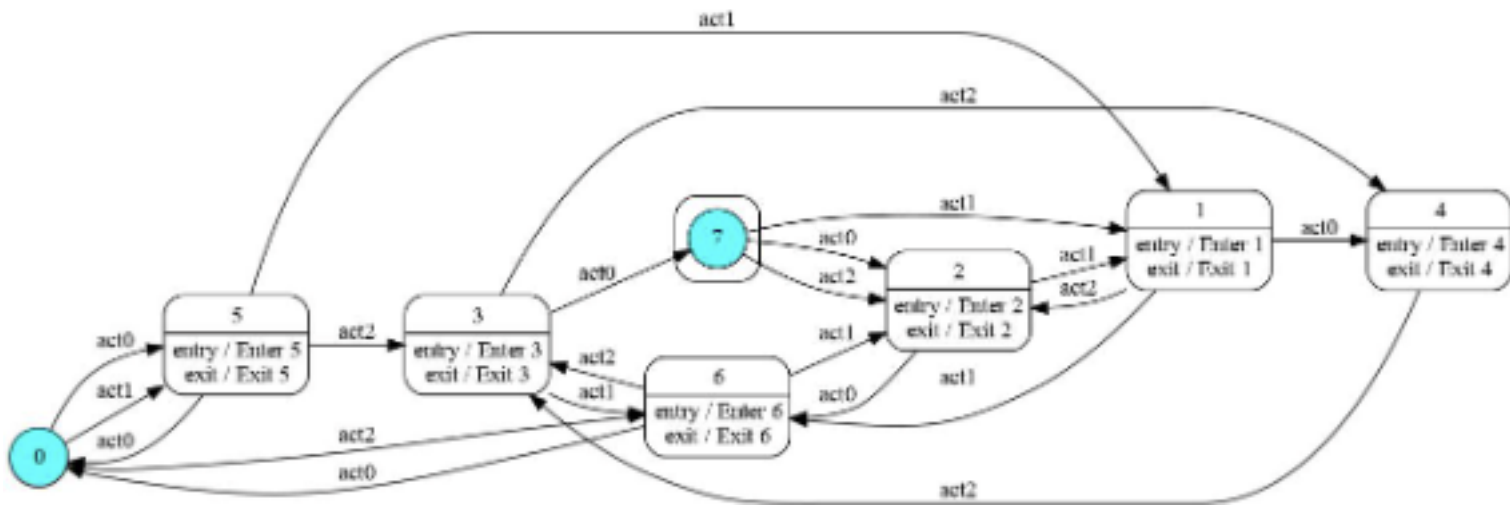
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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%

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.

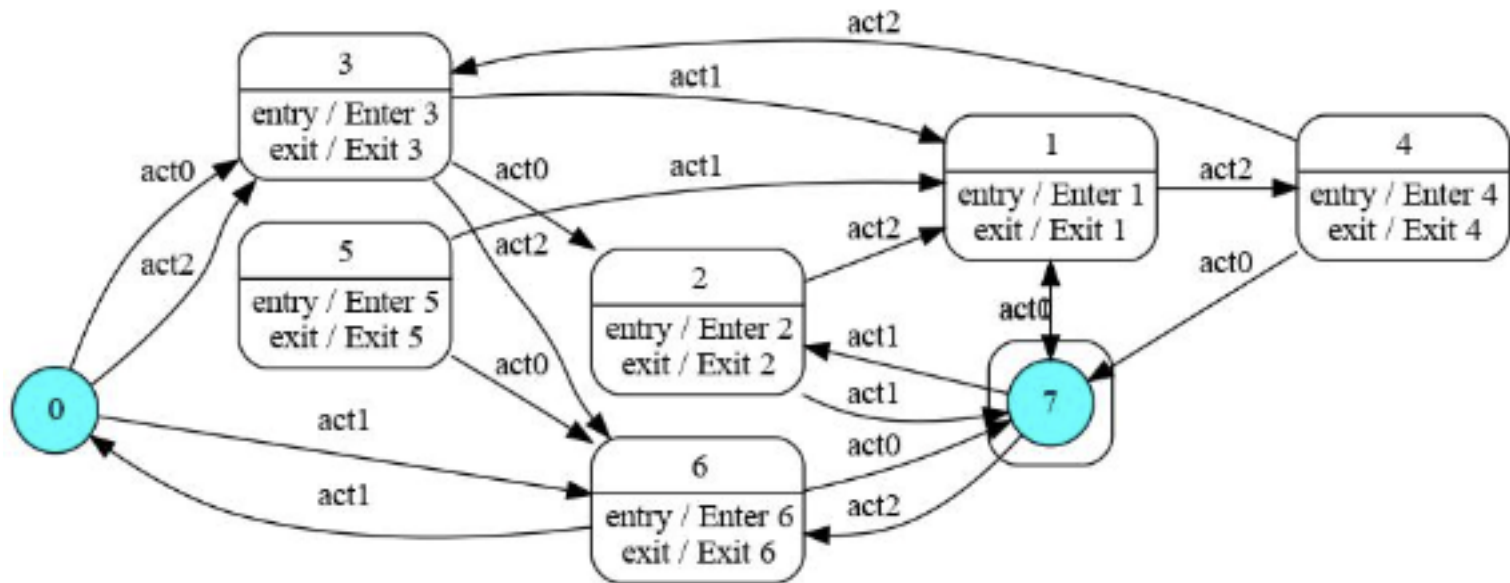
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act2 act0 act2 act2 act2

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

Test case 3: act2 act2 act2 act0 act1 act0

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



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

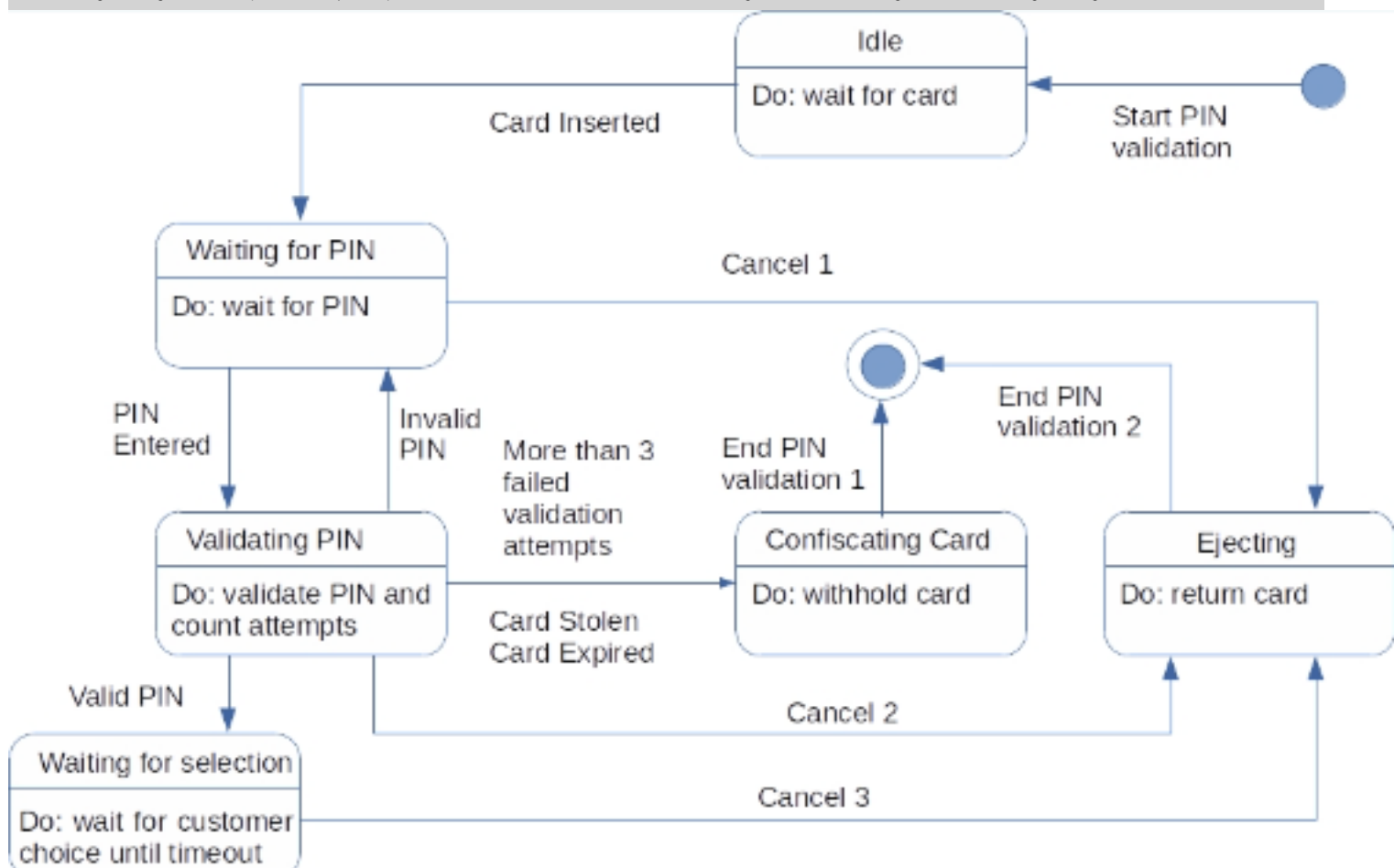
Forms and Query Manager

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.

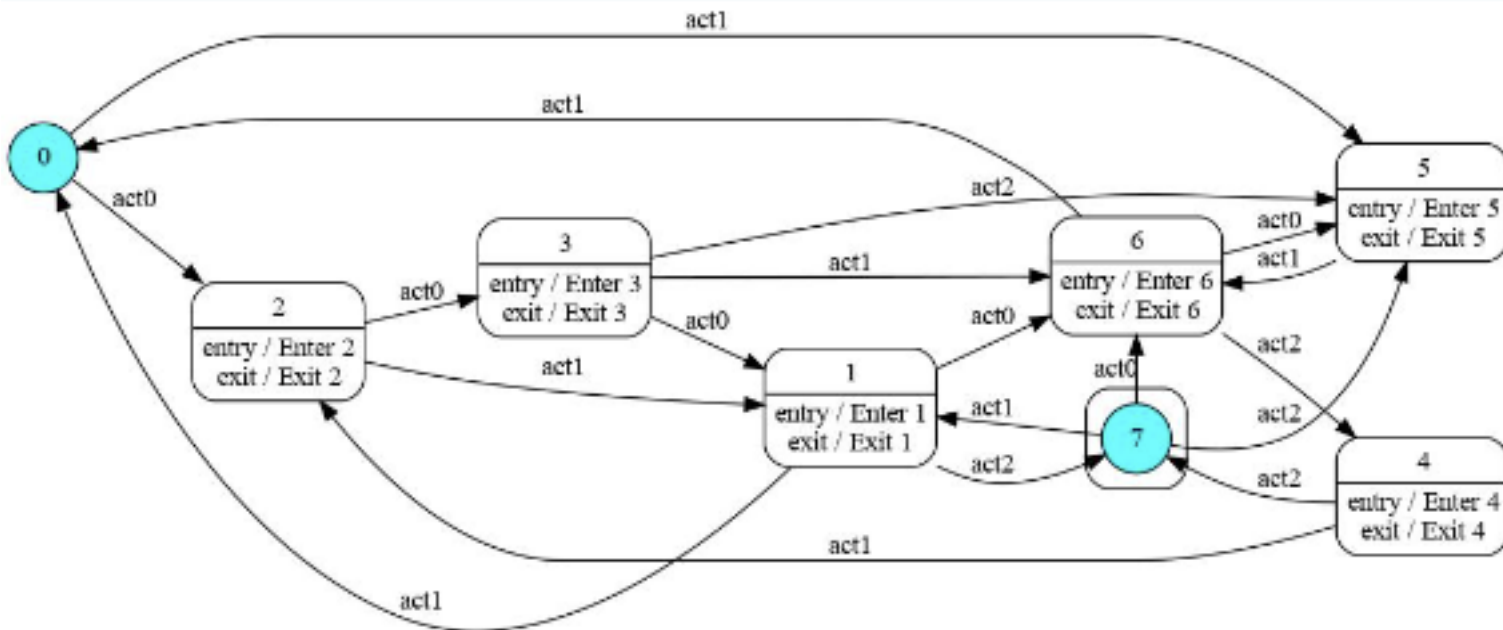
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

Test case 1: act1 act1 act0 act1 act1 act2 act0

Test case 2: act2 act0 act0

Test case 3: act1 act1 act2 act0 act0

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



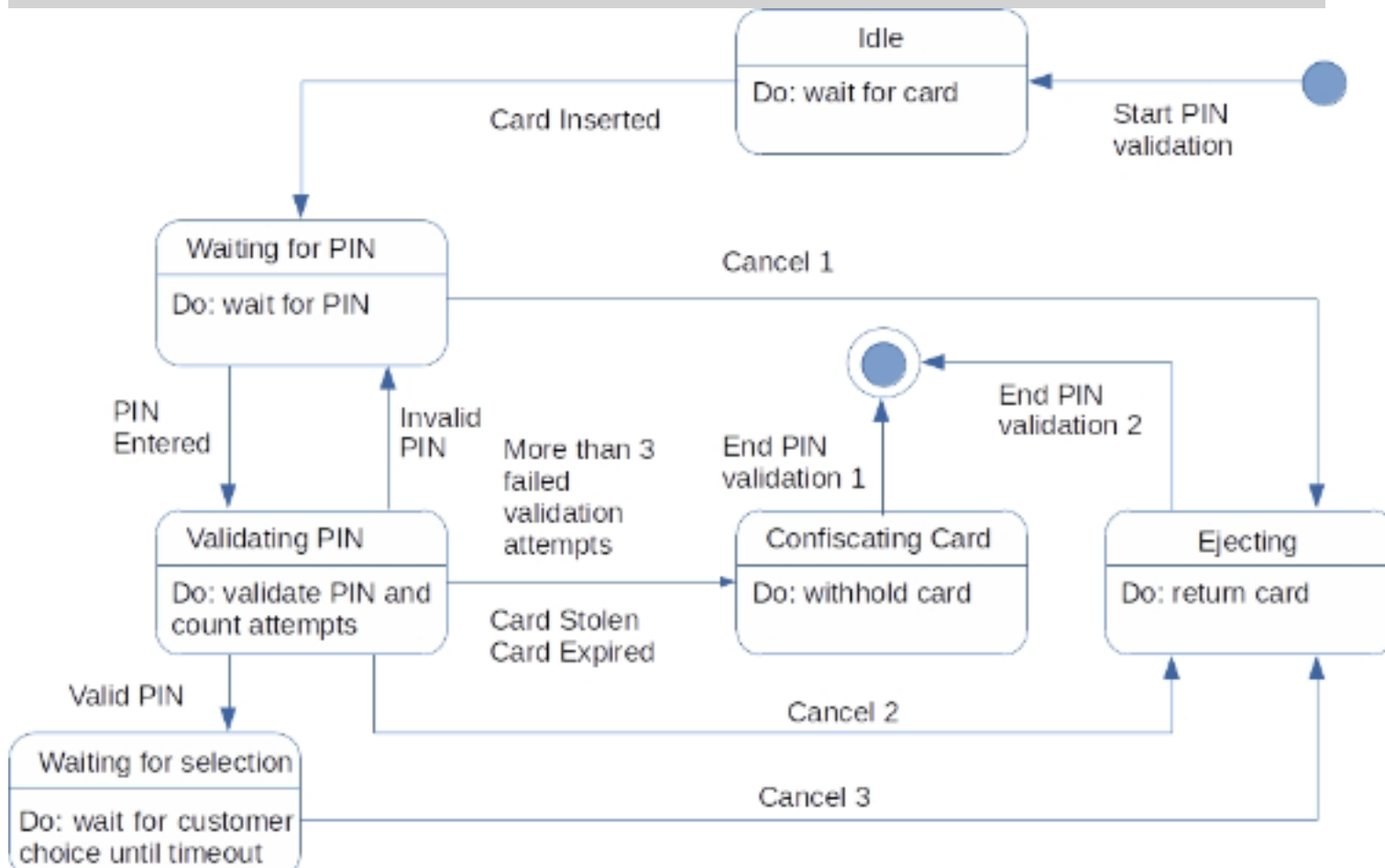
1.State coverage: 90%

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 ?



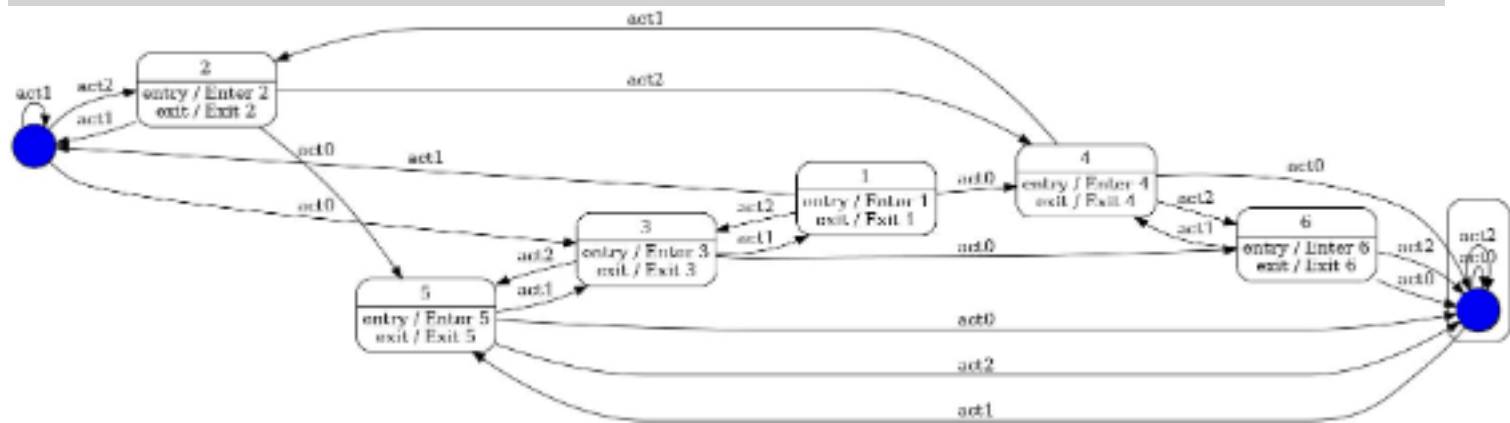
1. Anche se ci sono abbastanza monete potrebbe non essere possibile ottenere la bevanda selezionata.

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

Risposta : 3

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.

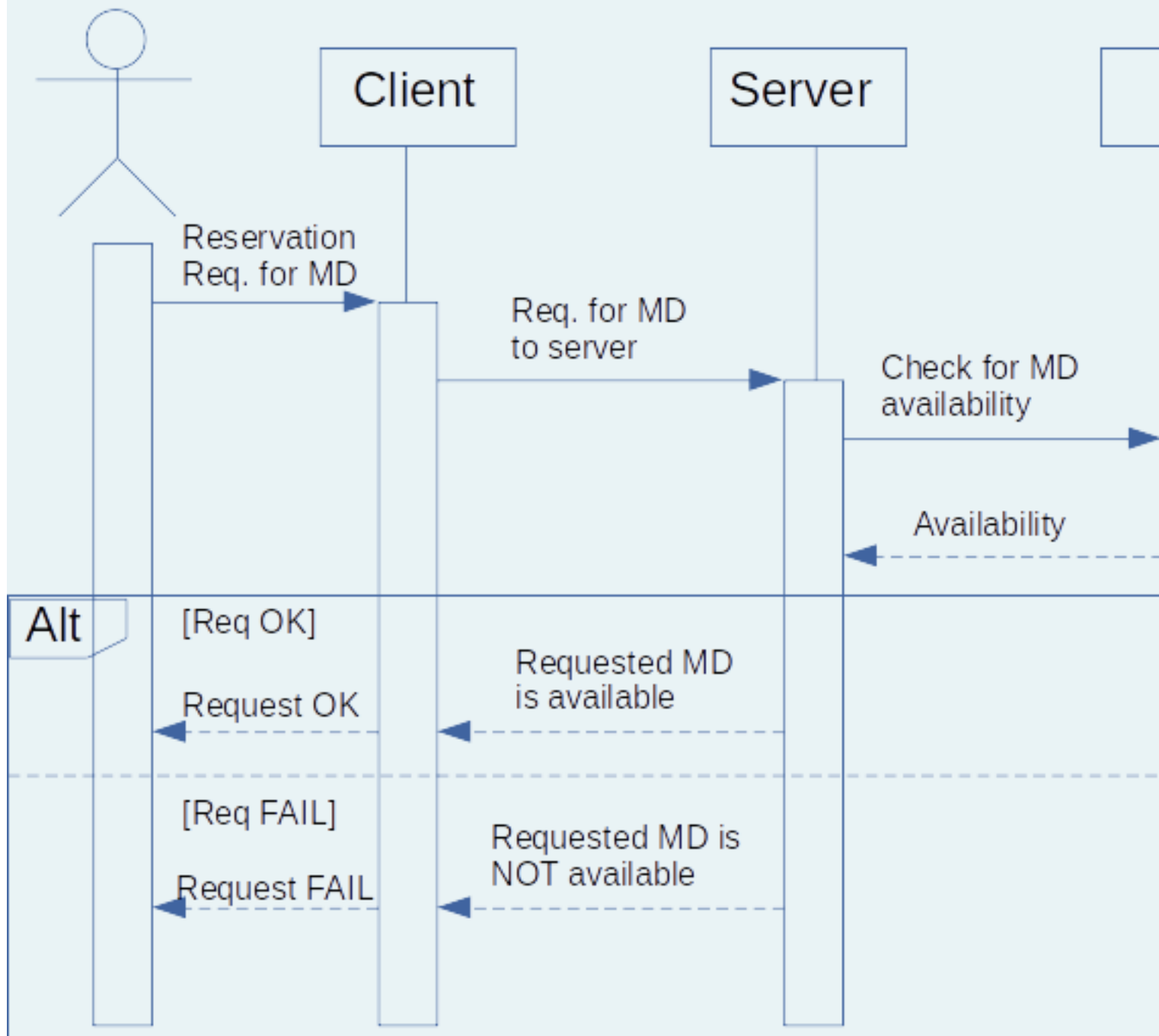
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

- 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

Patient

MD: Medical Doctor



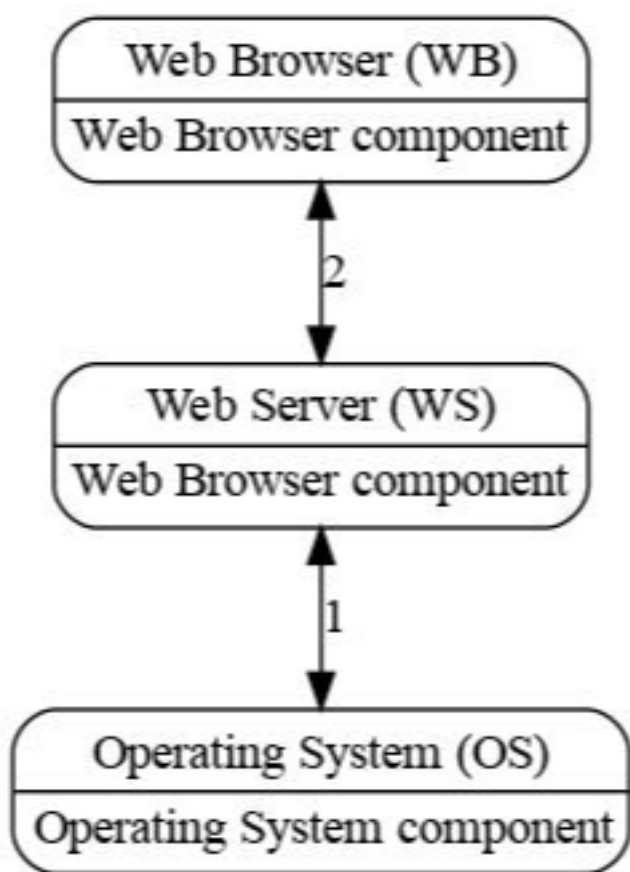
1.State coverage: 75%

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.

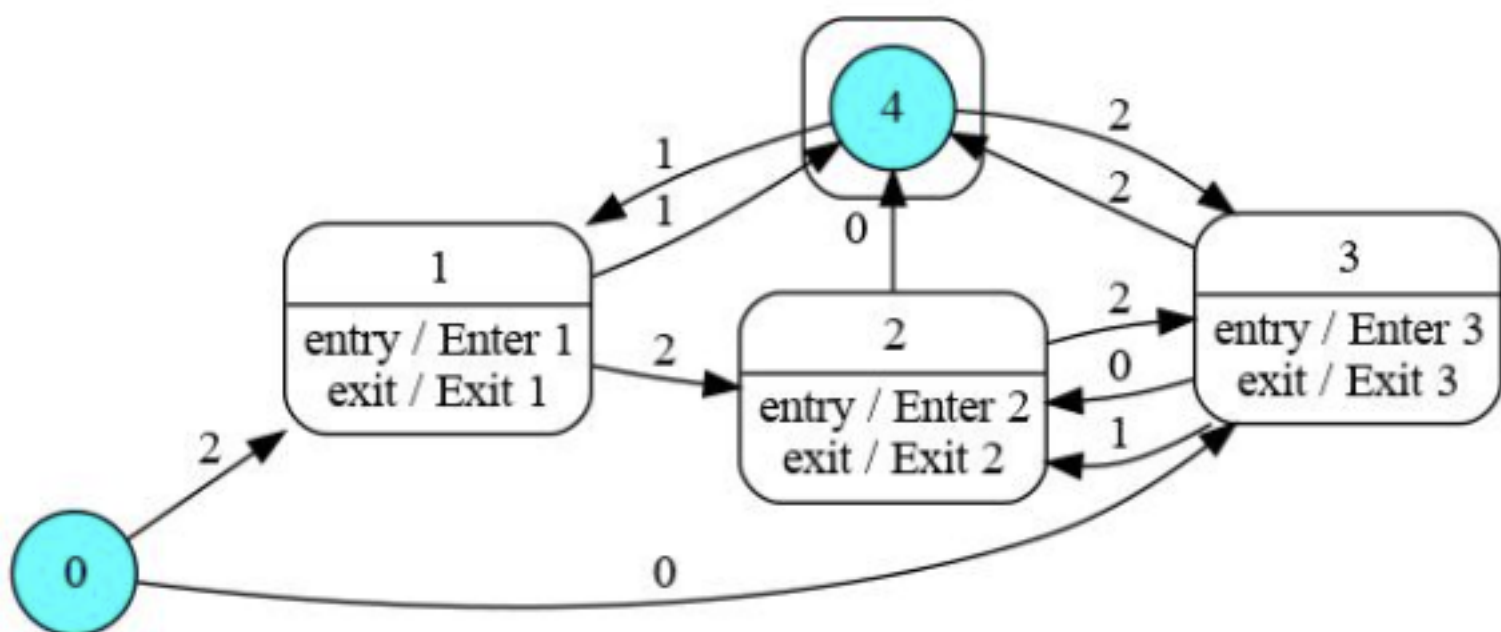
Risposta : 3

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

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

Uno scenario è una sequenza di stati.

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

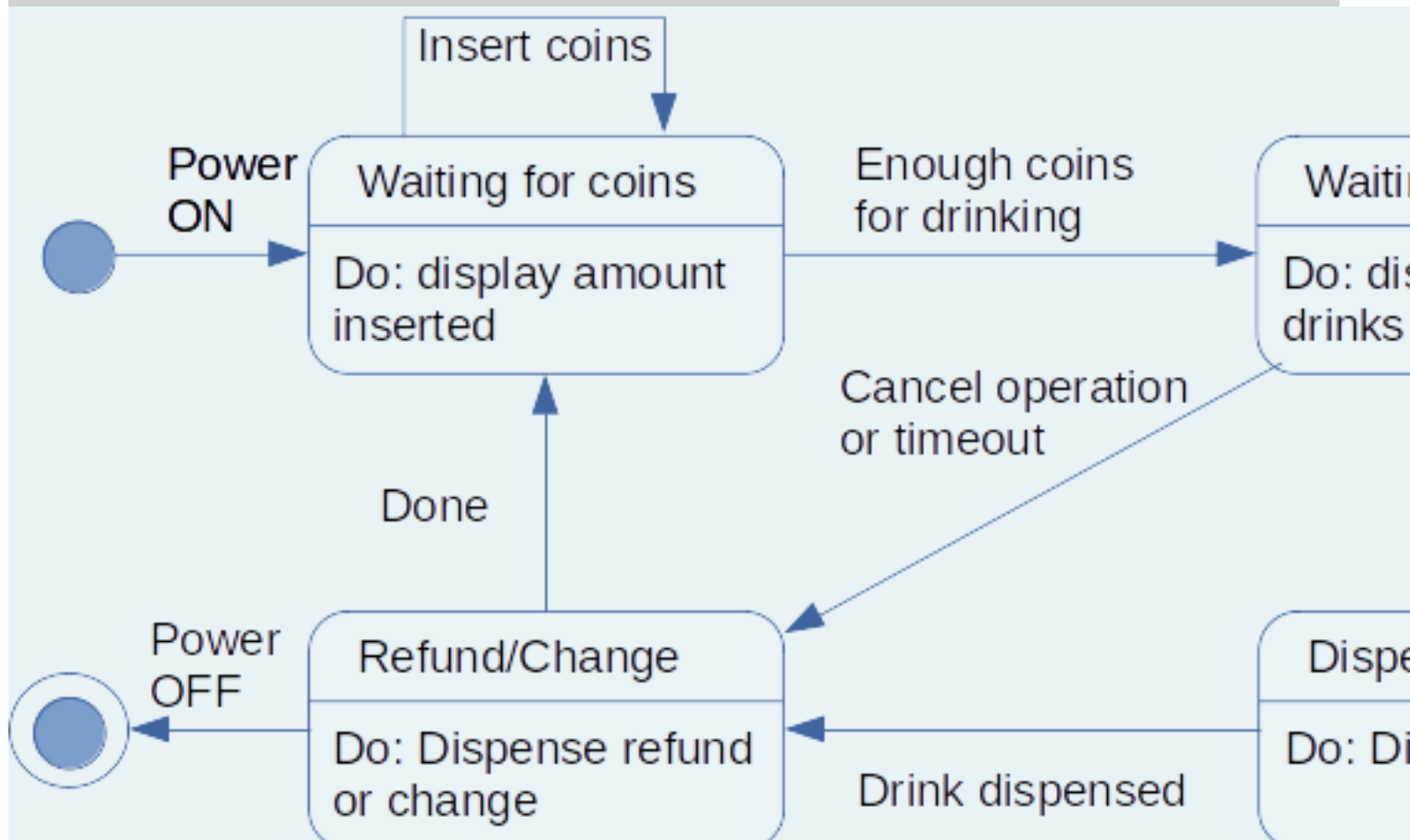


1.0.28

2.0.42

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.



1. $1/(1 - p)$

2. $(1 - p)/p$

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

Risposta : 1

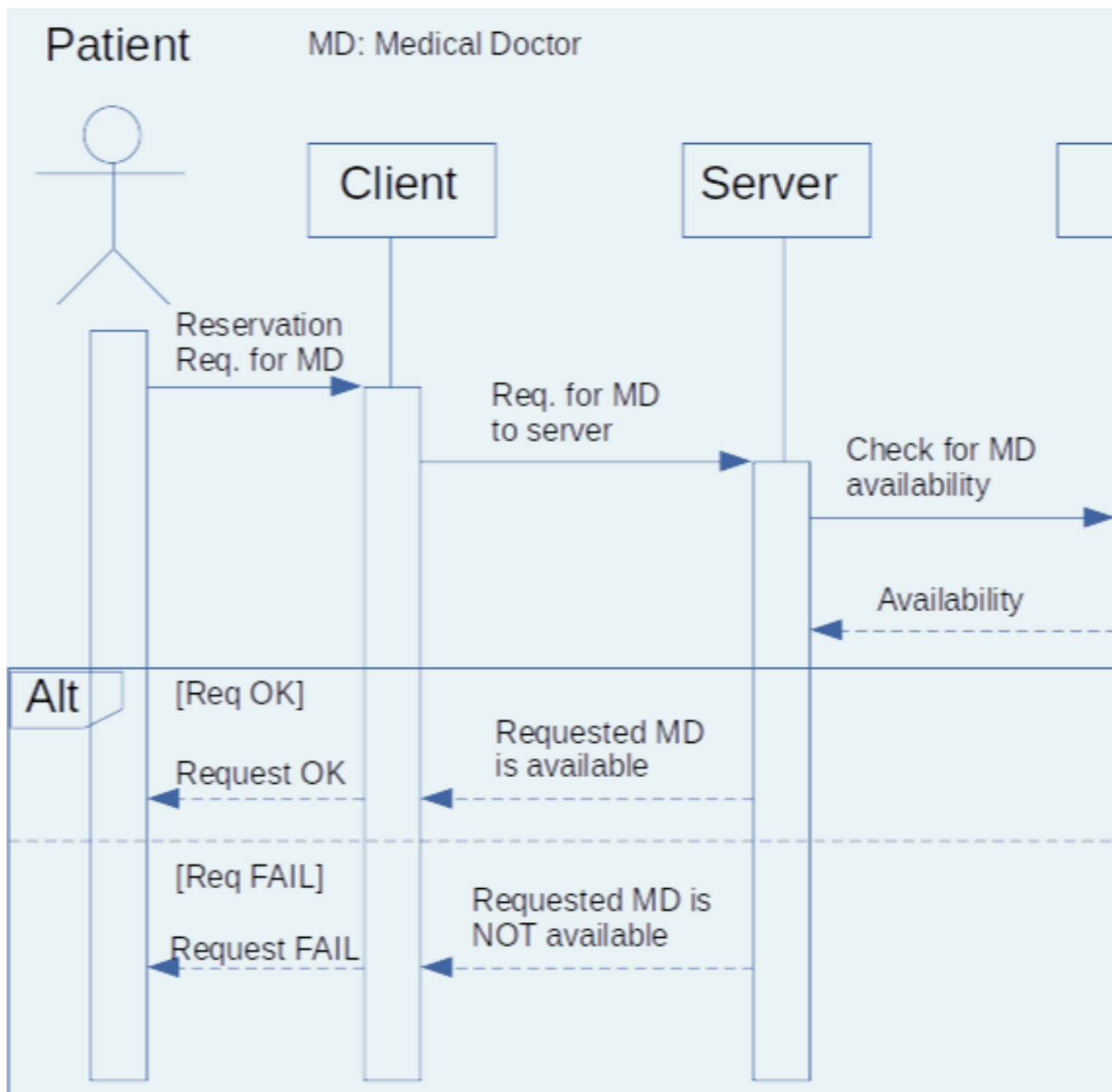
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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 act0 Quale 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.

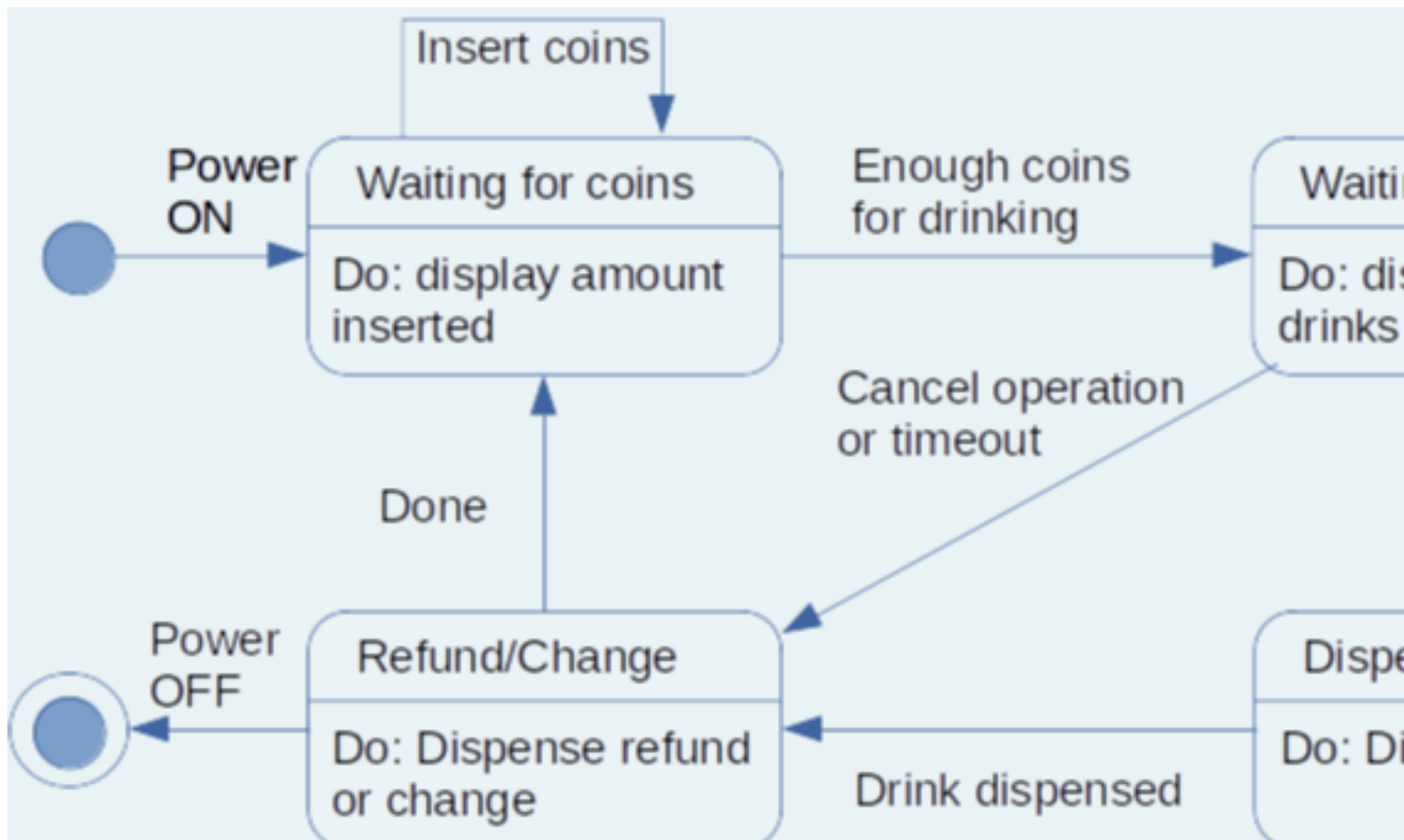
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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

Test case 2: act1

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

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



- 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.

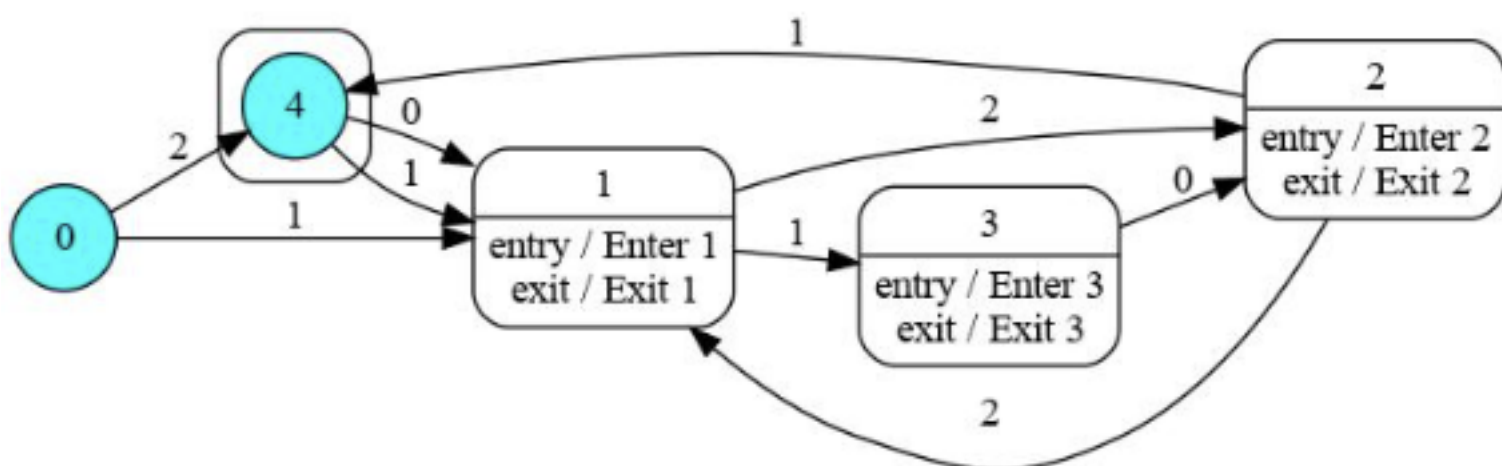
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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%

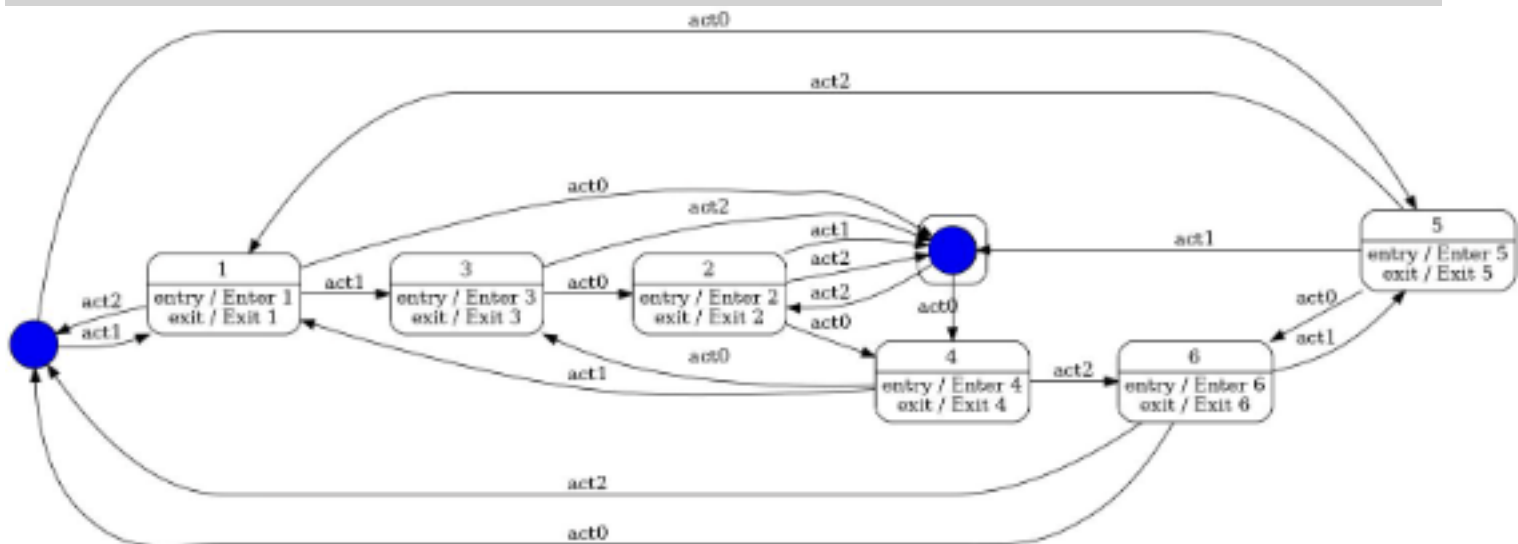
Risposta : 1

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 act2 act1 act2 act1 act2 act0 act1

Test case 2: act0 act2 act0

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



1.State coverage: 60%

2.State coverage: 90%

3.State coverage: 40%

Risposta : 1

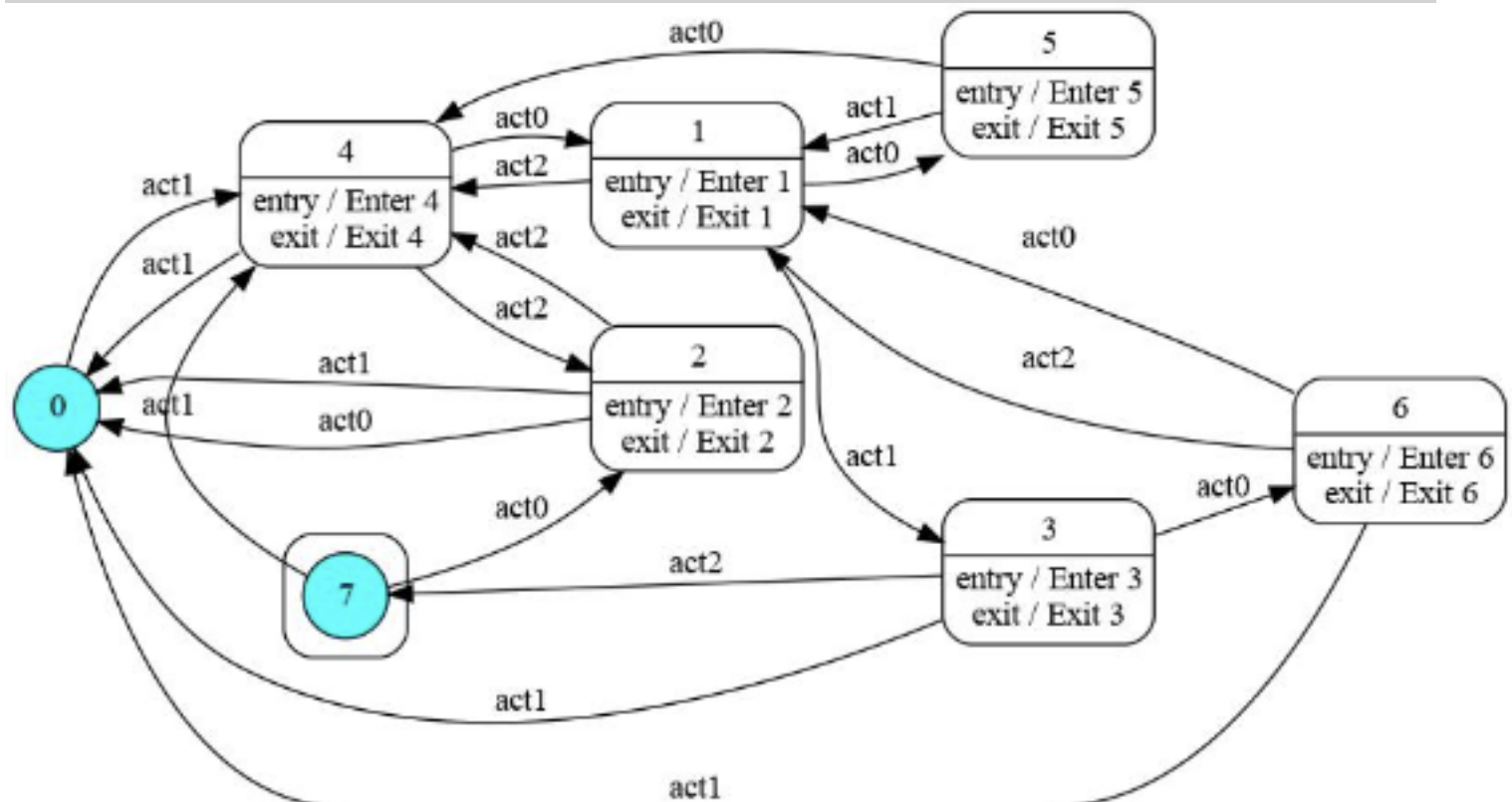
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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 act2 act1 act2 act1 act1 act1 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.

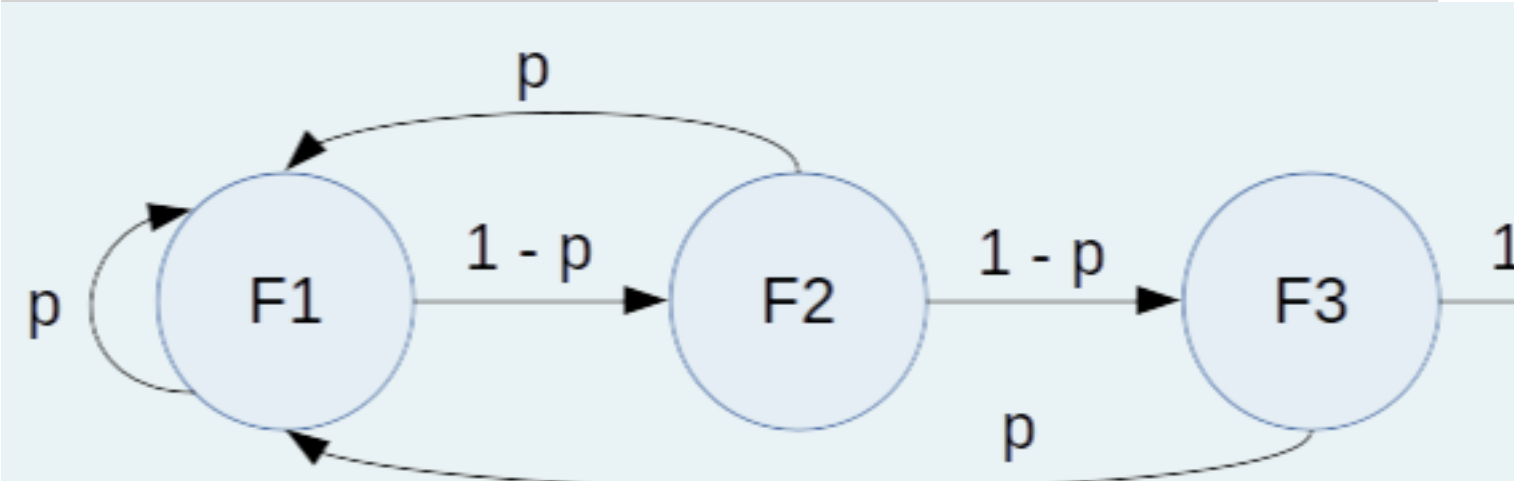
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act1 act1 act1

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

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

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



1.100%

2.60%

3.80%

Risposta : 3

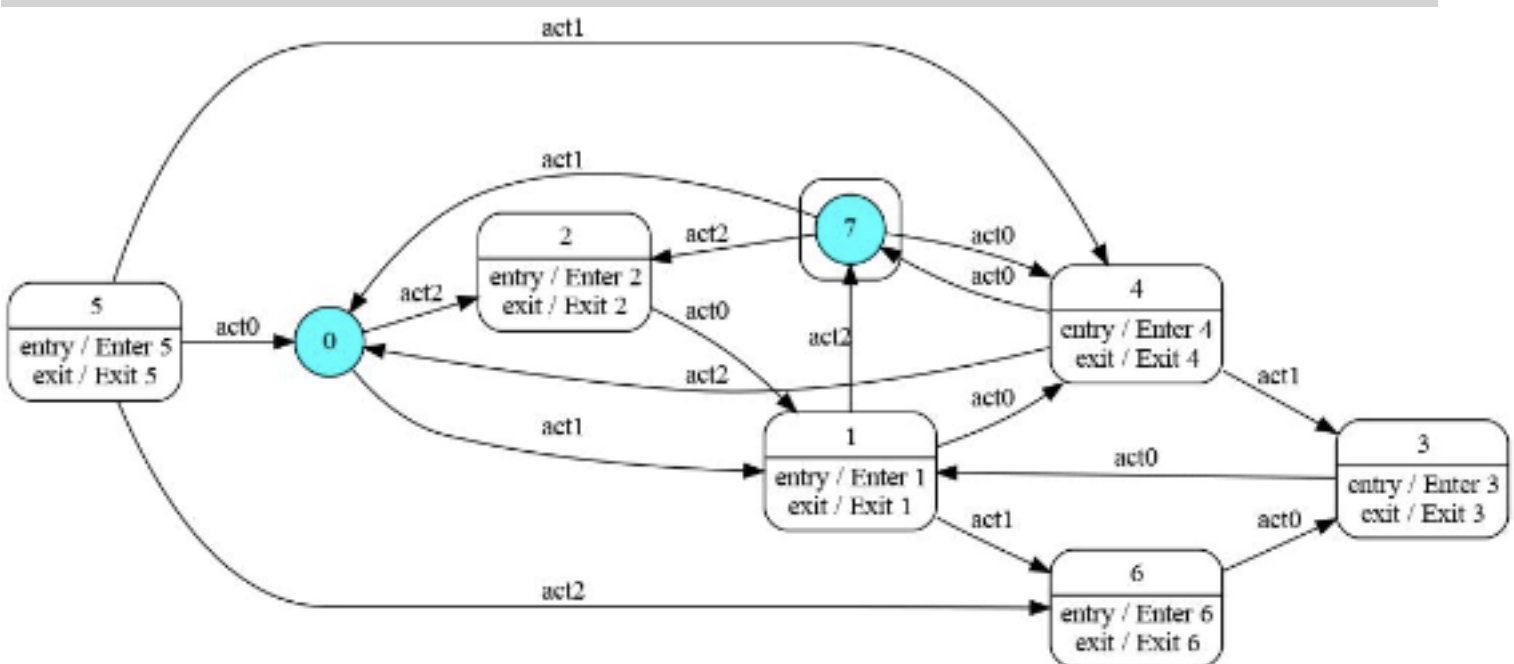
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:Test case 1: act0 act0 act2 act0 act0 act0 act0 act2 act1 act2 act0 act2 act2 act2 act2 act0 act0 act1 act2 act2 act0 act2

act0 act2 act1 act0 act2 act1 act2 act2 act0 act2

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

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



1.State coverage: 100%

2.State coverage: 80%

3.State coverage: 60%

Risposta : 1

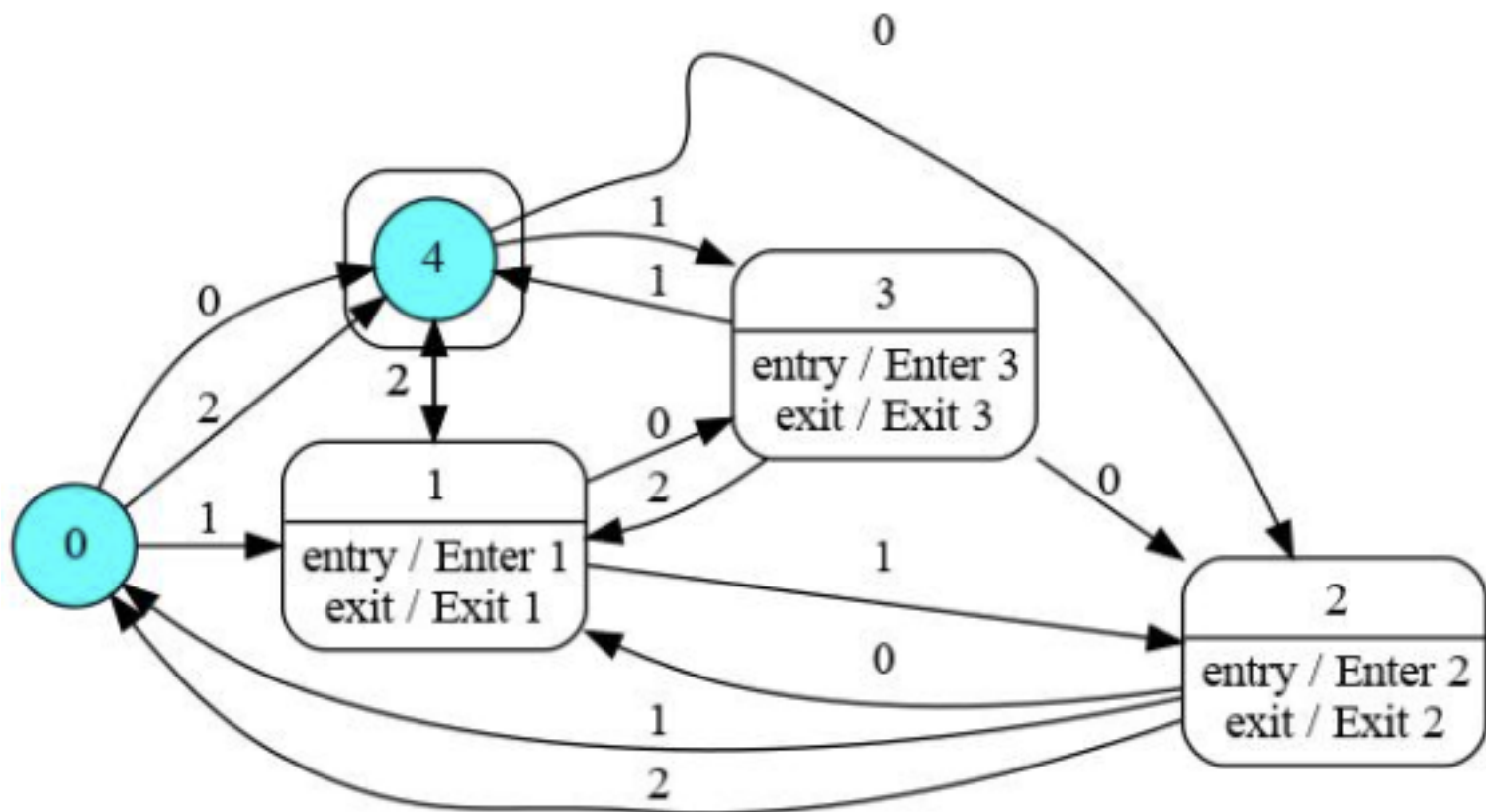
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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.

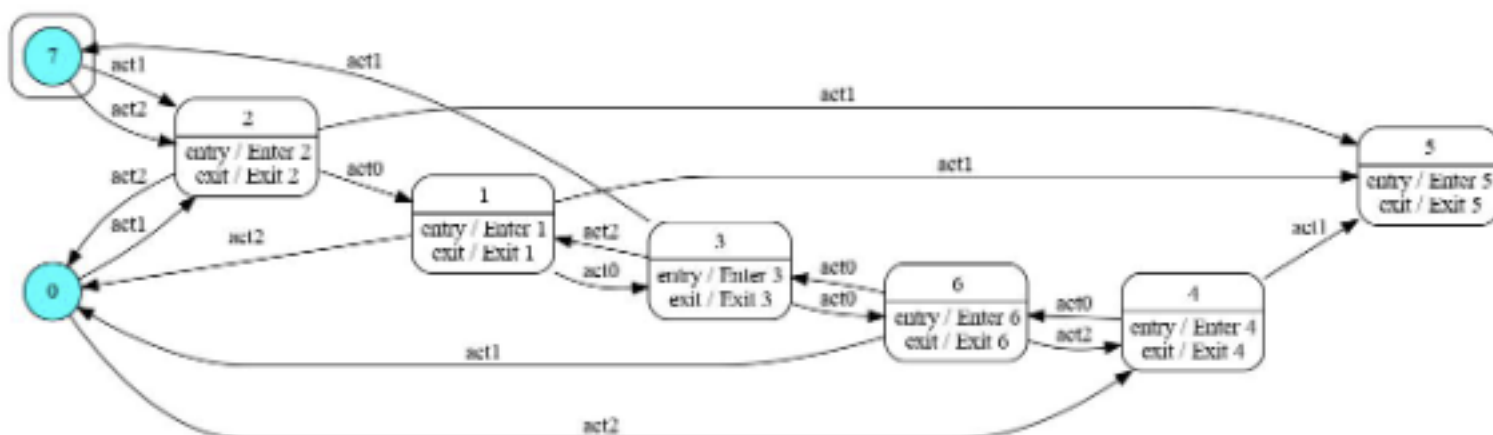
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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%

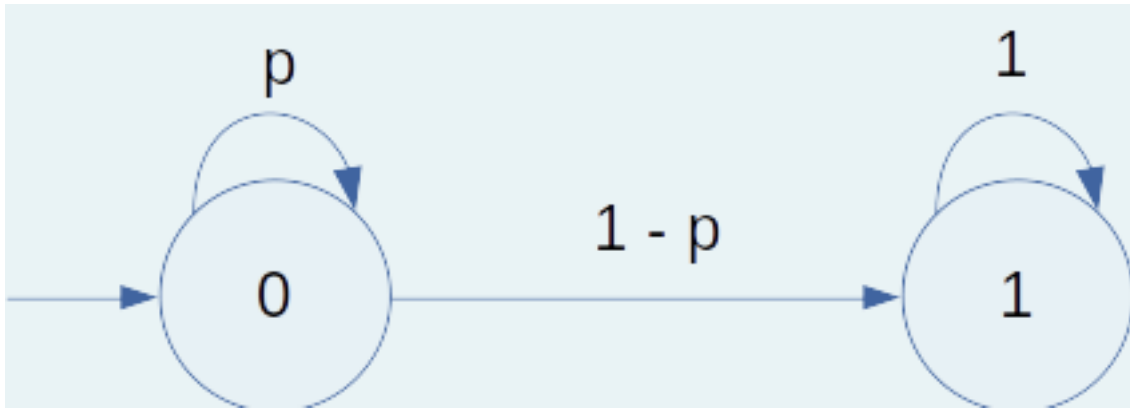
Risposta : 2

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 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

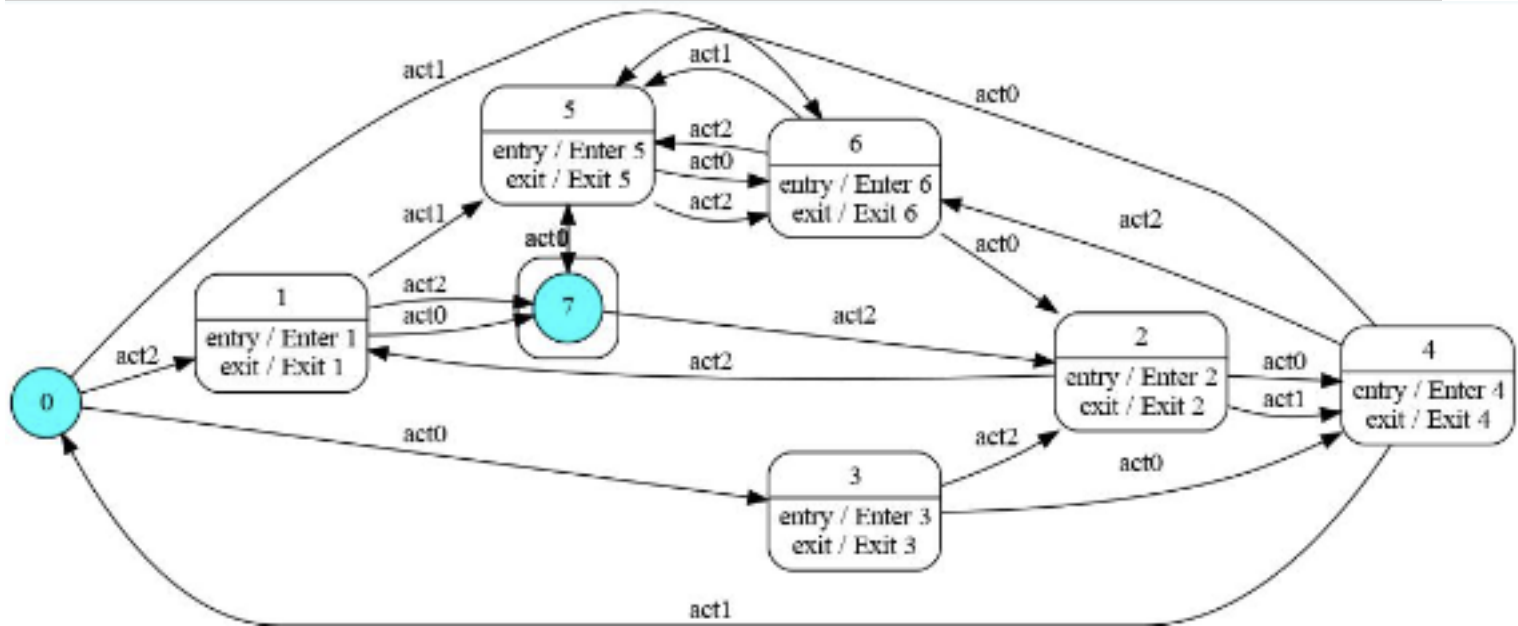
Si consideri il seguente insieme di test cases:

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%

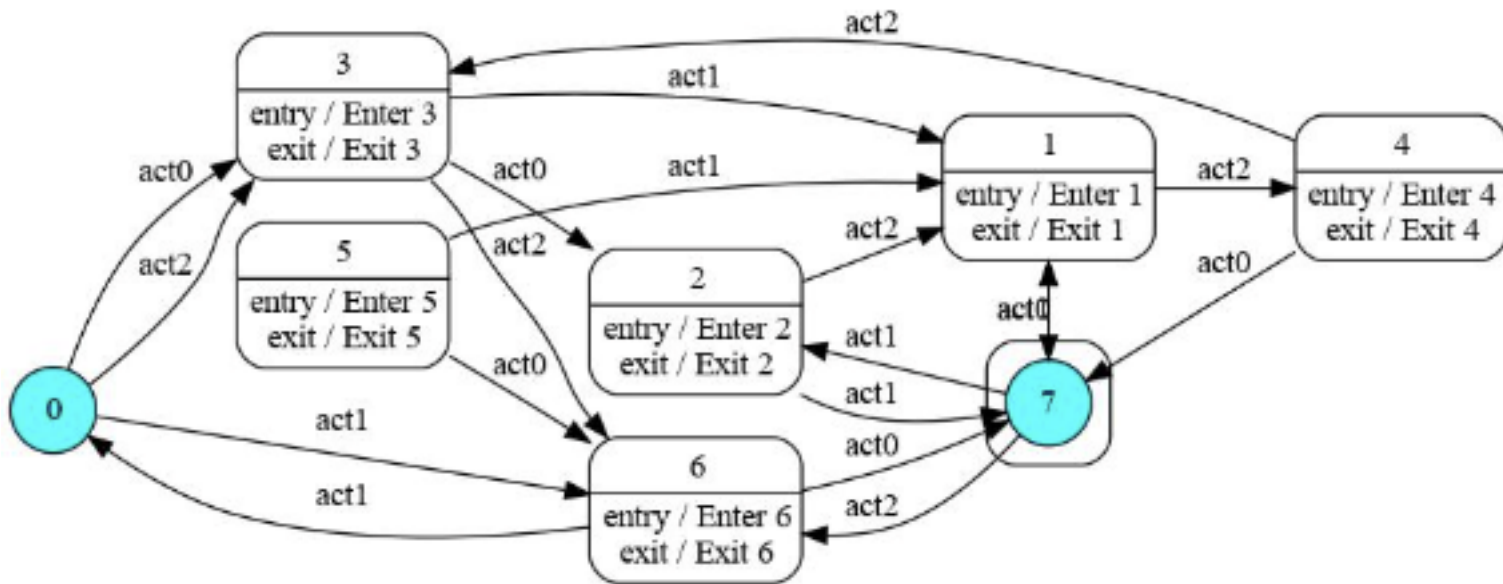
Risposta : 2

Un processo software può essere rappresentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del procoesso 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 0.1 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

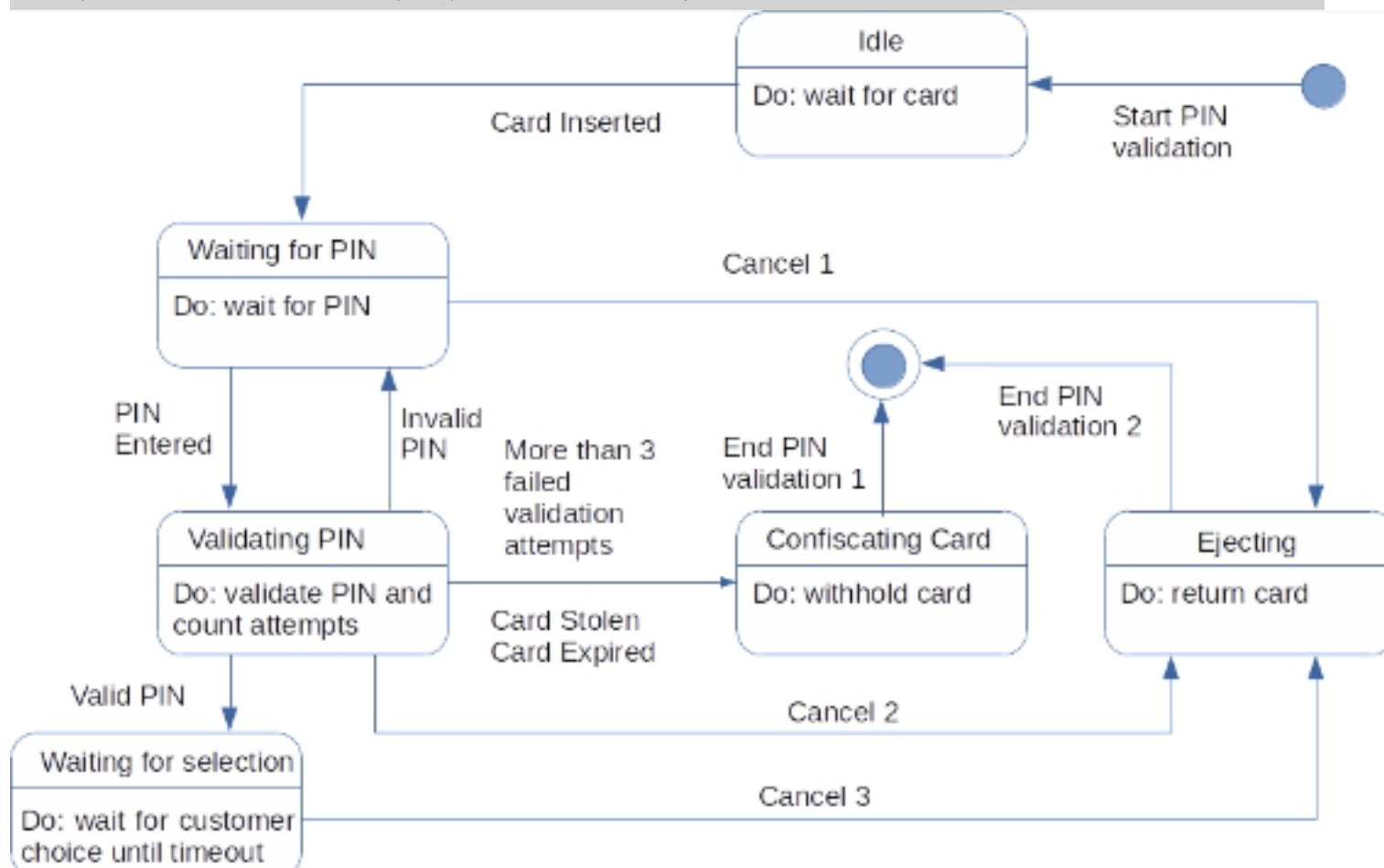
Risposta : 2

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

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

Uno scenario è una sequenza di stati.

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



1.0.56

2.0.14

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.

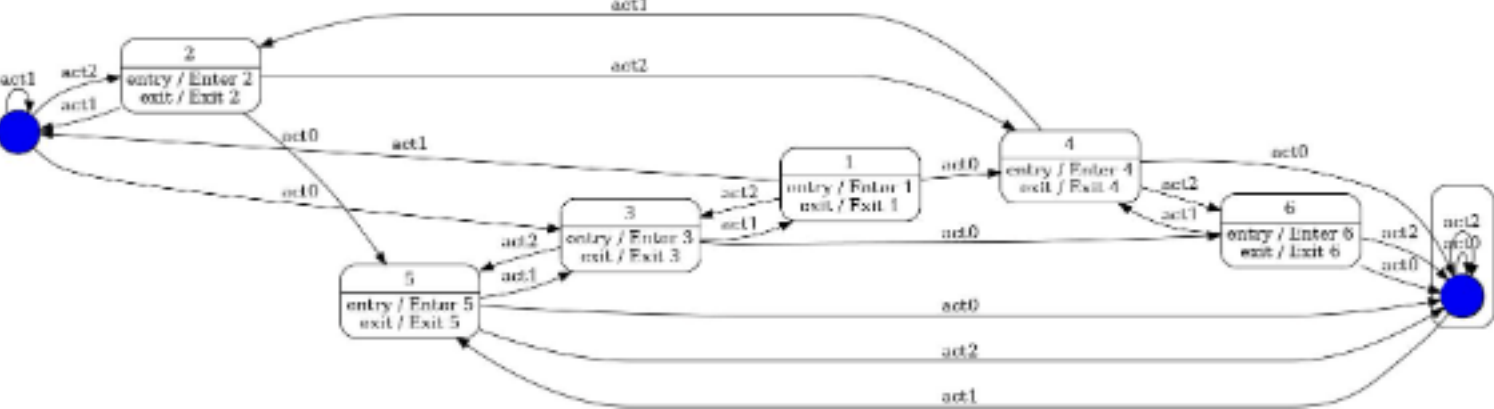
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

Test case 1: 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.

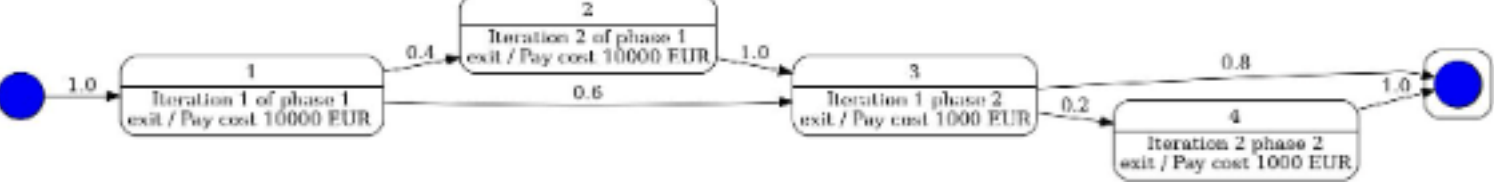
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

Test case 2: act0 act1 act2 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?



1.100%

2.60%

3.80%

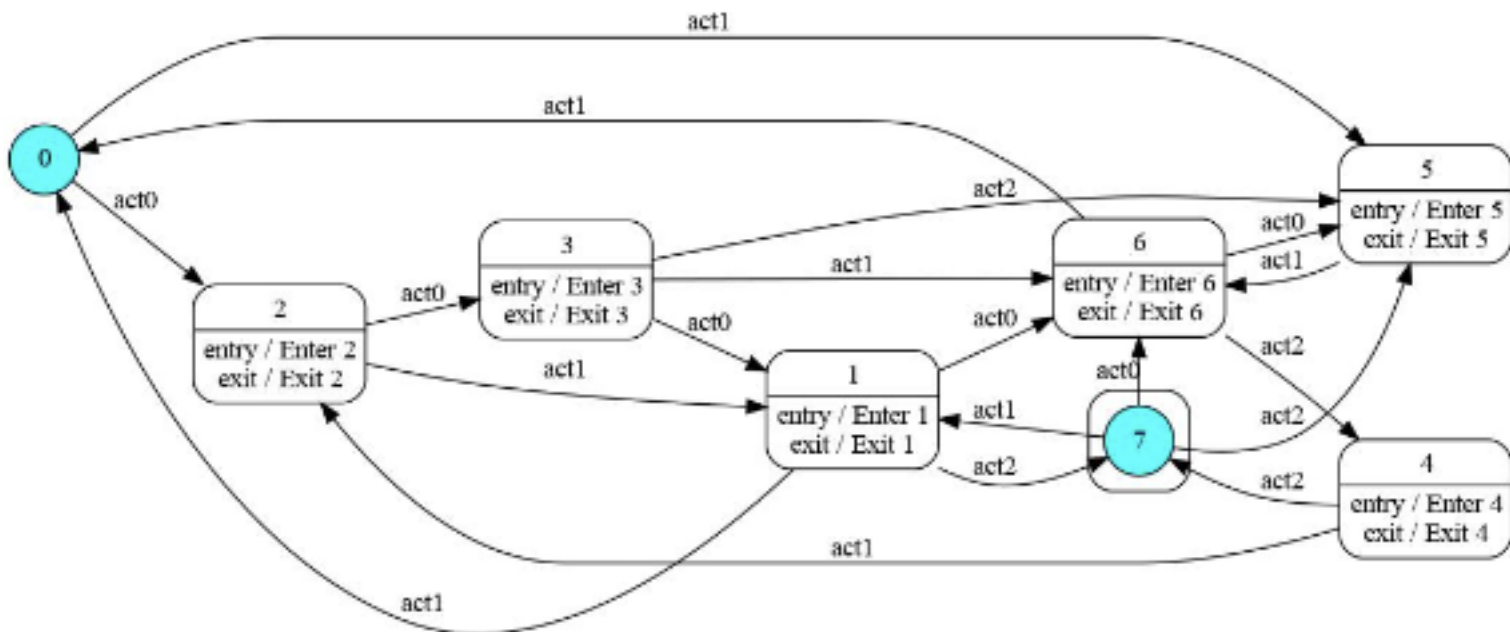
Risposta : 3

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.

Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:Test case 1: act2 act2 act2 act1 act2 act1 act2 act2 act1

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

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



- 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.

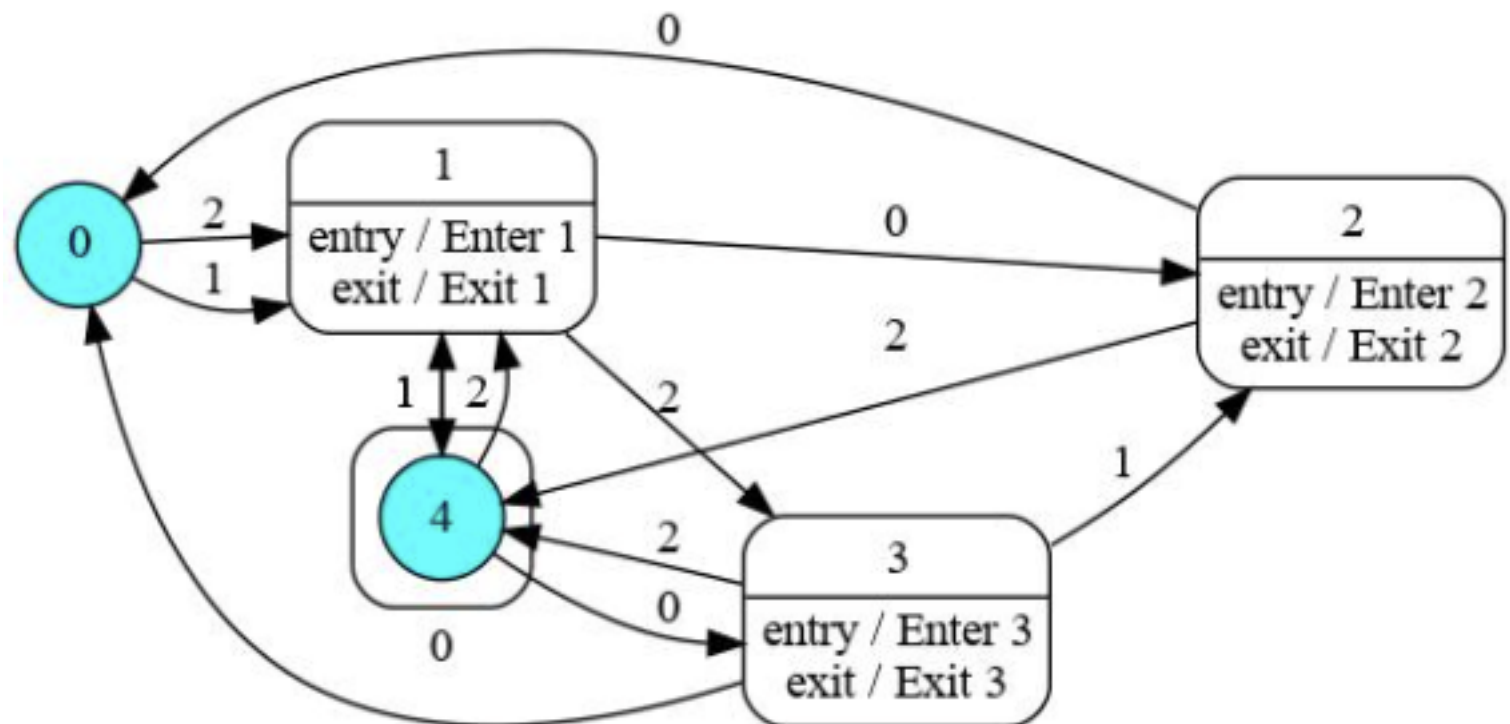
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

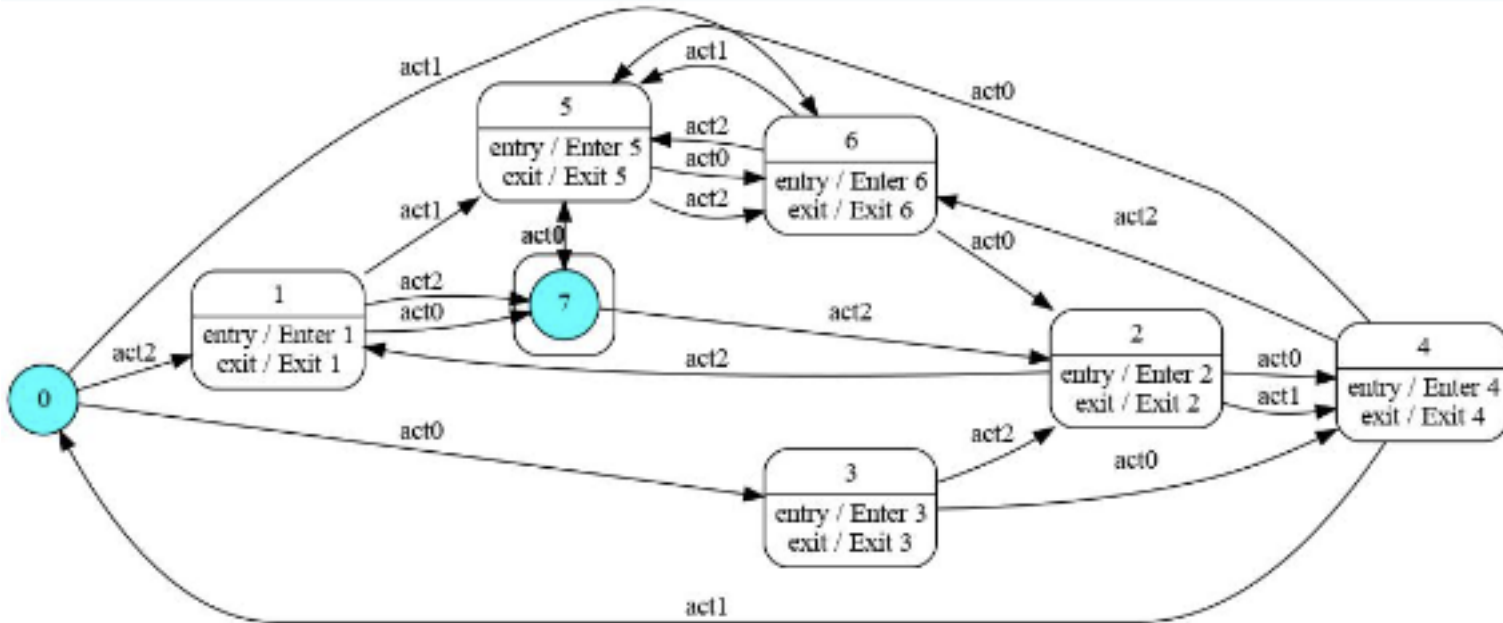
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il tempo necessario per completare la fase x è time(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappresenta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

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

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



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

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

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

Risposta : 3

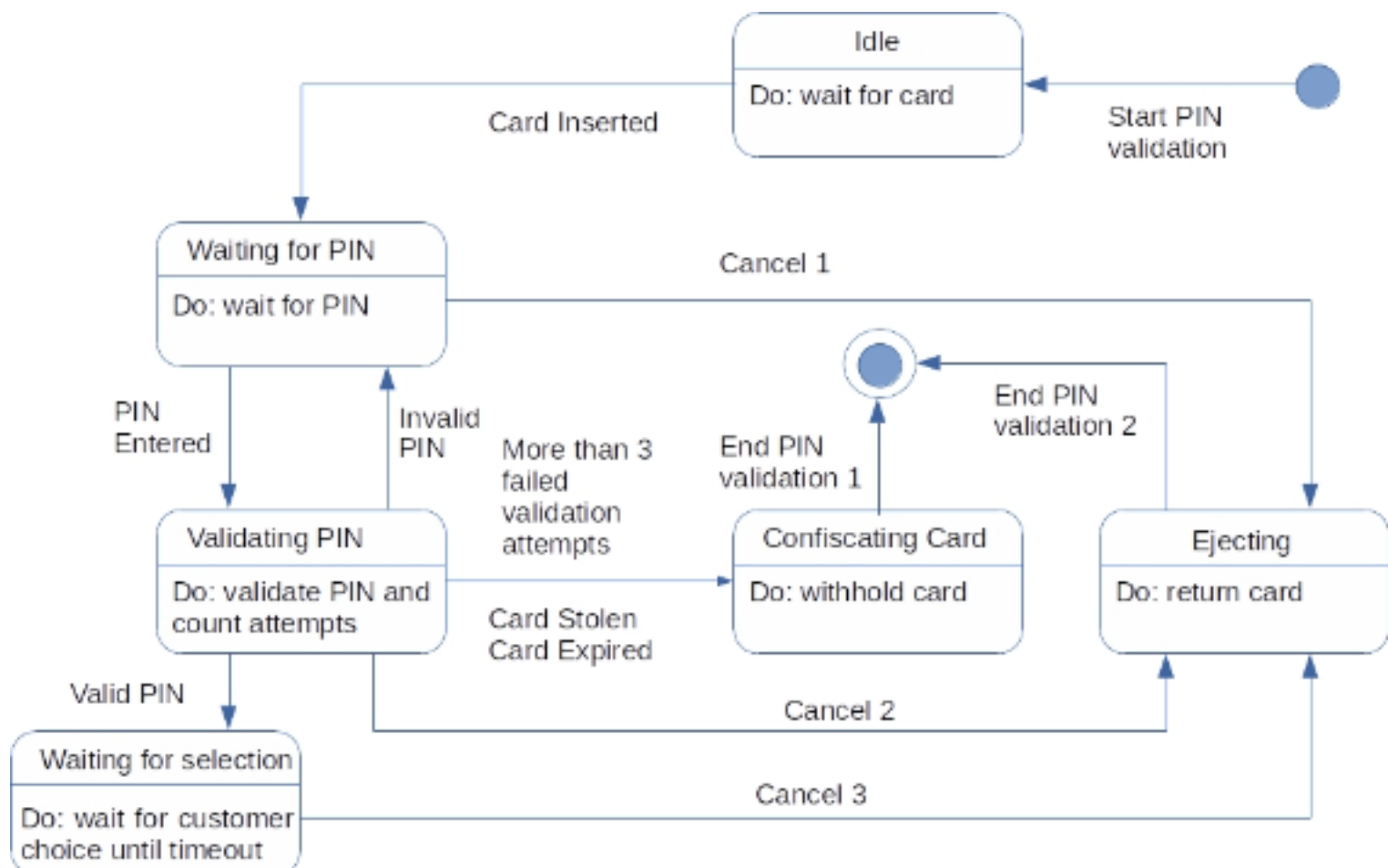
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il costo dello stato (fase) x è $c(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappresenta il completamento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

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

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

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

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



1. $c(0)/(1 - p)$
2. $c(0) \cdot (1 - p)/p$
3. $c(0)/(p \cdot (1 - p))$

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.

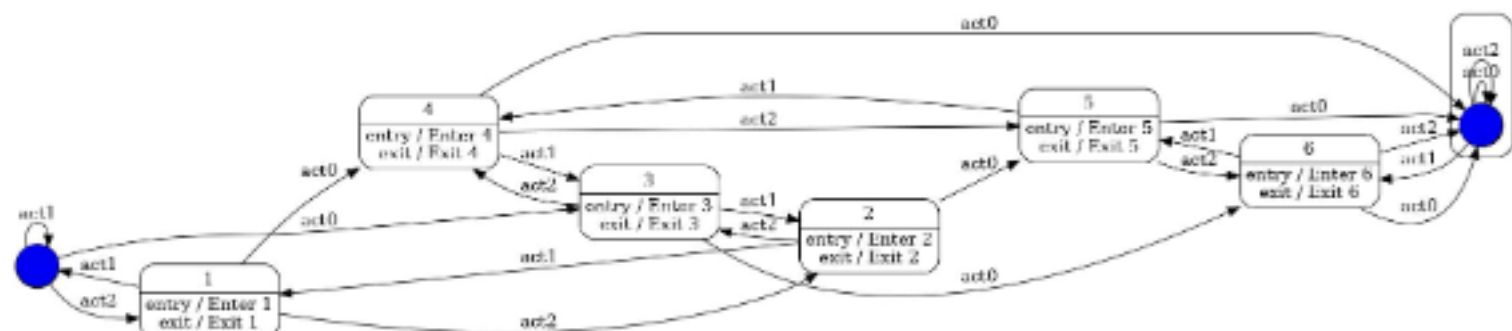
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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%

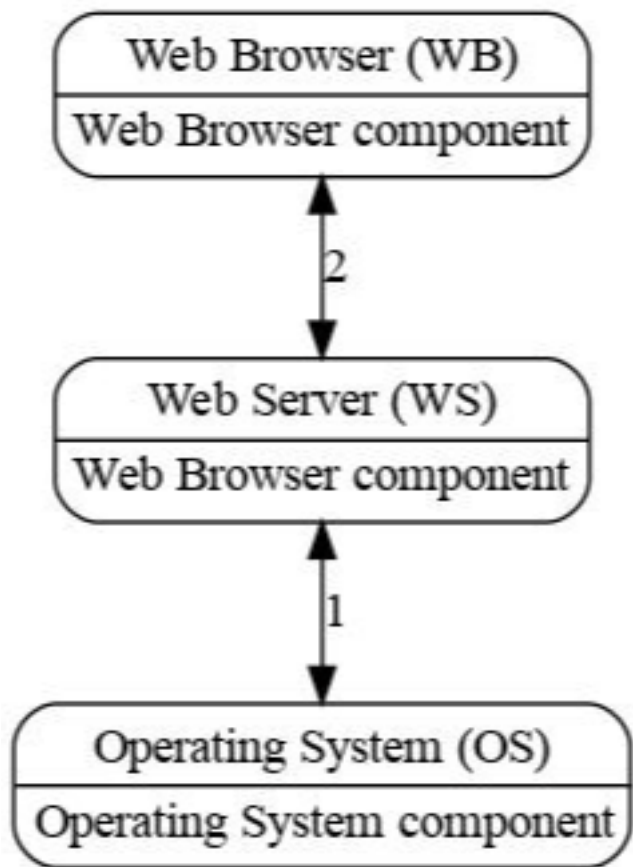
Risposta : 3

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

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

Uno scenario è una sequenza di stati.

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



1.0.08

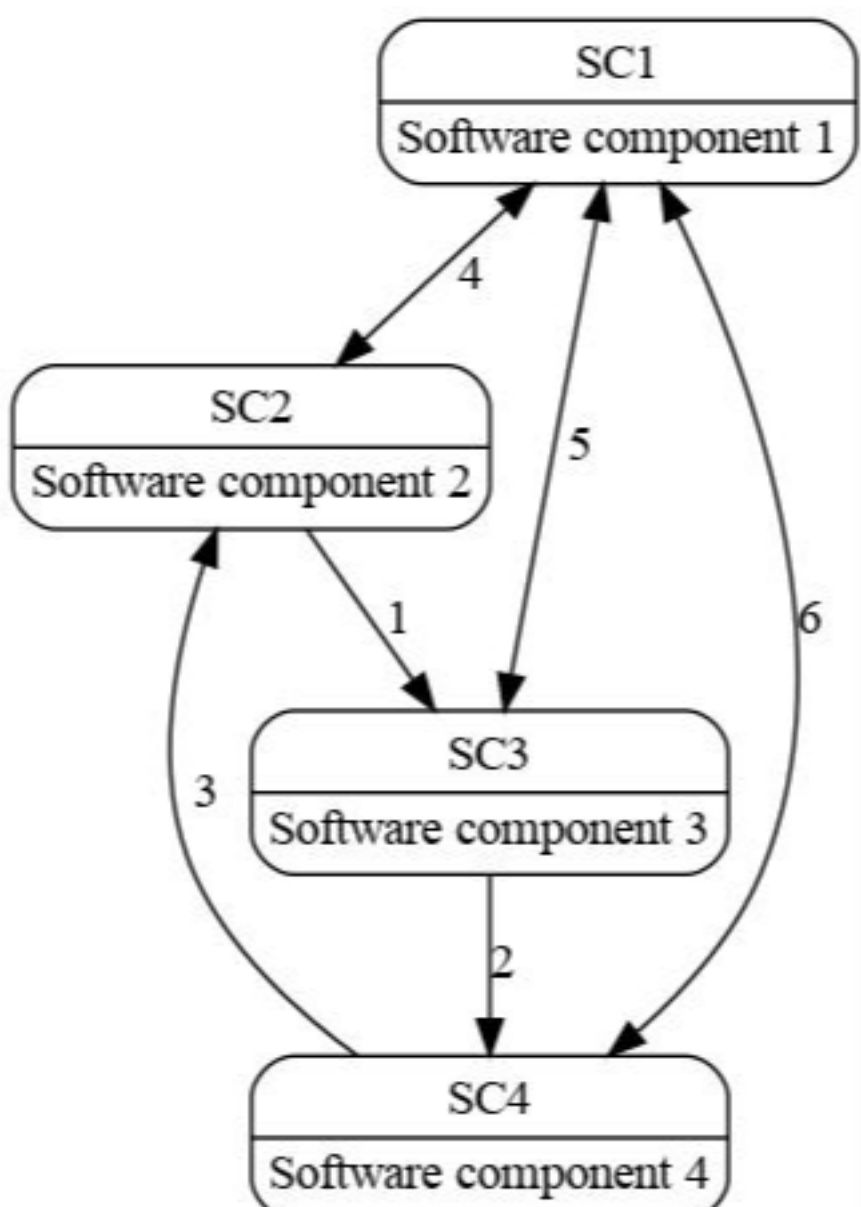
2.0.32

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?



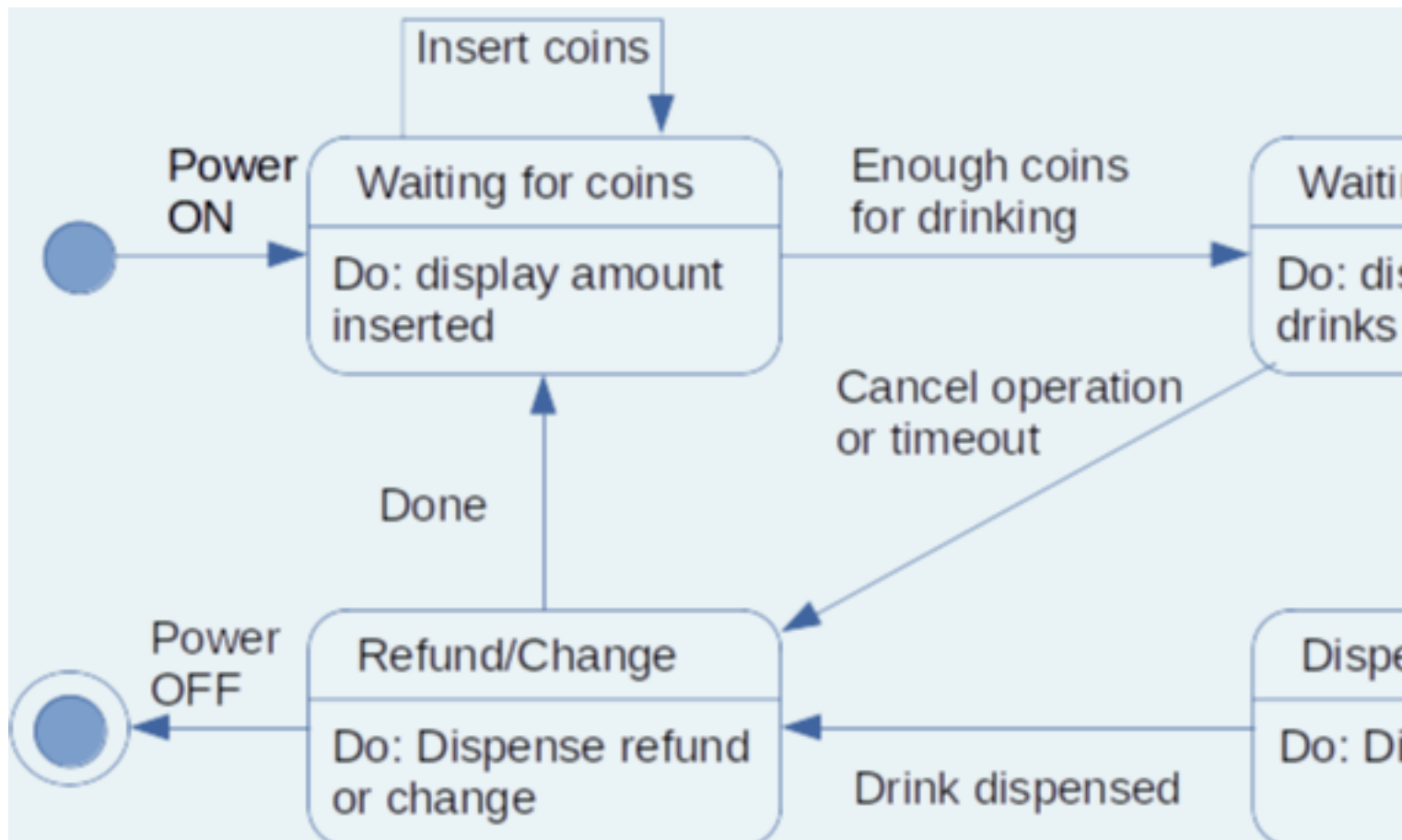
1.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] = [p, 0, 1-p, 0;0, p, 1-p, 0;p, 0, 0, 1-p;0, 0, 0, 1];Integer x; Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen 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] + 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, 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] + 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, 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 l'automa seguente:

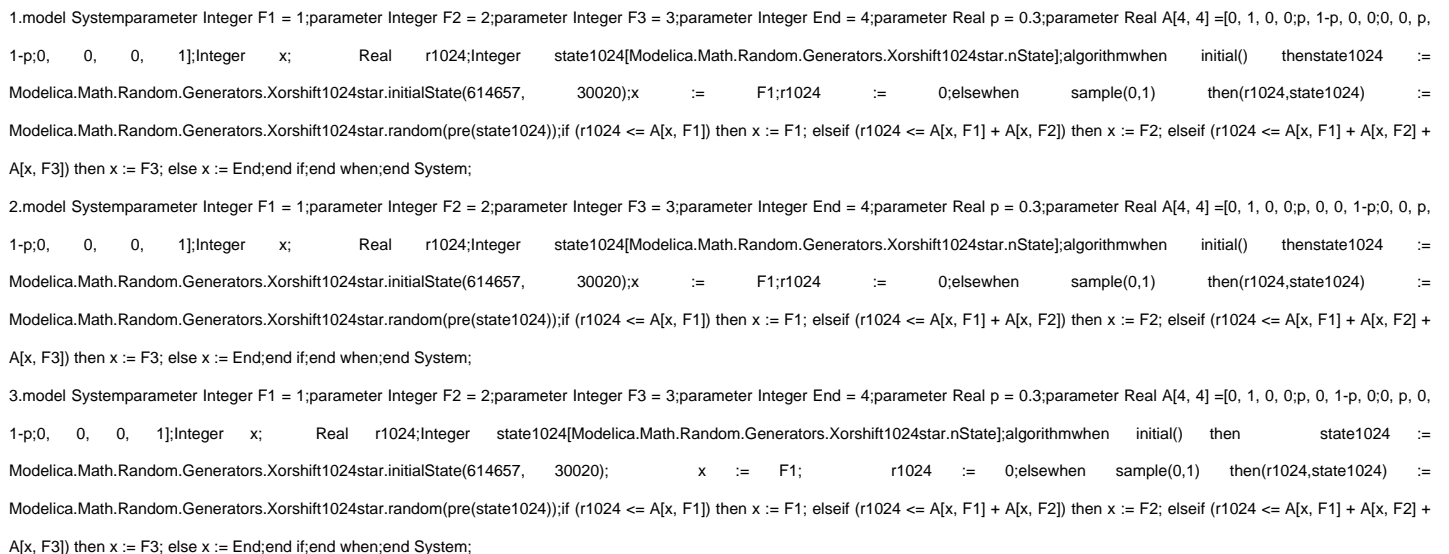
Quale dei seguenti modelli Modelica fornisce un modello ragionevole per l'automa di cui sopra.



- 1.function nextinput Integer x;output Integer y;algorithm $y := 1 + x$;end next;class SystemInteger x;initial equation $x = 0$;equationwhen sample(0, 1) then $x = \text{next}(\text{pre}(x))$;end when;end System;
- 2.function nextinput Integer x;output Integer y;algorithm $y := x$;end next;class SystemInteger x;initial equation $x = 0$;equationwhen sample(0, 1) then $x = \text{next}(\text{pre}(x))$;end when;end System;
- 3.function nextinput Integer x;output Integer y;algorithm $y := 1 - x$;end next;class SystemInteger x;initial equation $x = 0$;equationwhen sample(0, 1) then $x = \text{next}(\text{pre}(x))$;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?

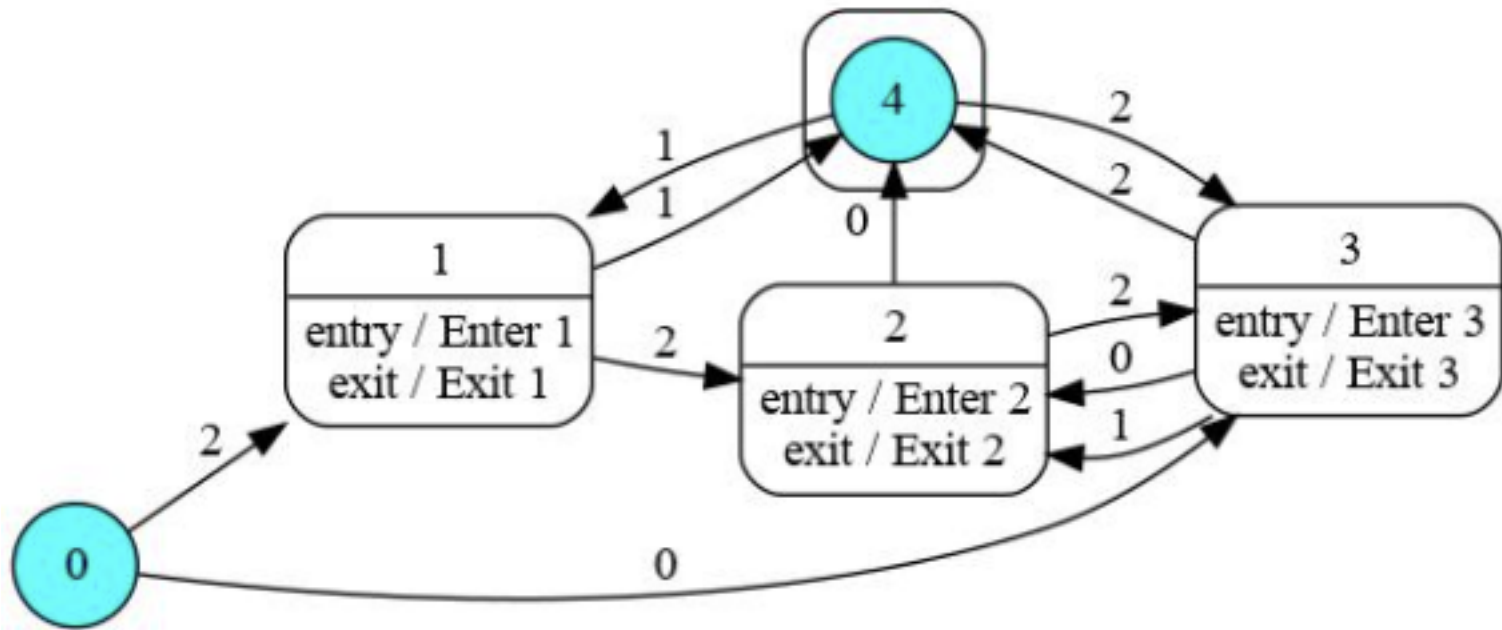


Test case 3: act2 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



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.



1. $(1 - p)/p$
2. $1/(p \cdot (1 - p))$
3. $1/(1 - p)$

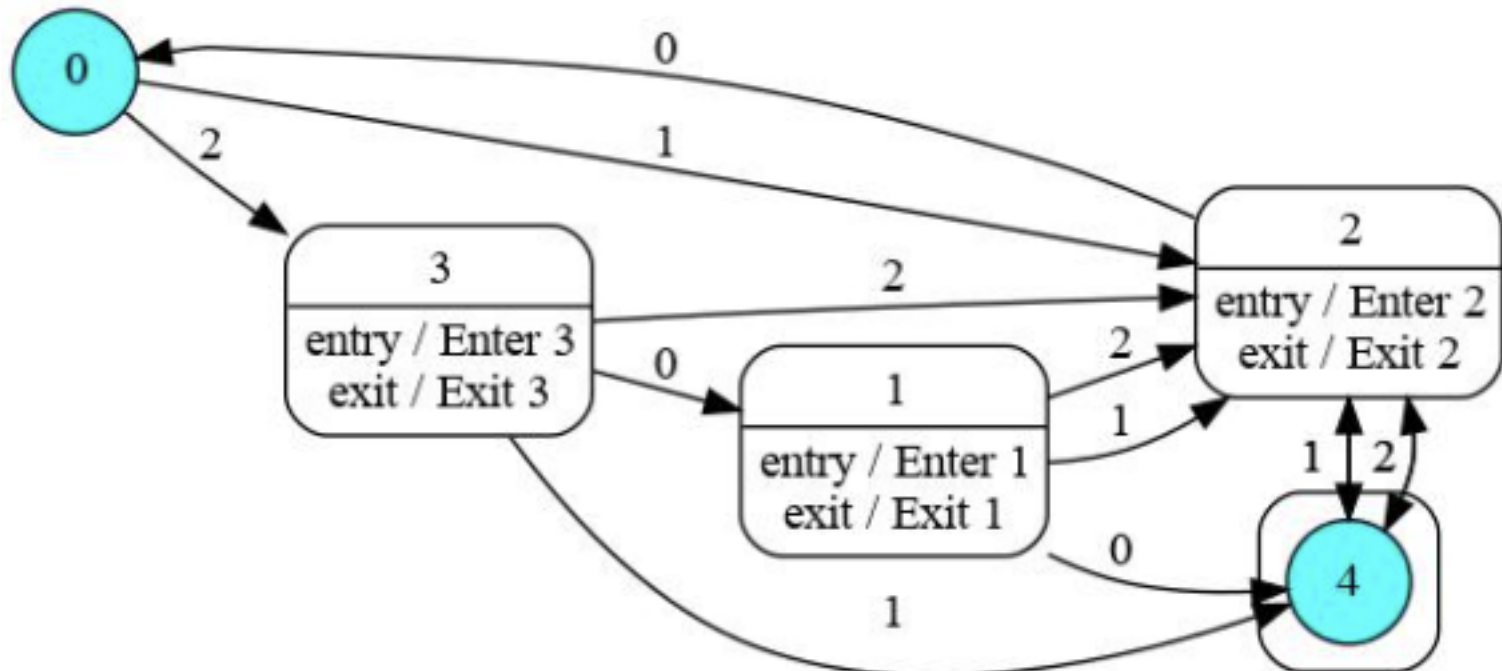
Risposta : 3

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

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

Uno scenario è una sequenza di stati.

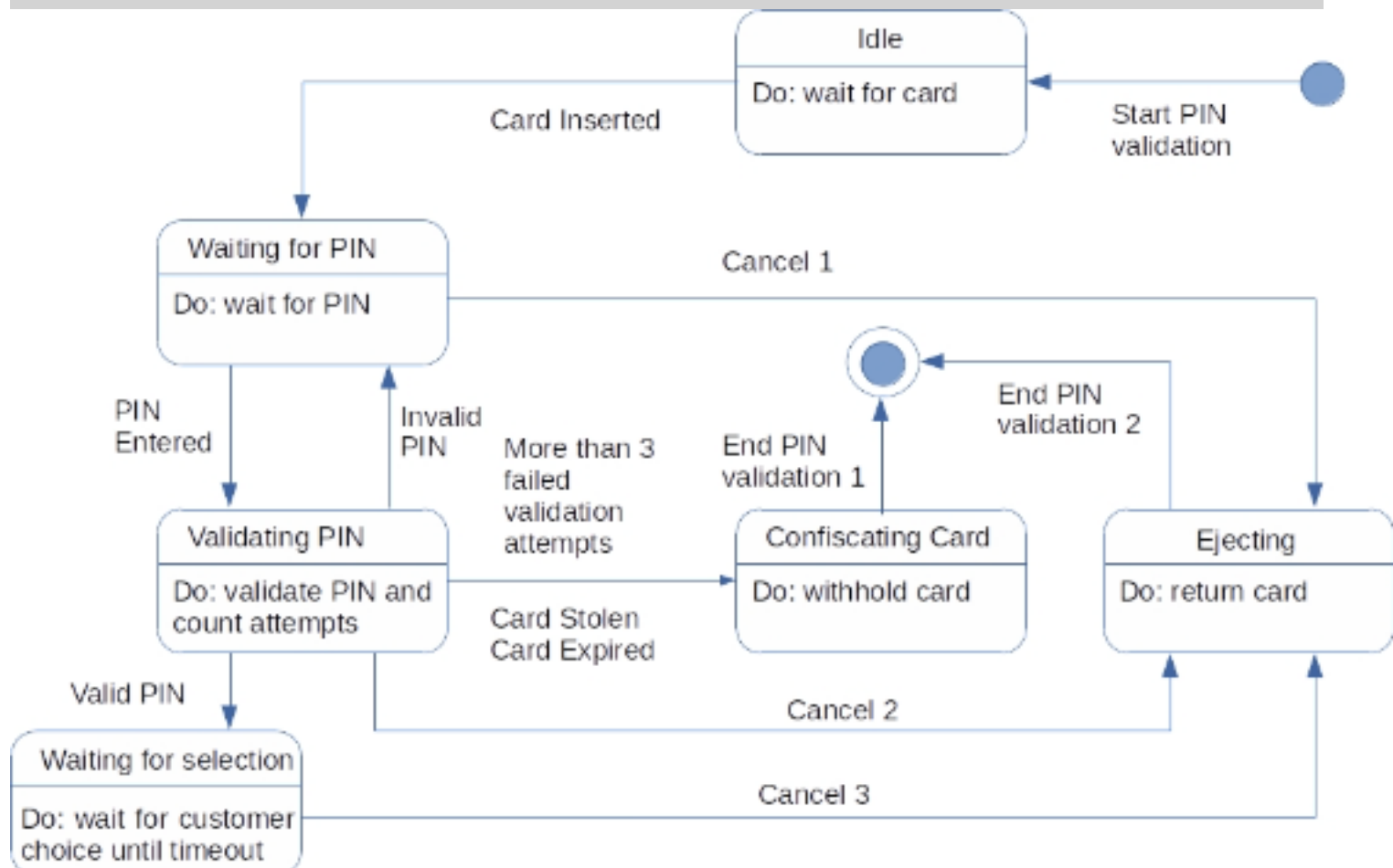
Qual'è la probabilità dello scenario: 1, 3 ? In altri termini, qual'è la probabilità che non sia necessario ripetere nessuna fase?



1. 0.24
2. 0.14
3. 0.56

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

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

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

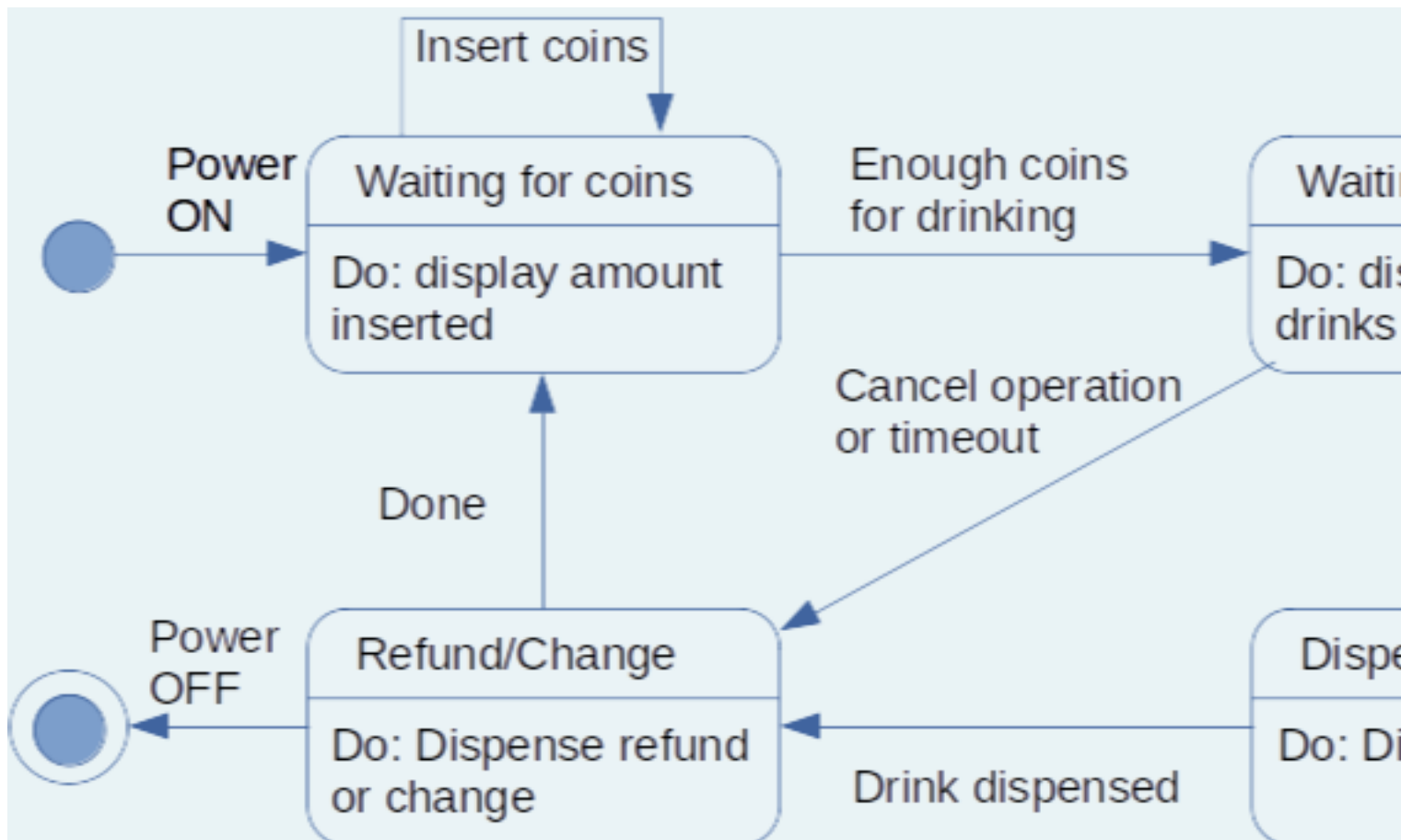
Risposta : 2

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 act1 act2 act1 act2 act2 act2

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



1.80%

2.60%

3.100%

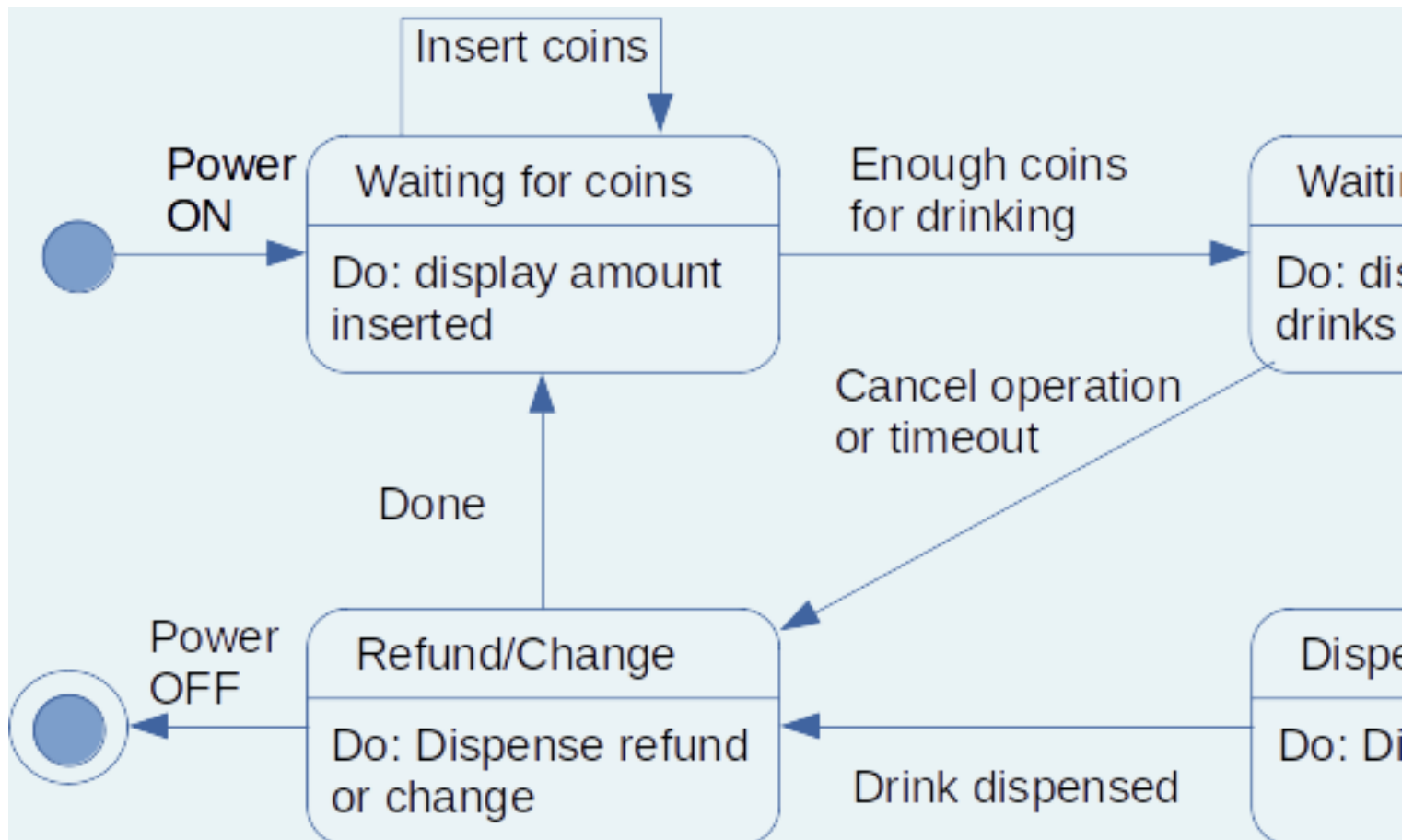
Risposta : 2

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

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

Uno scenario è una sequenza di stati.

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



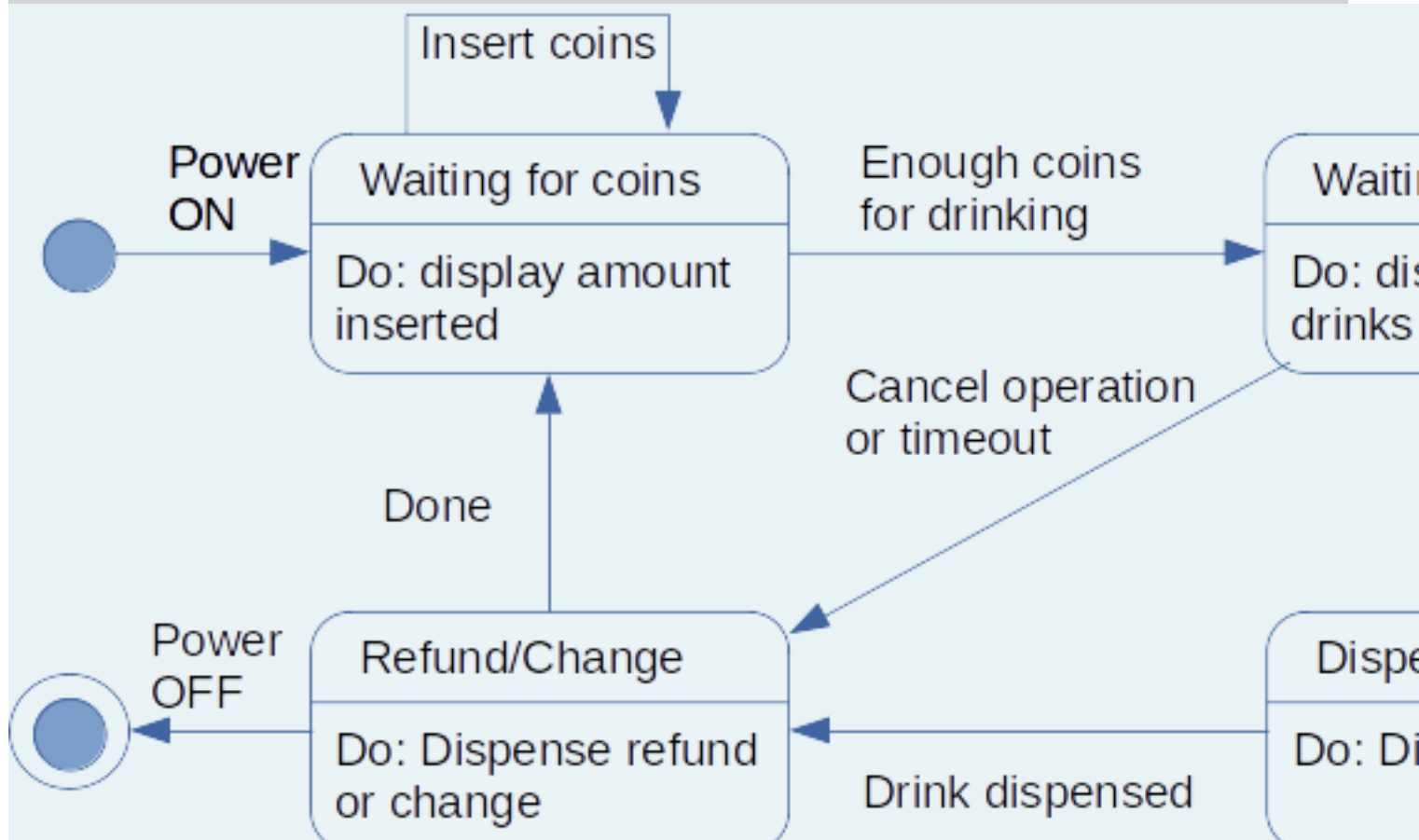
1.0.12

2.0.32

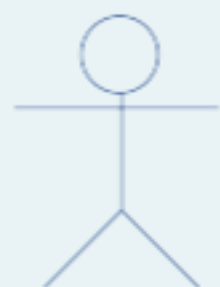
3.0.08

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 ?



Customer



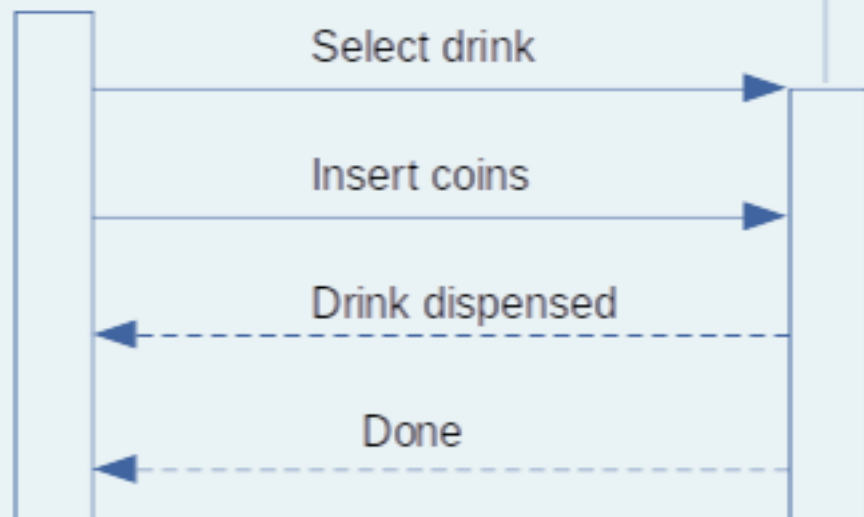
Vending Machine

Select drink

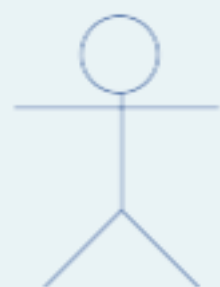
Insert coins

Drink dispensed

Done



Customer



Vending
Machine

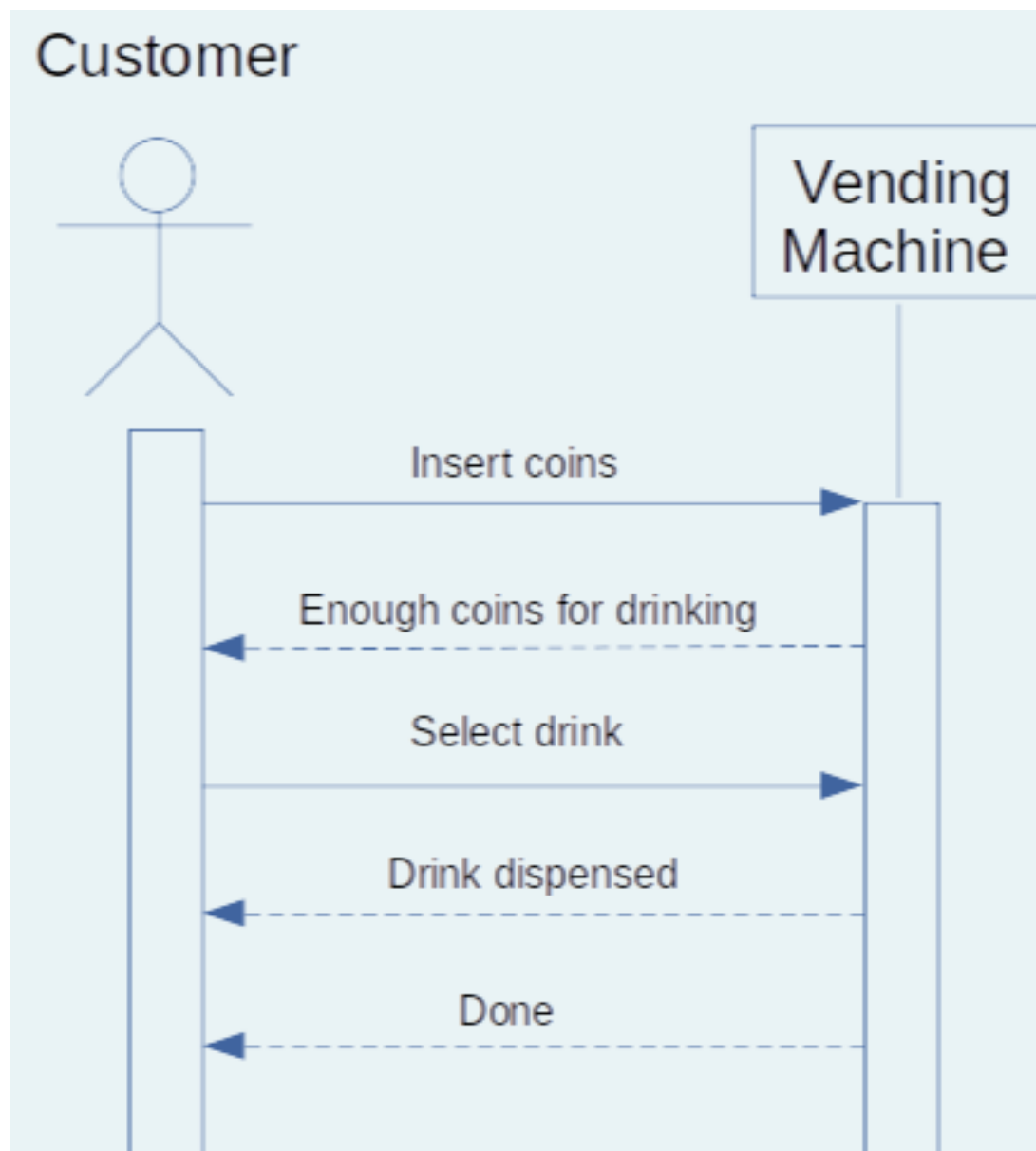
Insert coins

Enough coins for drinking

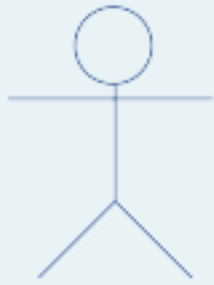
Select drink

Drink dispensed

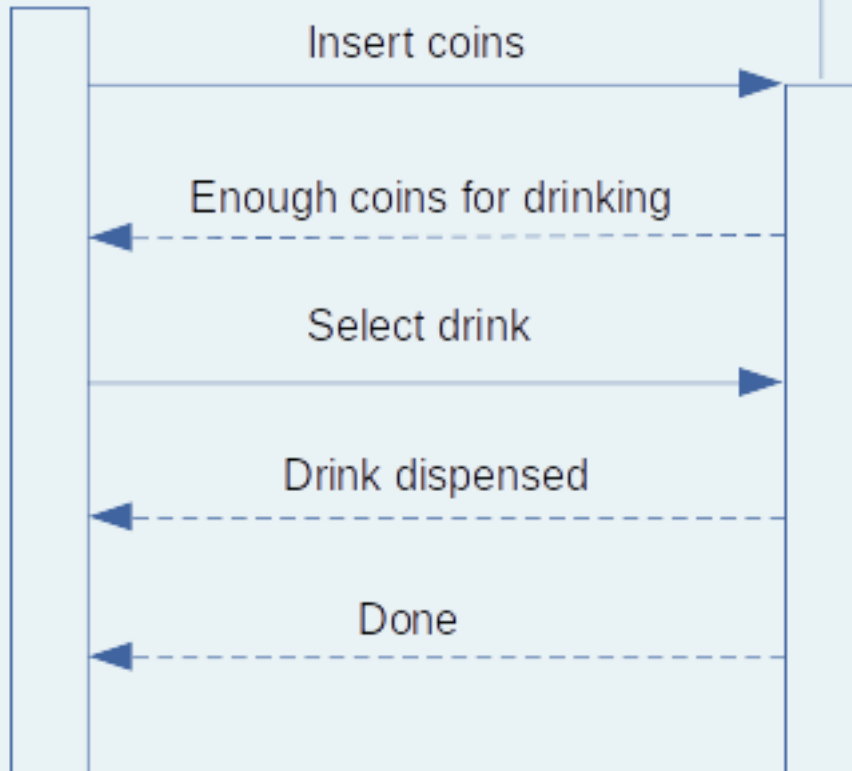
Done



Customer



Vending Machine



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.

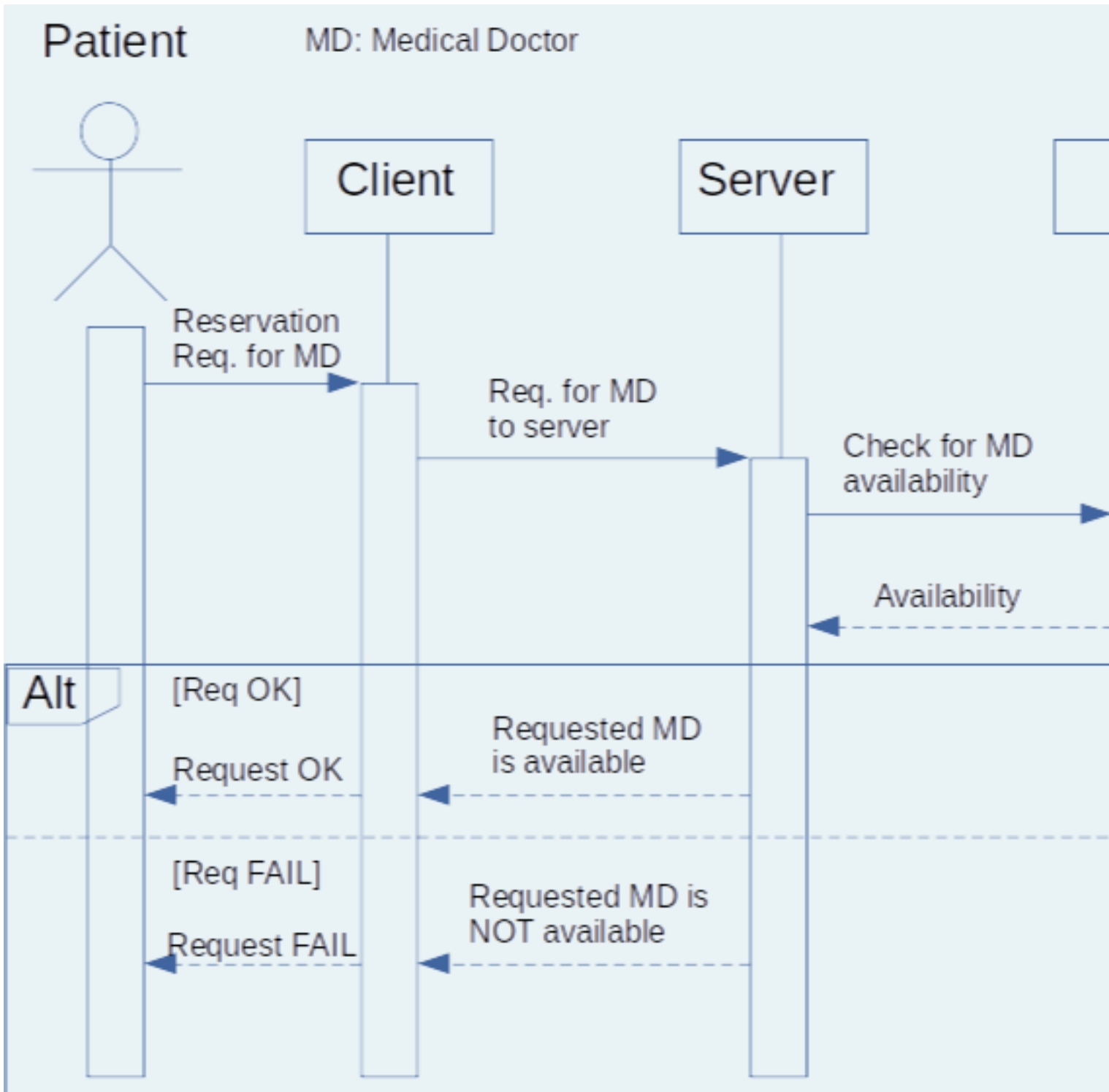
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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?



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.

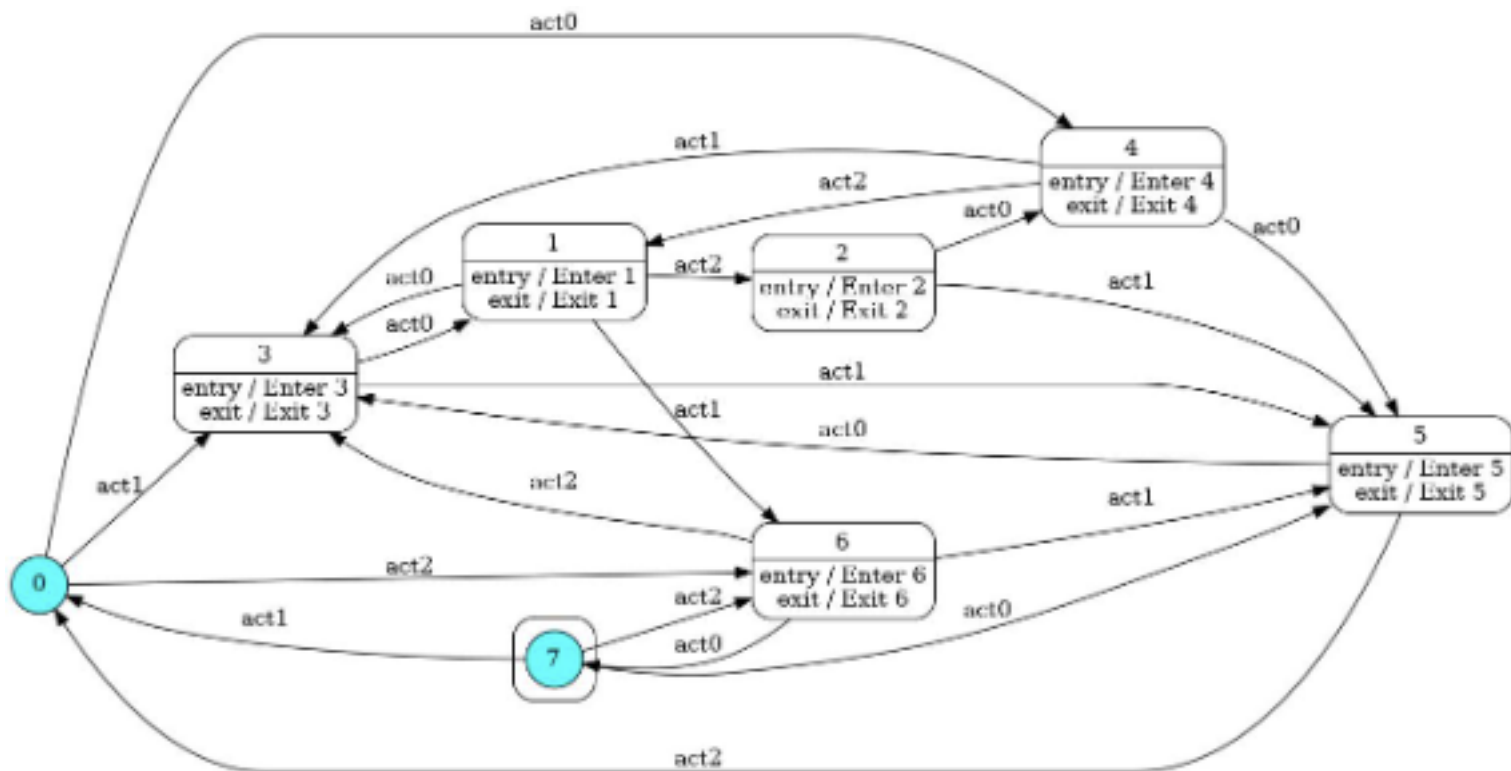
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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?



- 1.75%
- 2.60%
- 3.90%

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.

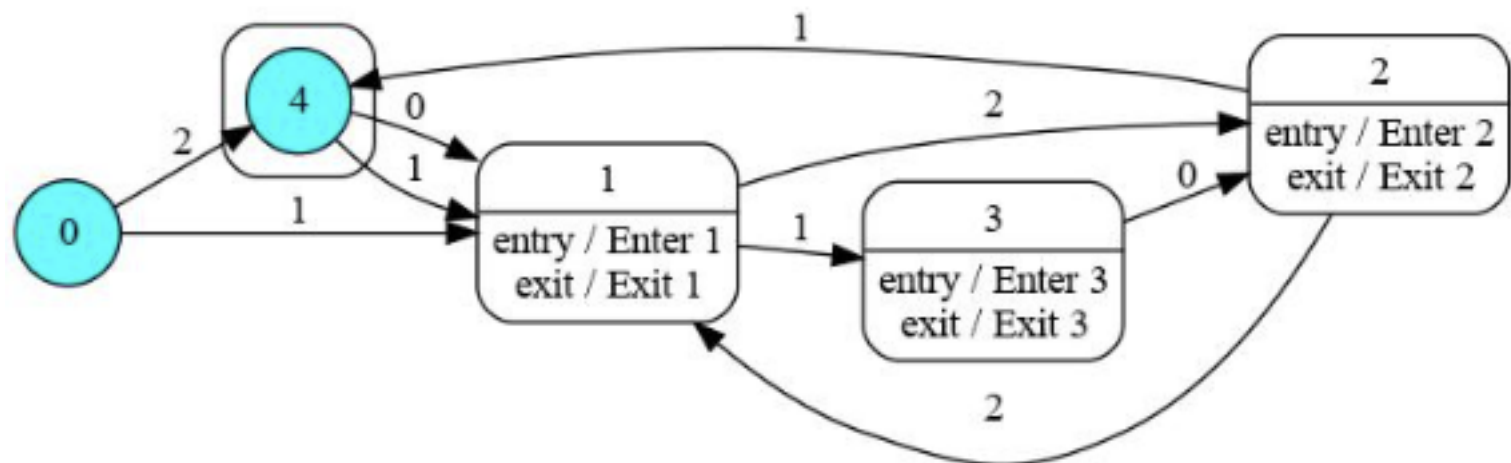
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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

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

Test case 3: act1 act1

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



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

Risposta : 3

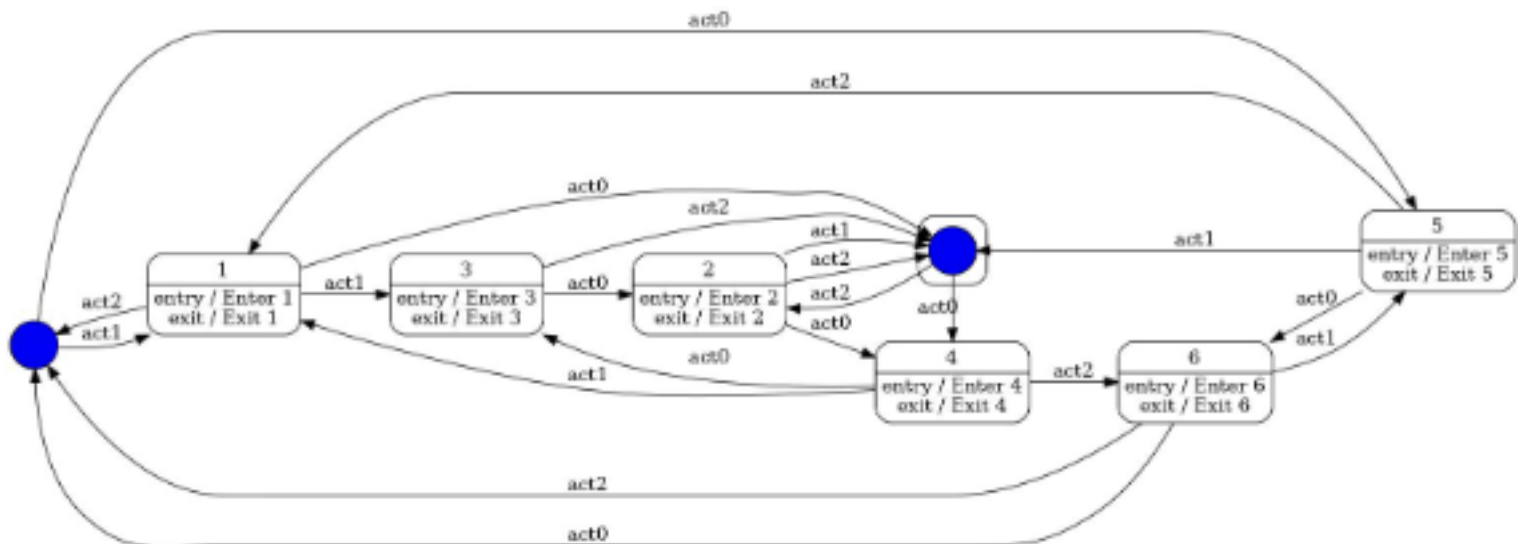
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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 act0 act1 act1 act2 act1 act2 act0 act0 act0 act2

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



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

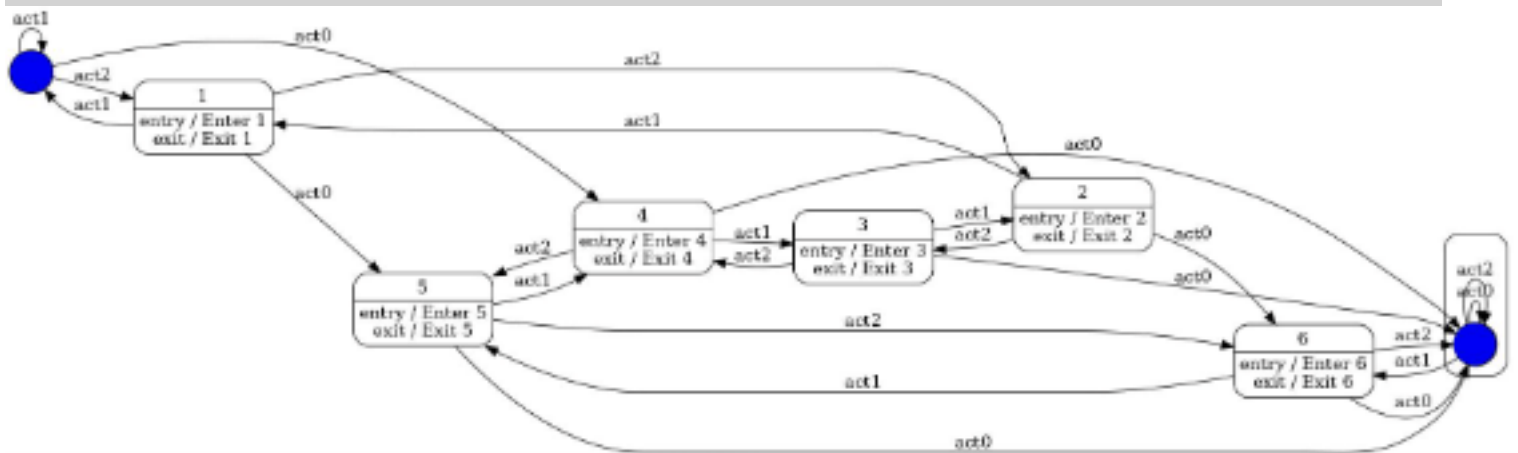
Risposta : 3

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

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

Uno scenario è una sequenza di stati.

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



- 1.0.03
- 2.0.07
- 3.0.27

Risposta : 1

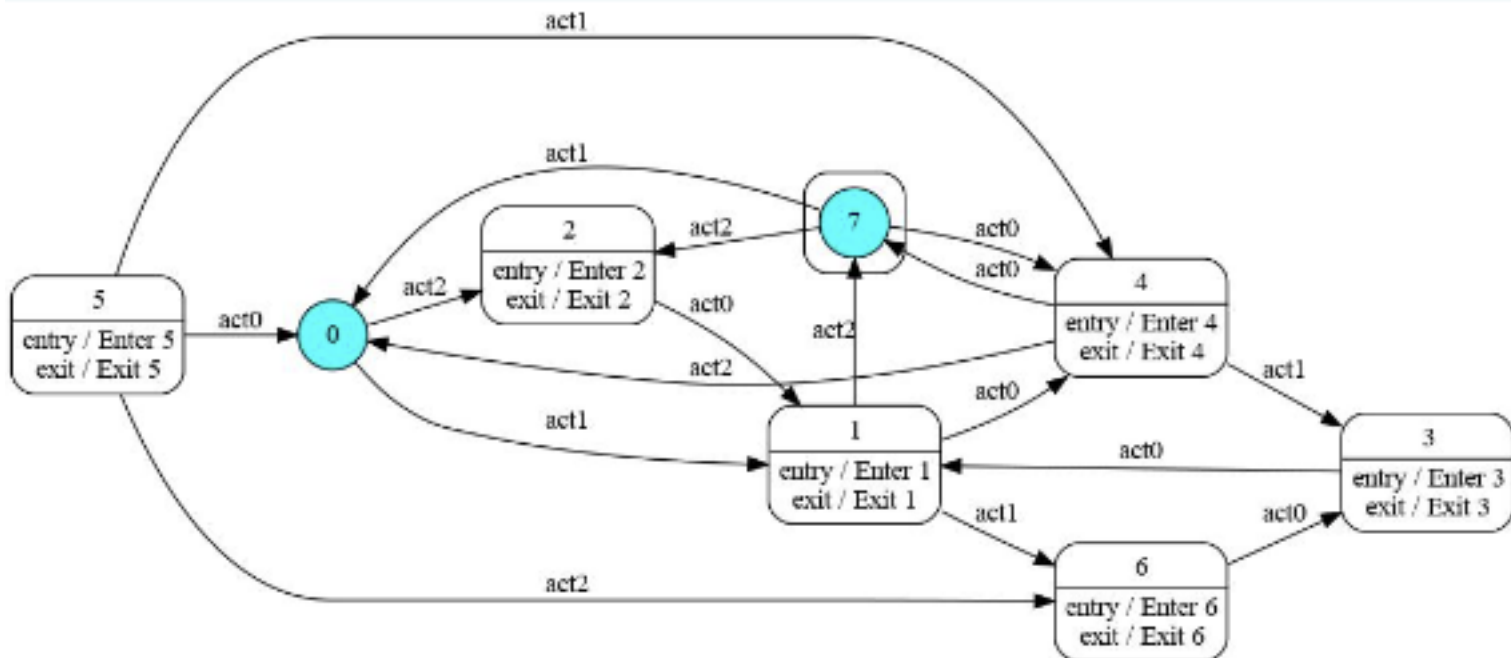
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il tempo necessario per completare la fase x è $\text{time}(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase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

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

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

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

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)

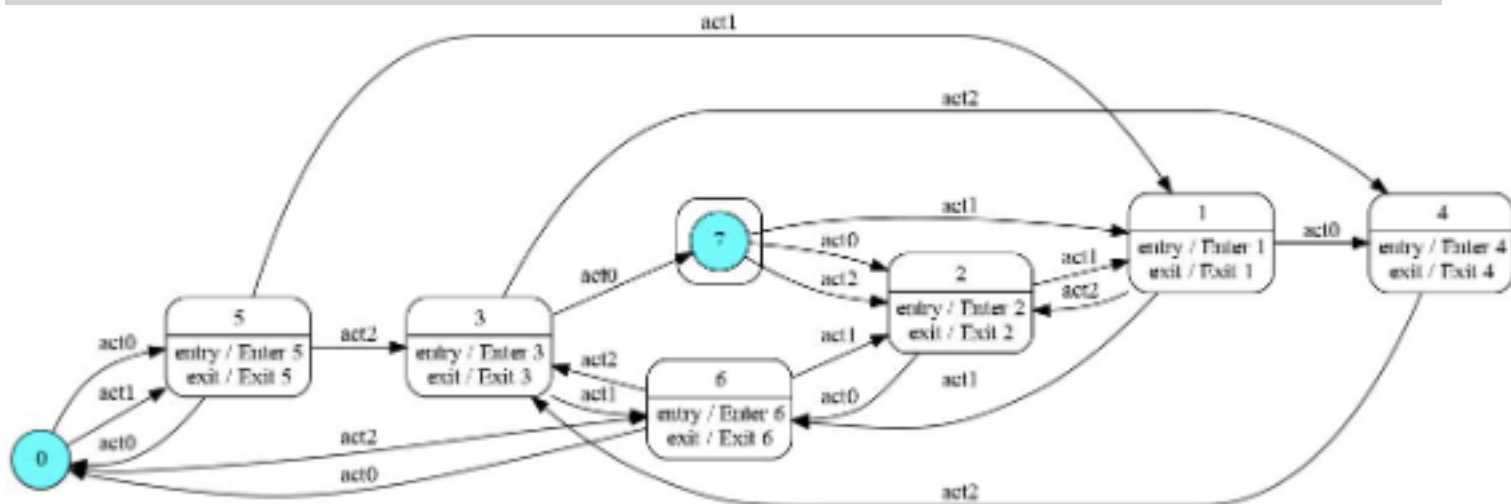
Risposta : 3

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

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

Uno scenario è una sequenza di stati.

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



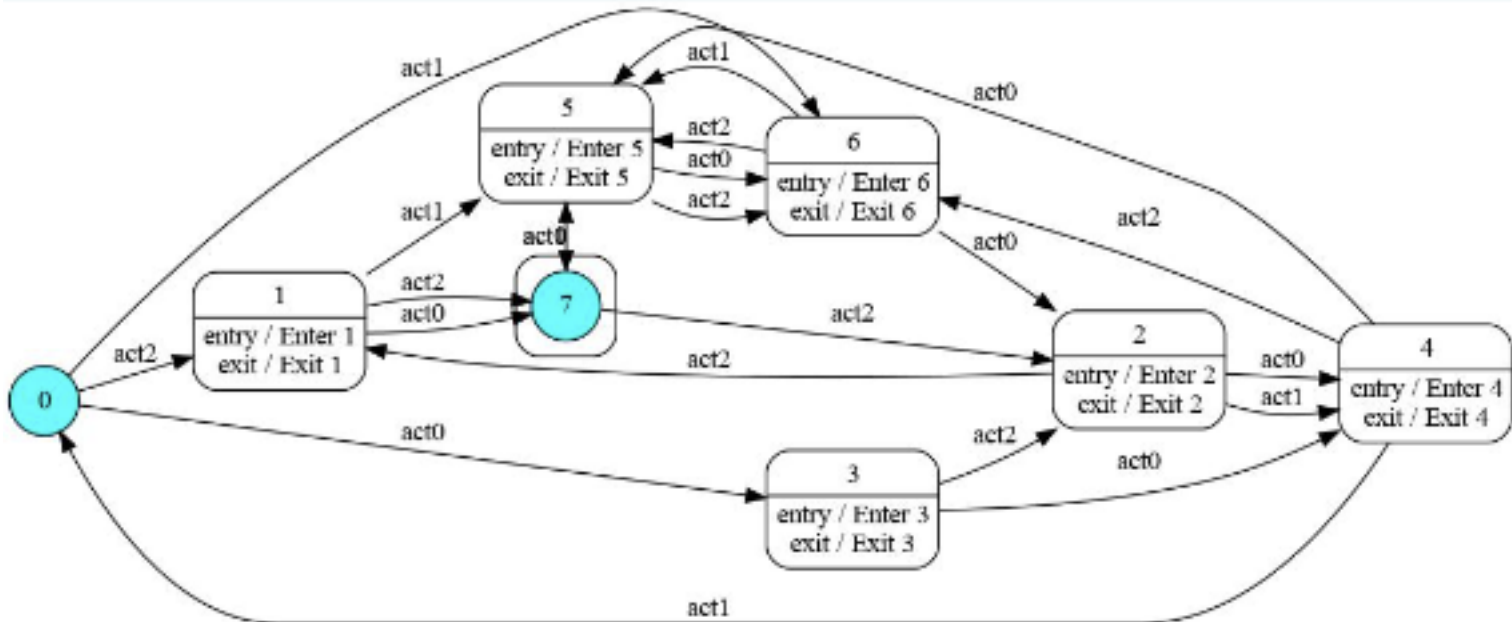
1.0.12

2.0.28

3.0.42

Risposta : 2

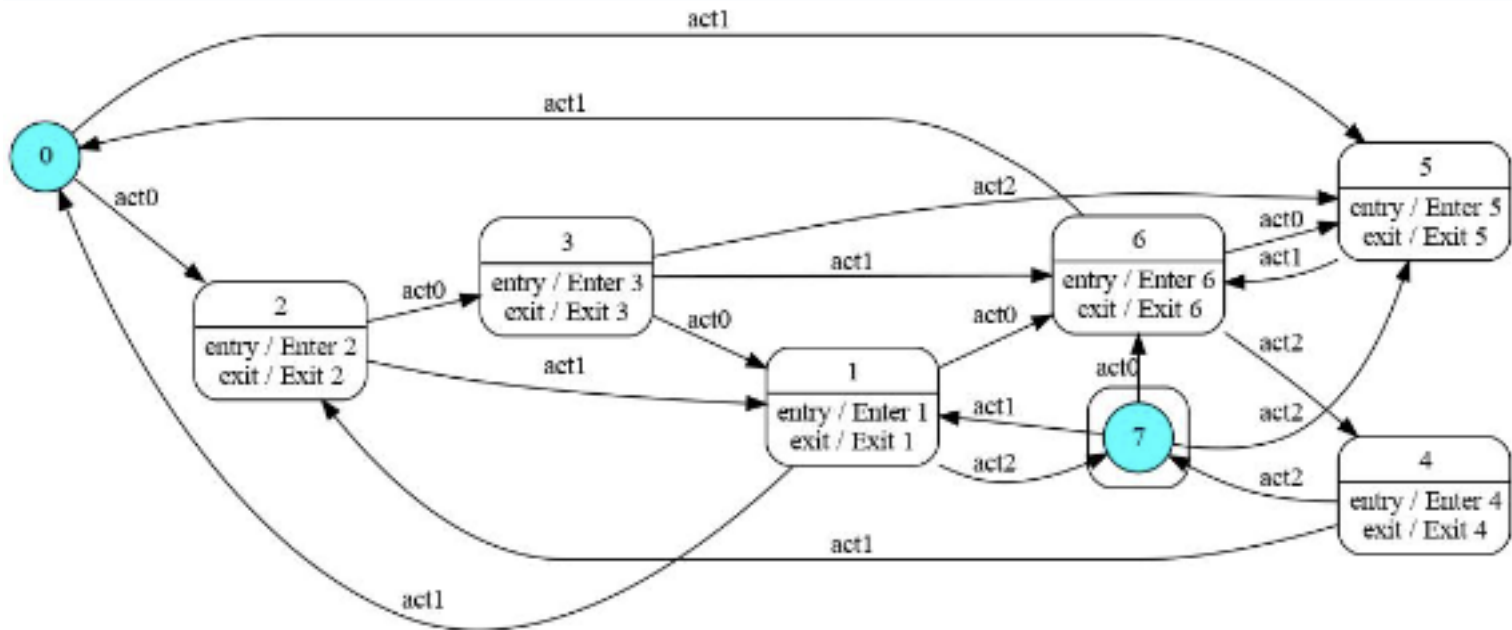
Quale delle seguenti frasi è corretta riguardo all'activity diagram in figura ?



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

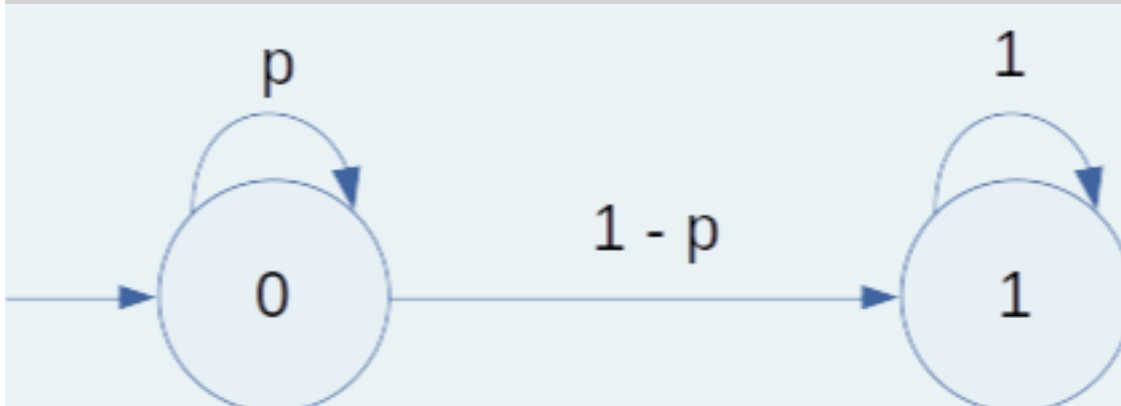
Quale dei seguenti modelli Modelica rappresenta lo state diagram in figura ?



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

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

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

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

Risposta : 3

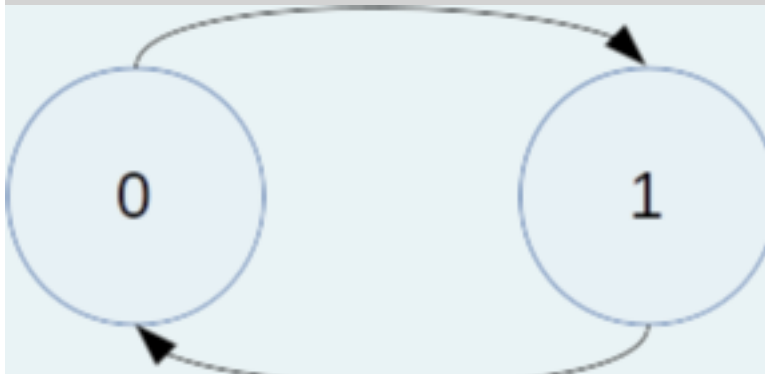
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il costo dello stato (fase) x è $c(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappresenta il completamento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

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

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

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

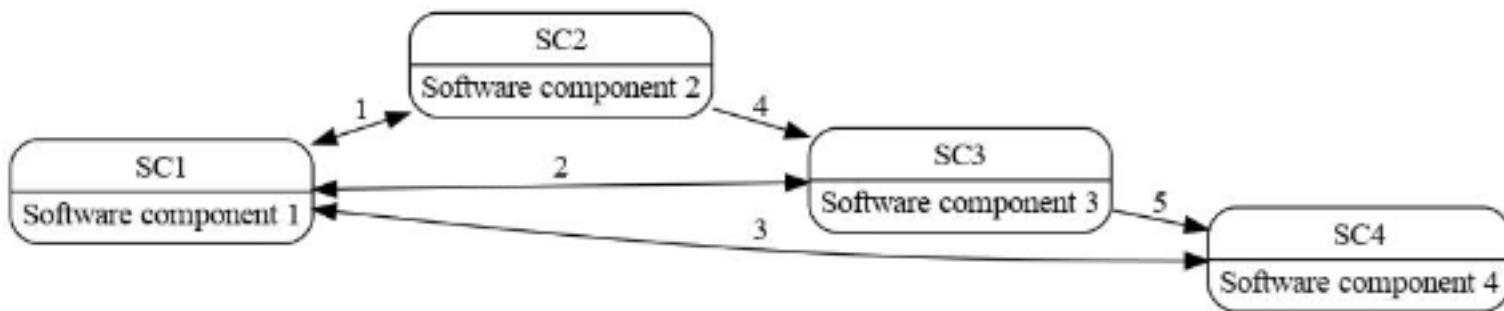
Quale delle seguenti formule calcola il valore atteso del costo per completare il processo software di cui sopra



1. $c(0) \cdot (1 - p) / p$
2. $c(0) / (1 - p)$
3. $c(0) / (p \cdot (1 - p))$

Risposta : 2

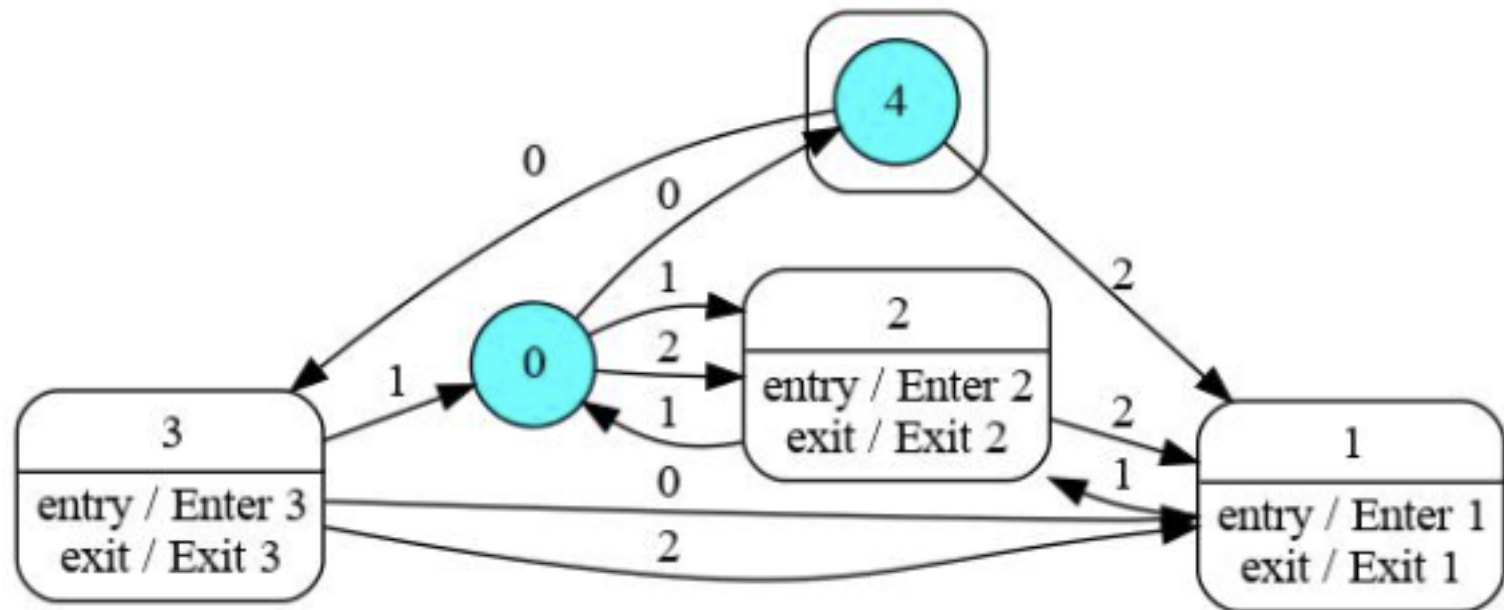
Quale delle seguenti frasi è corretta riguardo al Sequence Diagram in figura ?



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 specifico medico e, dopo una verifica sul database, riceve conferma dal server della disponibilità o meno del medico richiesto.

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

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

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

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.

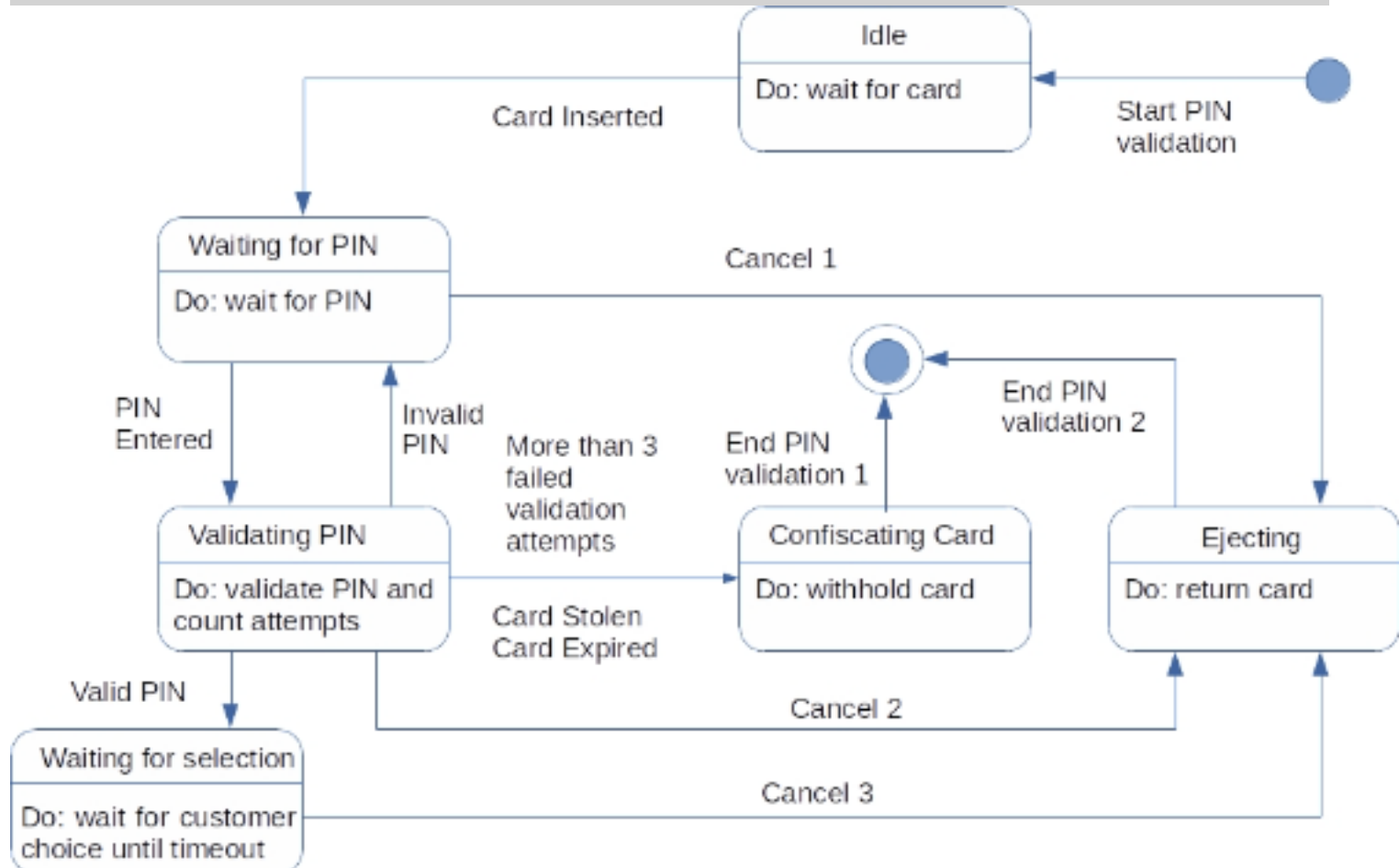
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

Test case 2: act0

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

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



1.90%

2.100%

3.50%

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.

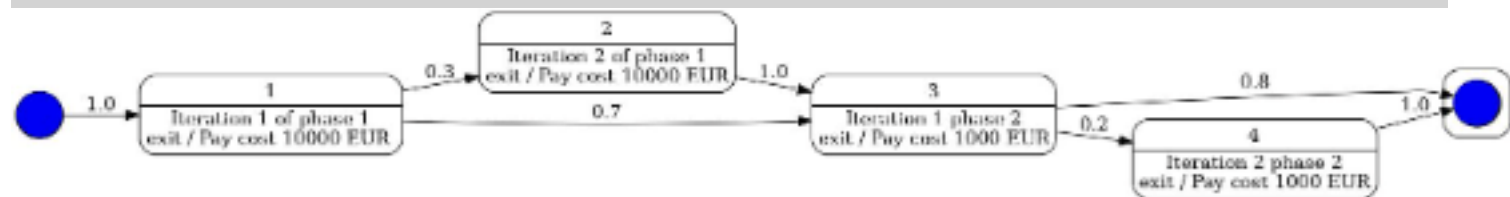
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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

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

Test case 3: act1 act1 act2 act0 act1 act0

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



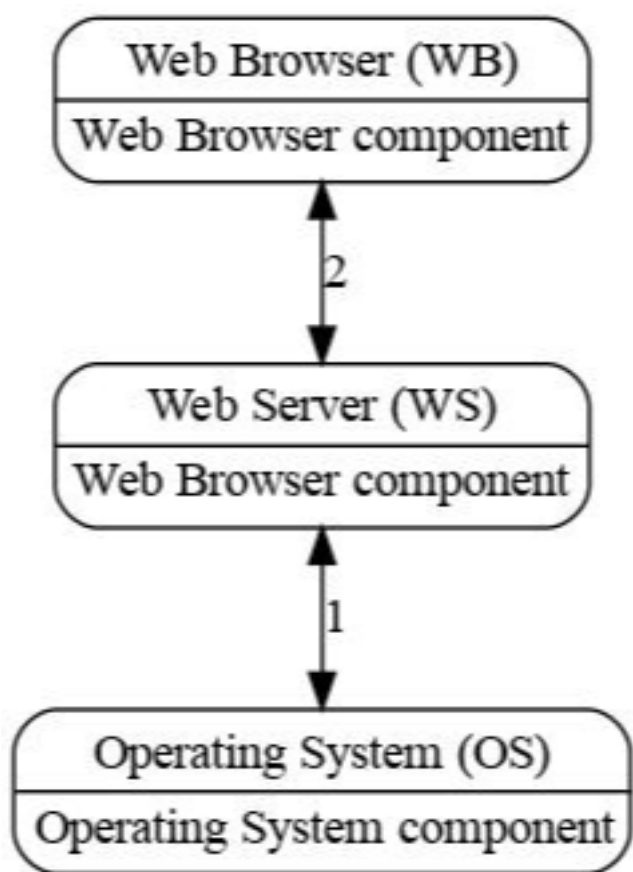
1.State coverage: 60%

2.State coverage: 100%

3.State coverage: 75%

Risposta : 3

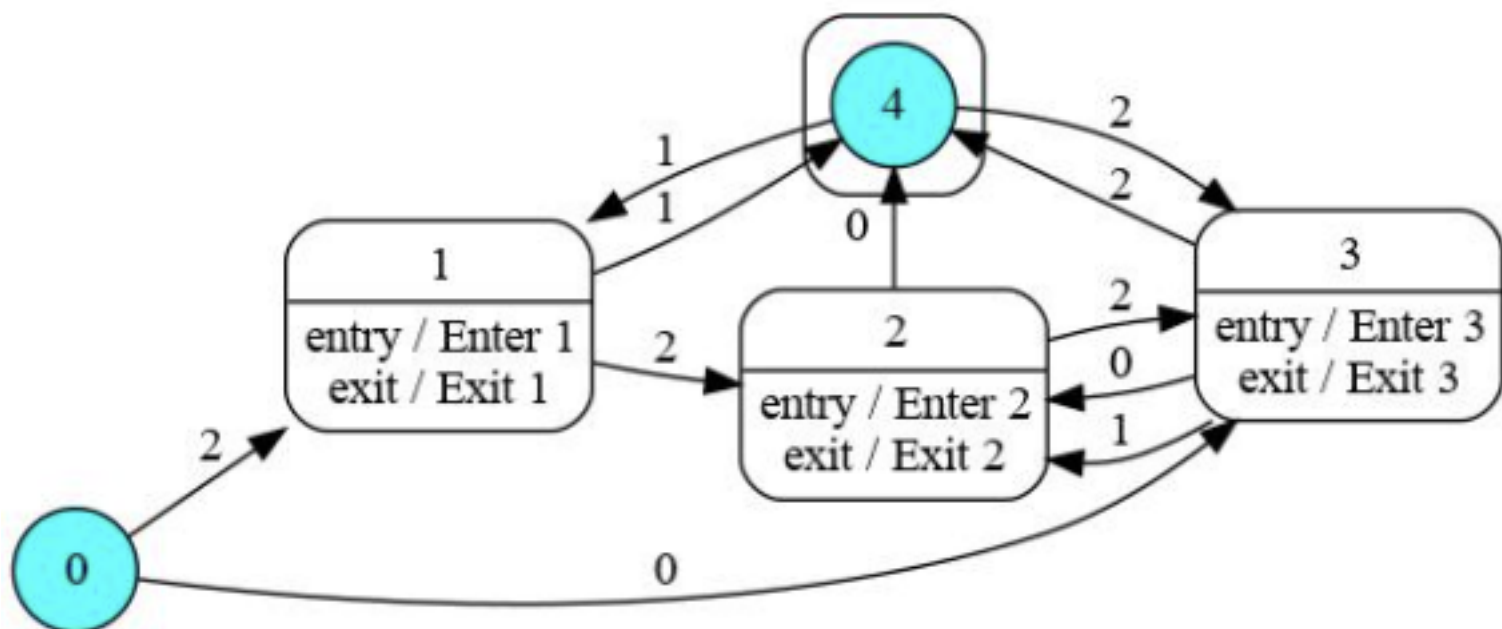
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?



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.

Risposta : 1

Quale delle seguenti frasi è corretta riguardo all'activity 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.

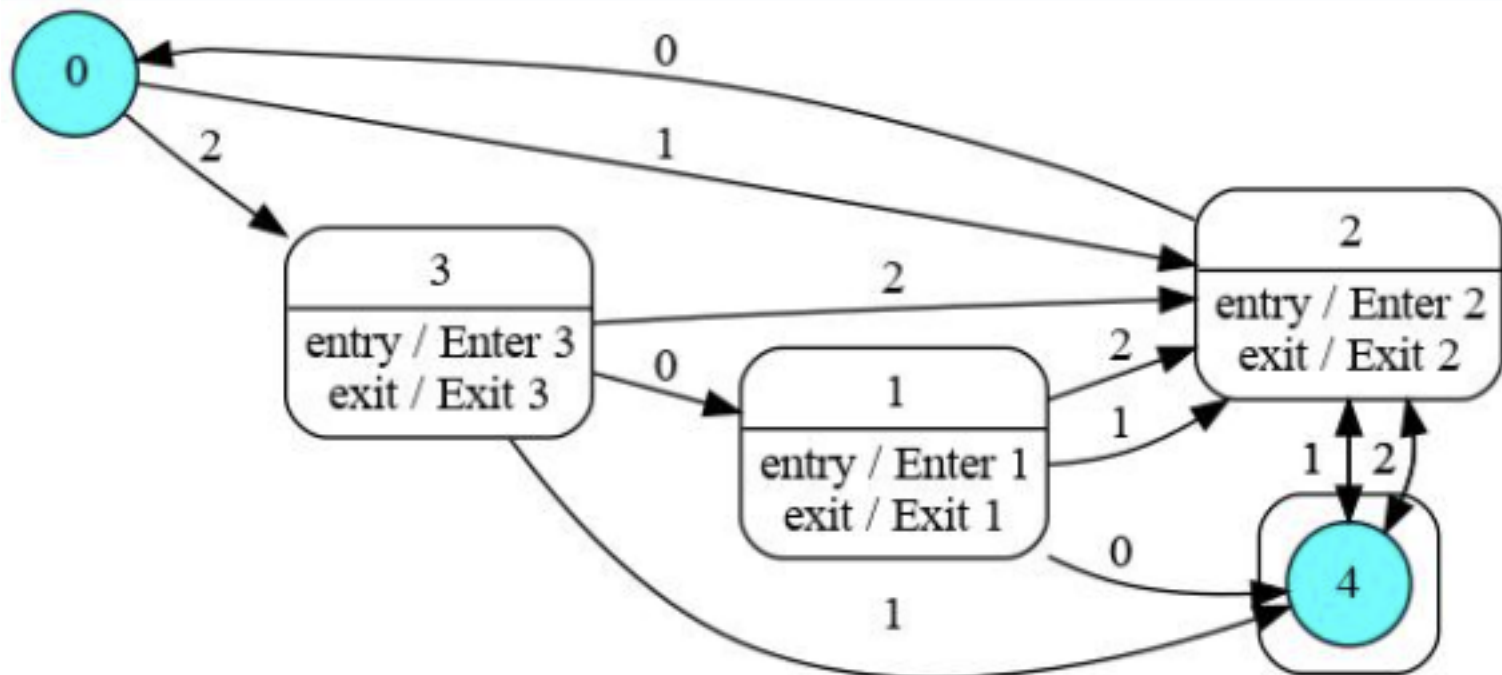
Risposta : 3

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) raggiunti almeno una volta.

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

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

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



1.80%

2.90%

3.60%

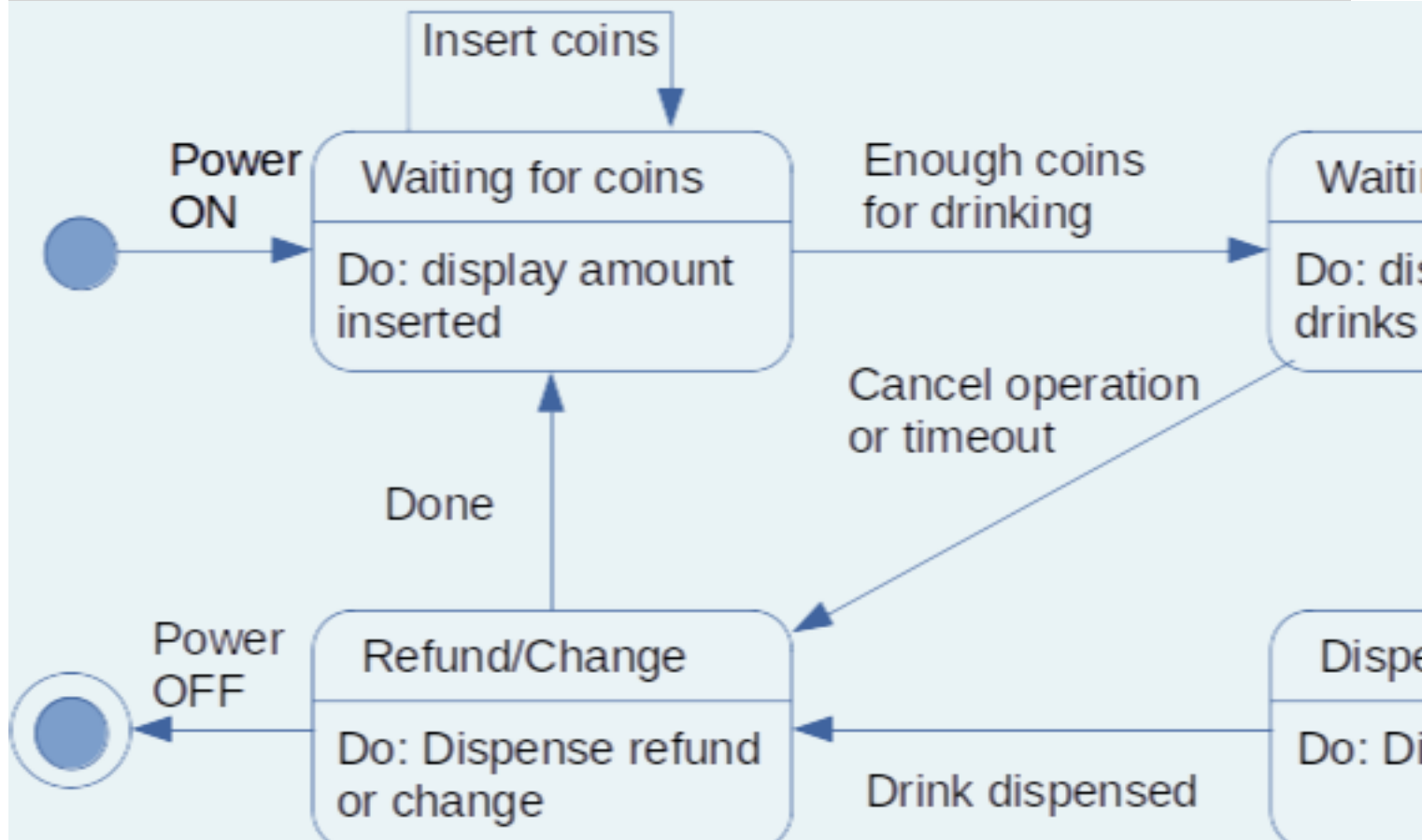
Risposta : 2

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) rggunti 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%

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 <= 2)

        {return ;}

        else {y = y + 1; goto LOOP;}

} /* f() */

int main() { int i, y; int x[N];

// define test cases

    x[0] = 3;

// testing

    for (i = 0; i < N; i++) {

        y = f(x[i]); /* function under testing

        assert(y == (abs(x[i]) <= 2) ? 0 : (abs(x[i]) - 2)); // oracle

    }

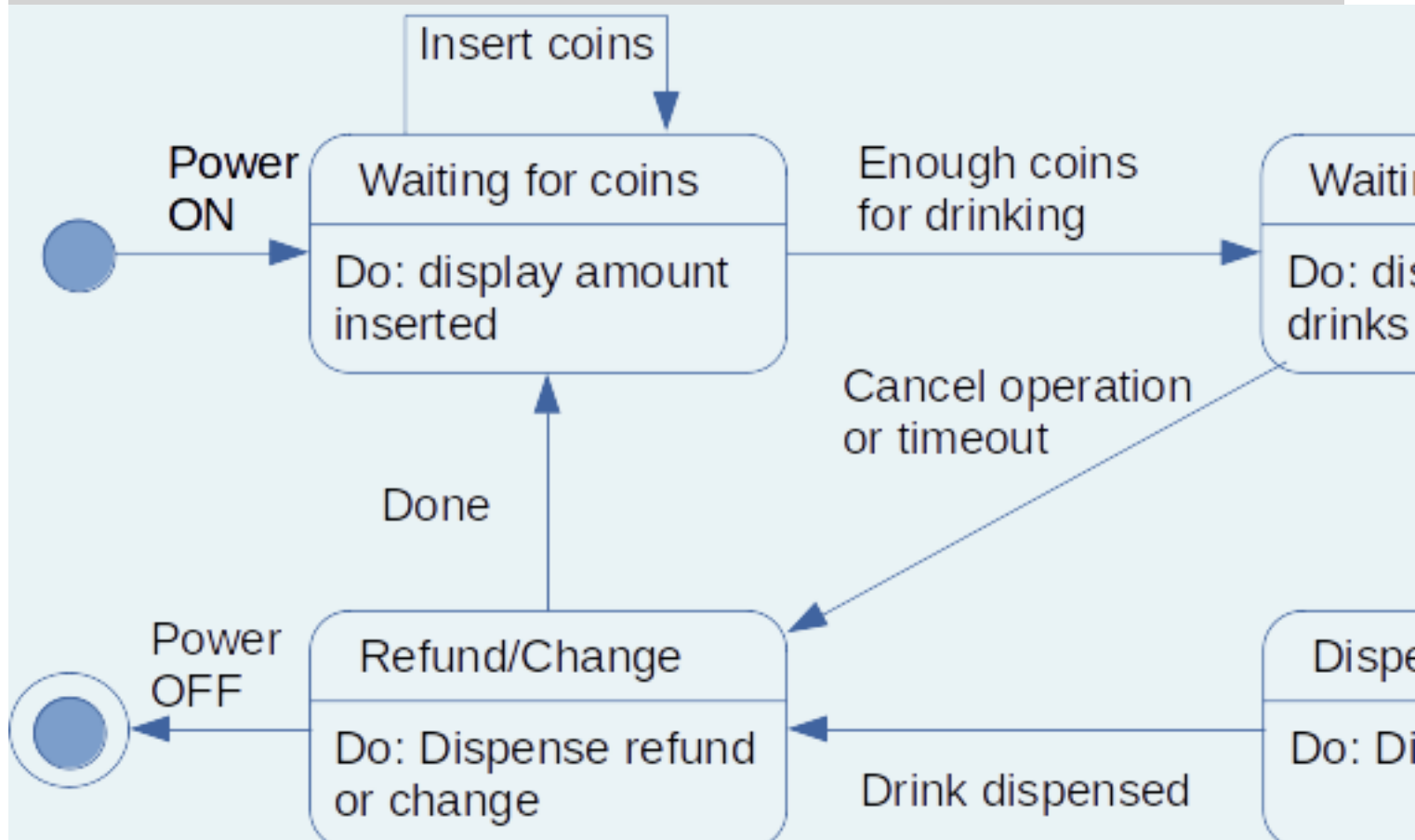
    printf("All %d test cases passed\n", N);

    return (0);

}
```

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

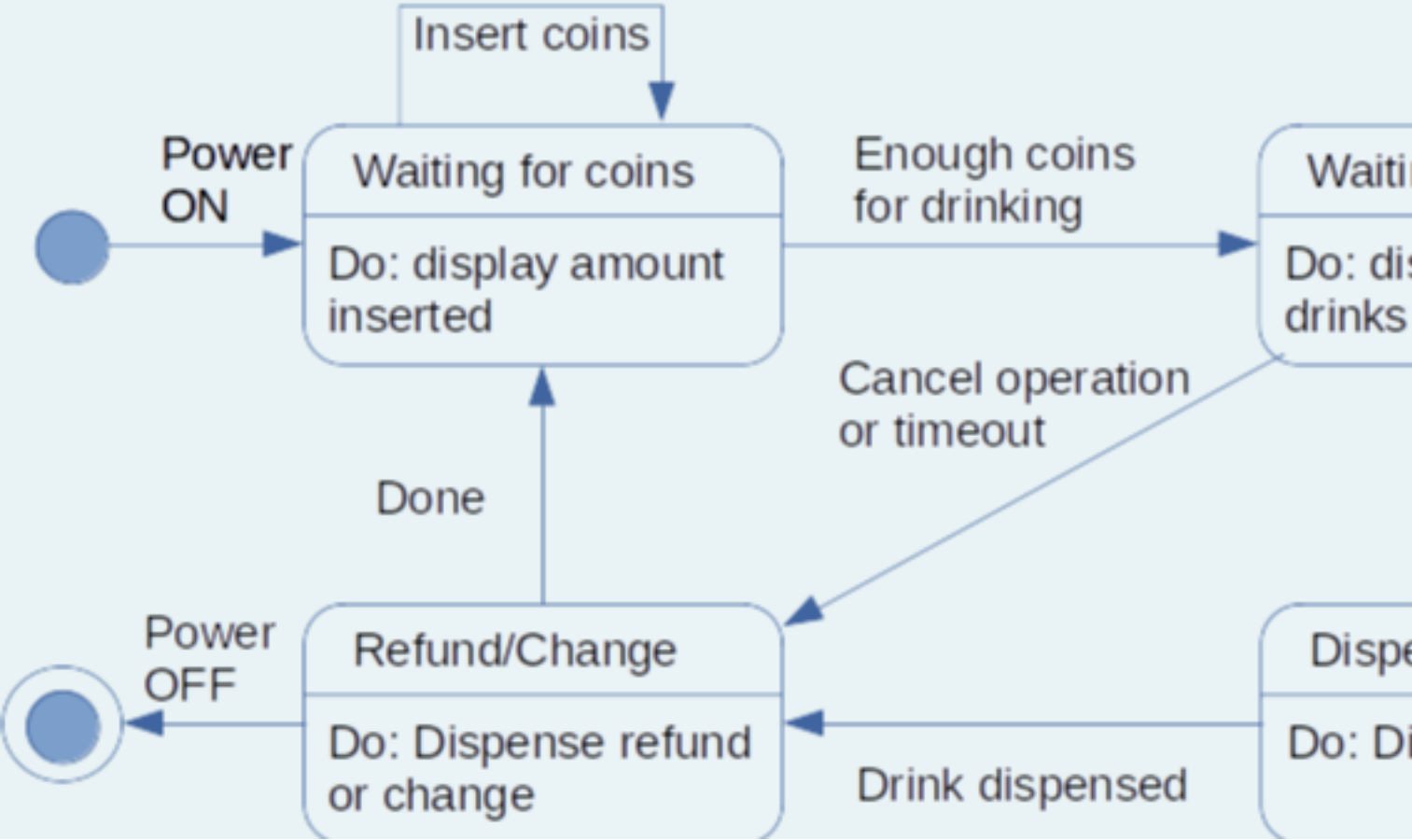
Quale delle seguenti è la branch coverage conseguita?



1.50%
2.100%
3.80%

Risposta : 2

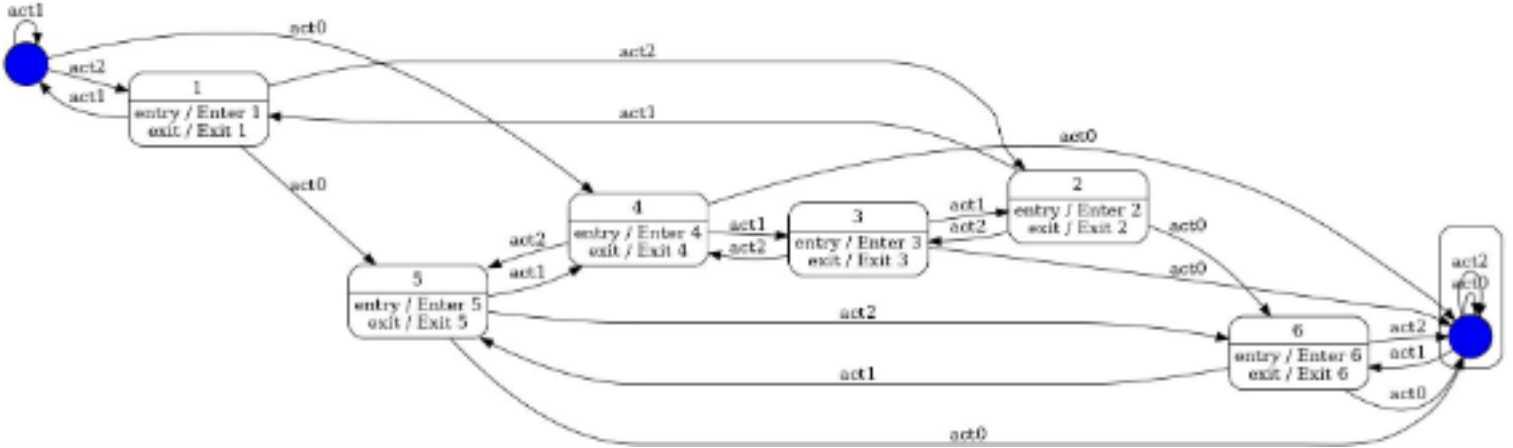
Quale delle seguenti frasi è corretta riguardo al Sequence Diagram in figura ?



- 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. 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.

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

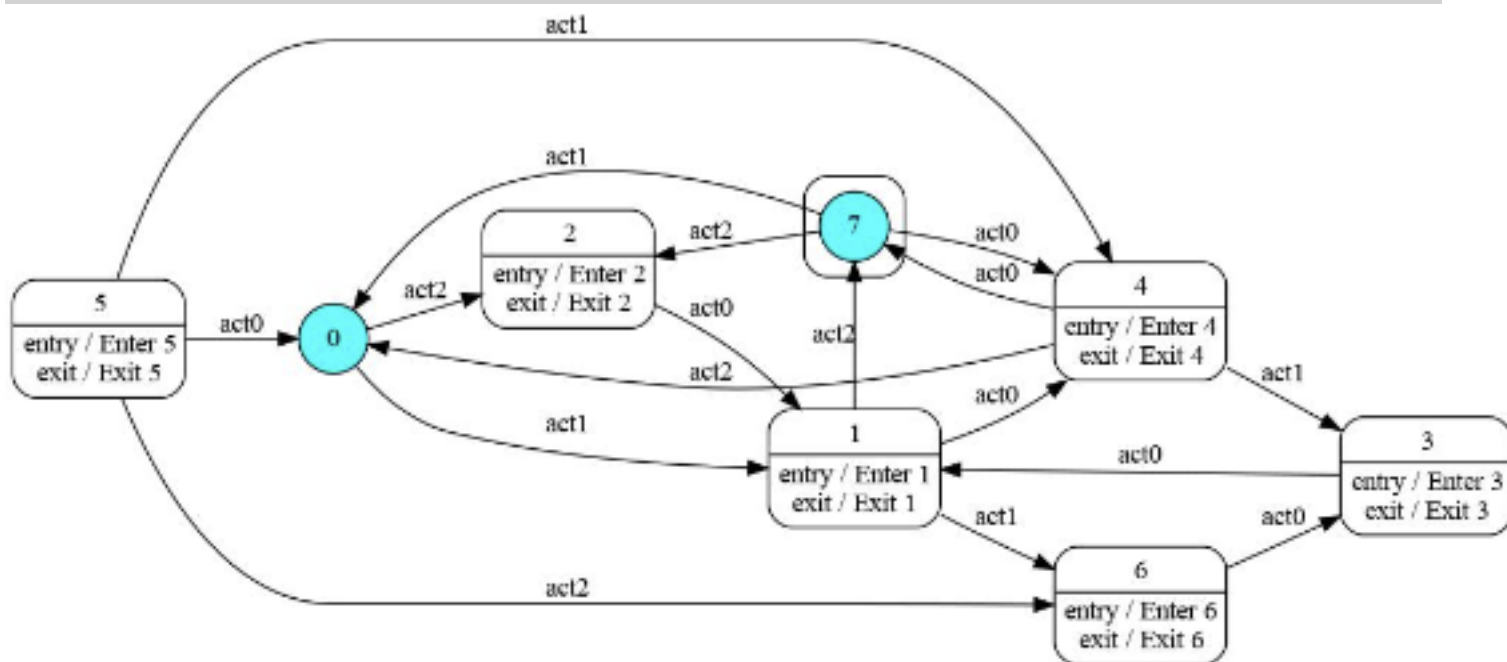
La transition coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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, 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



1.60%

2.40%

3.80%

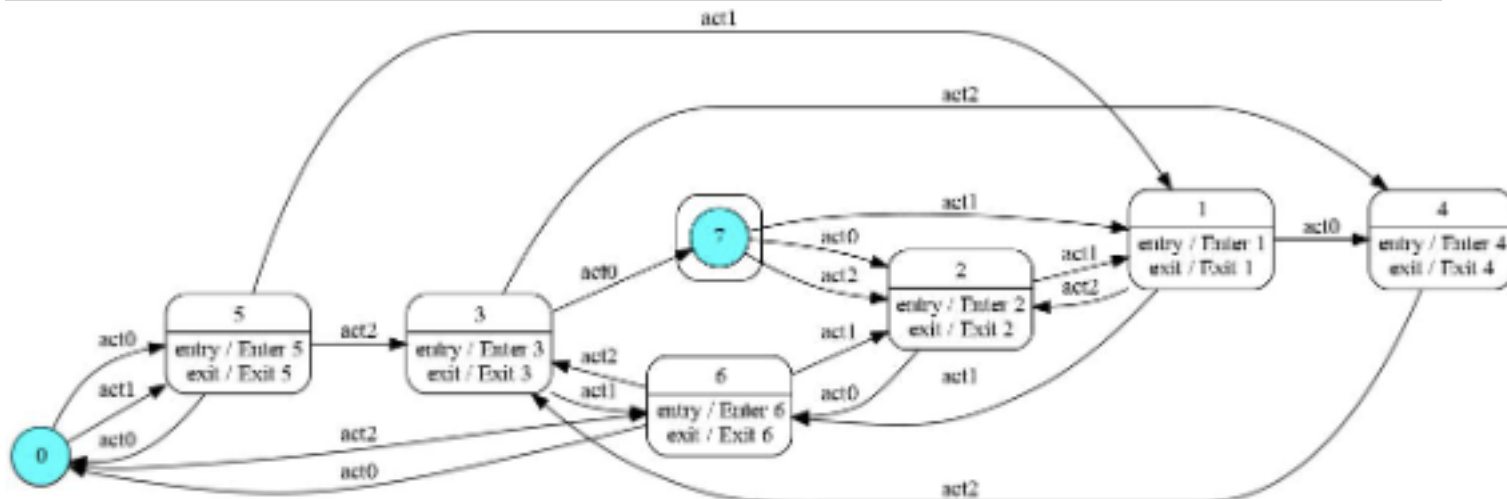
Risposta : 1

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

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

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

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



1.80%

2.40%

3.60%

Risposta : 3

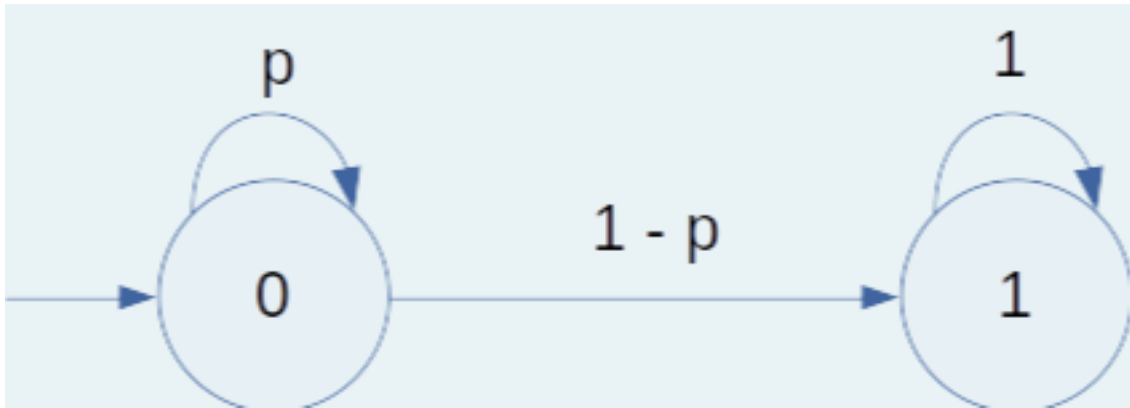
La transition coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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

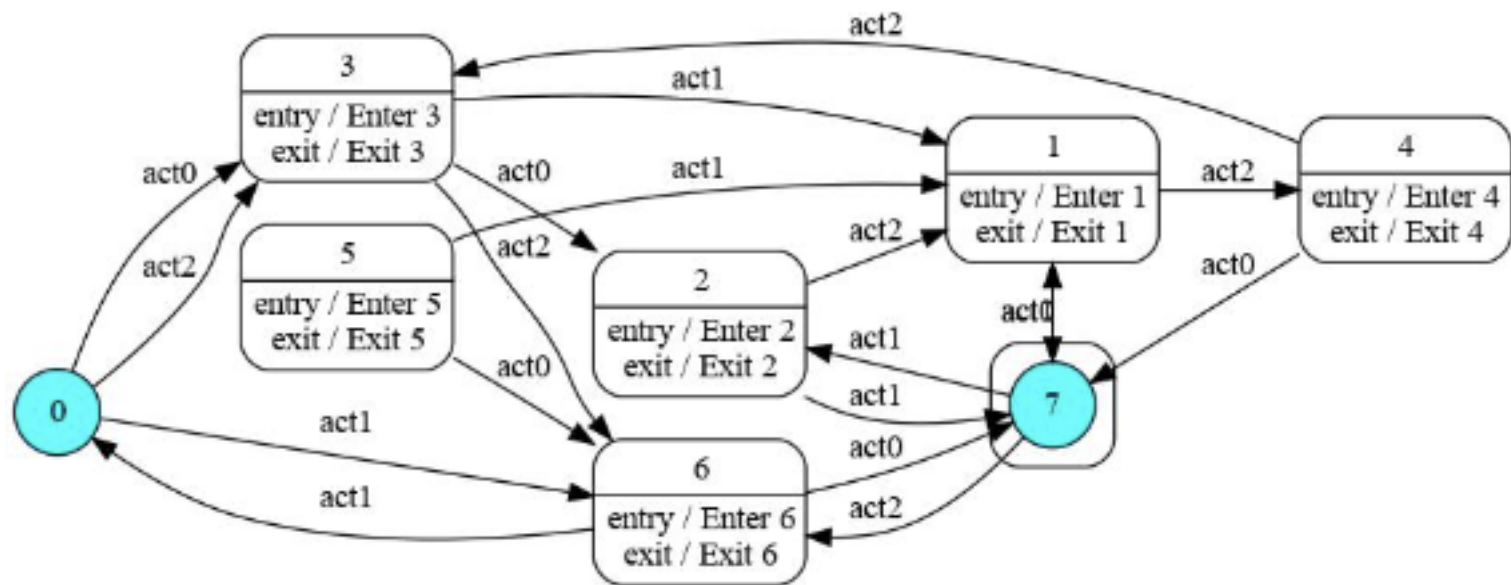
3) Start PIN validation, card inserted, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, PIN Entered, Invalid PIN, More than 3 failed..., END PIN validation 1; Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra



- 1.90%
- 2.100%
- 3.80%

Risposta : 1

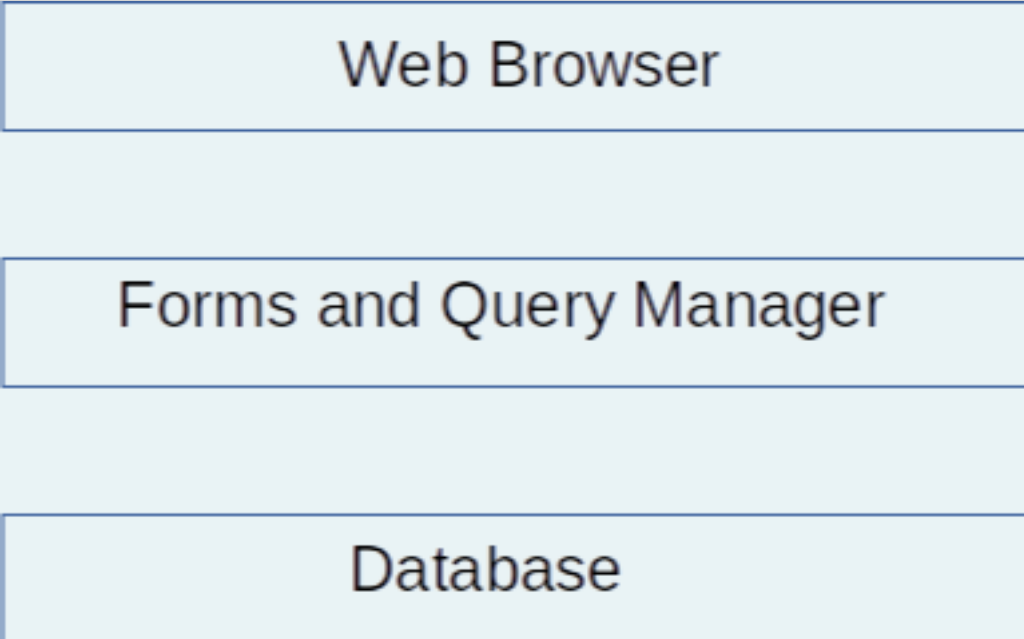
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?



- 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?




```

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

```

```

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

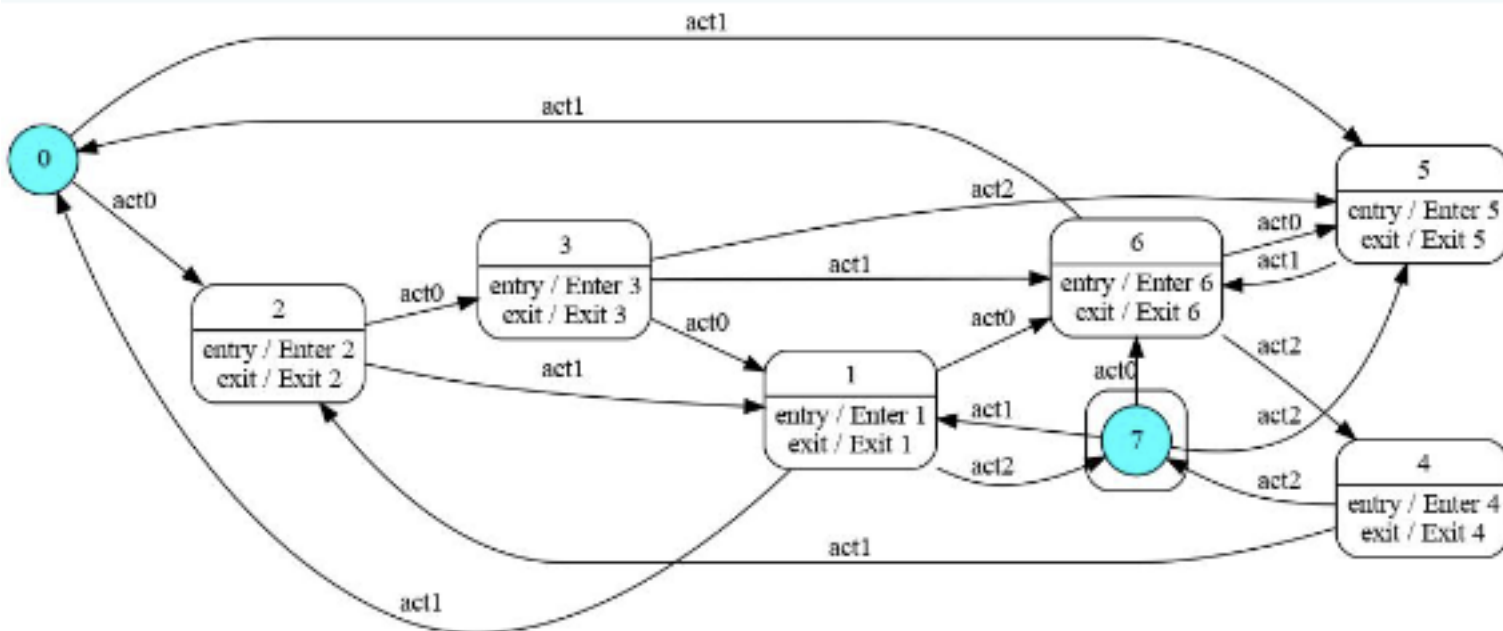
```

Risposta : 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) rggunti 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



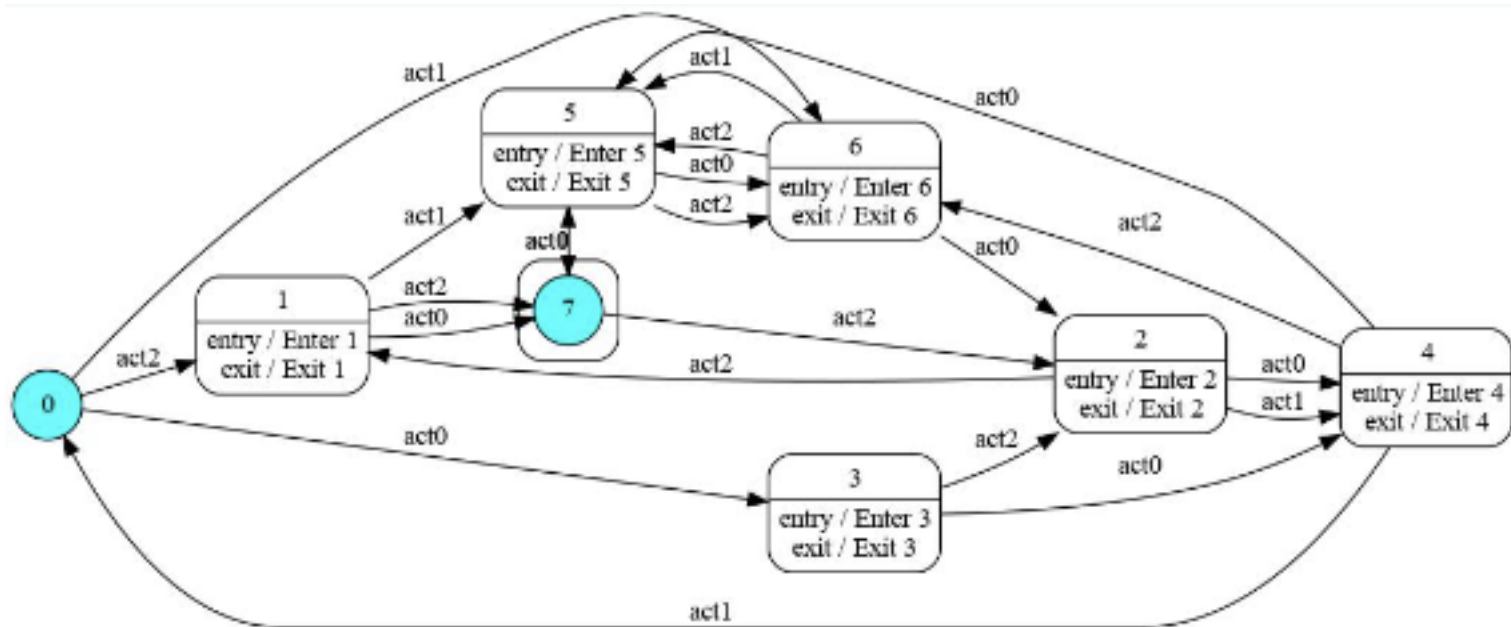
1.60%

2.80%

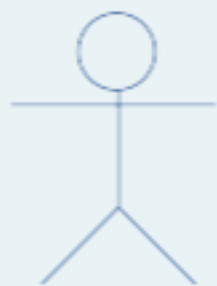
3.90%

Risposta : 3

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 ?



Customer



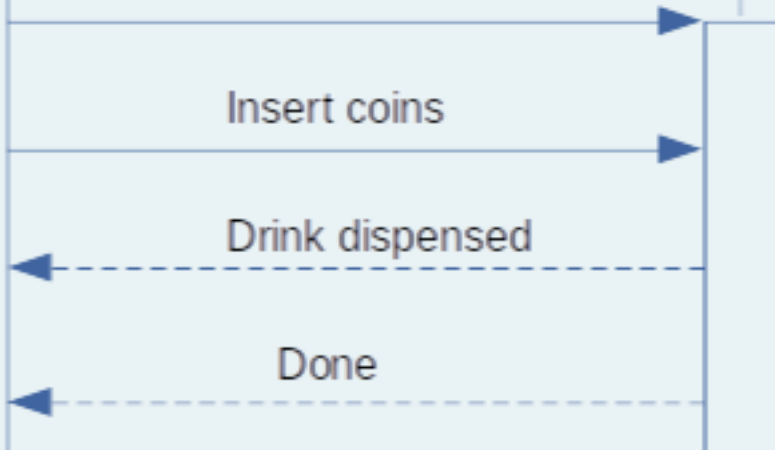
Vending Machine

Select drink

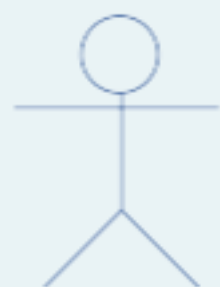
Insert coins

Drink dispensed

Done



Customer



Vending Machine

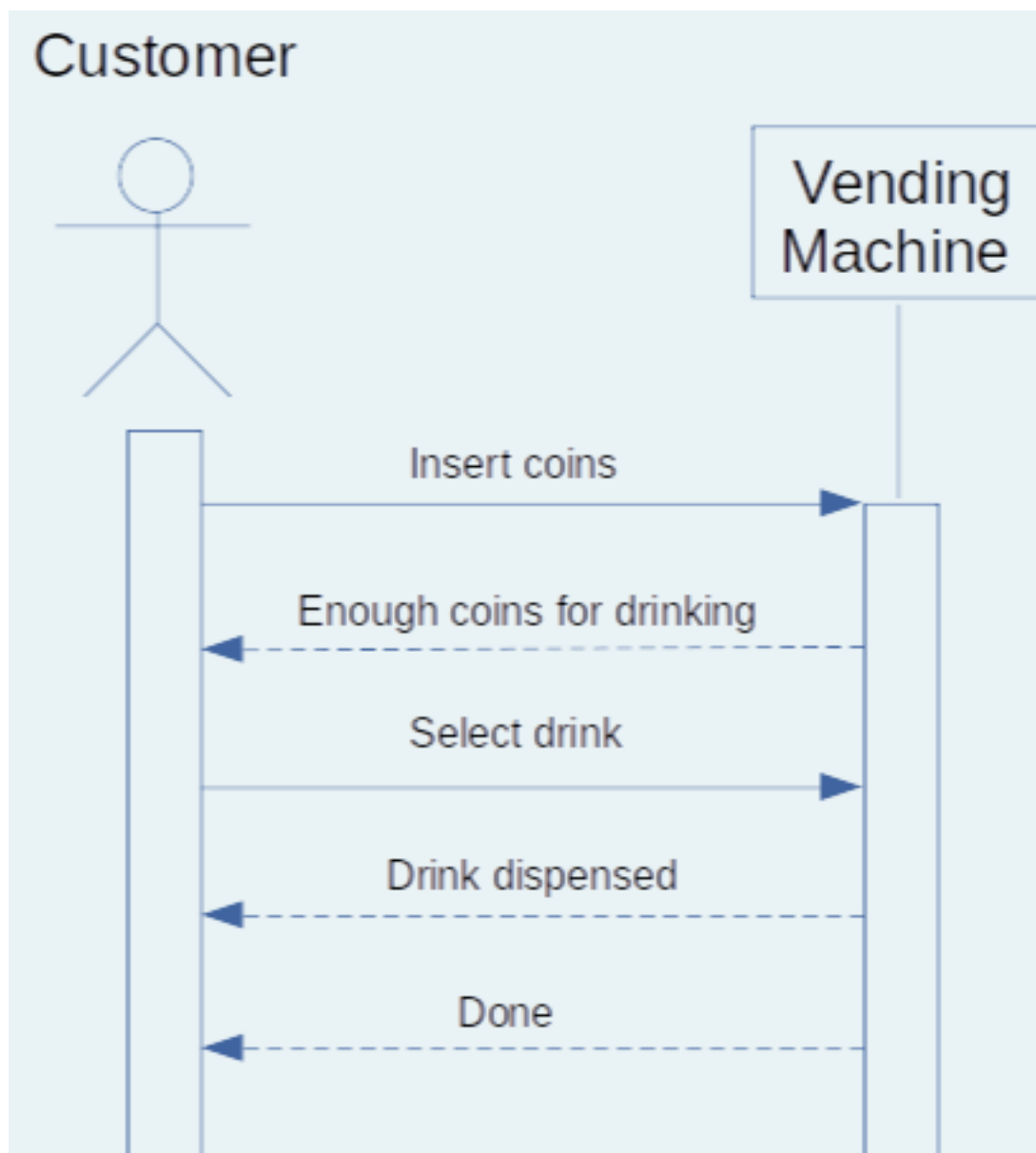
Insert coins

Enough coins for drinking

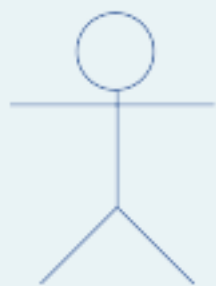
Select drink

Drink dispensed

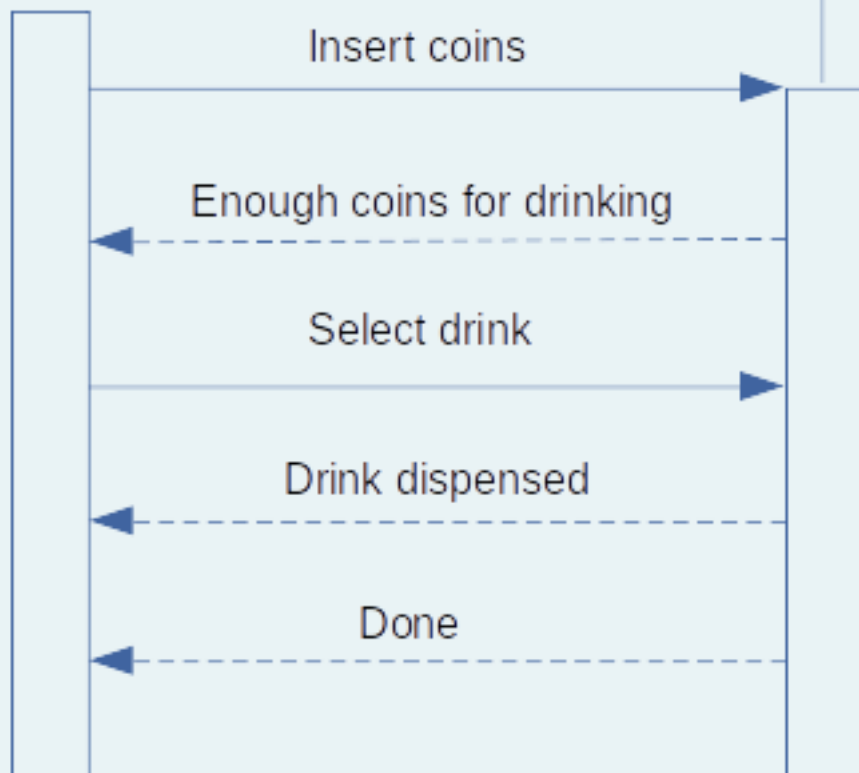
Done



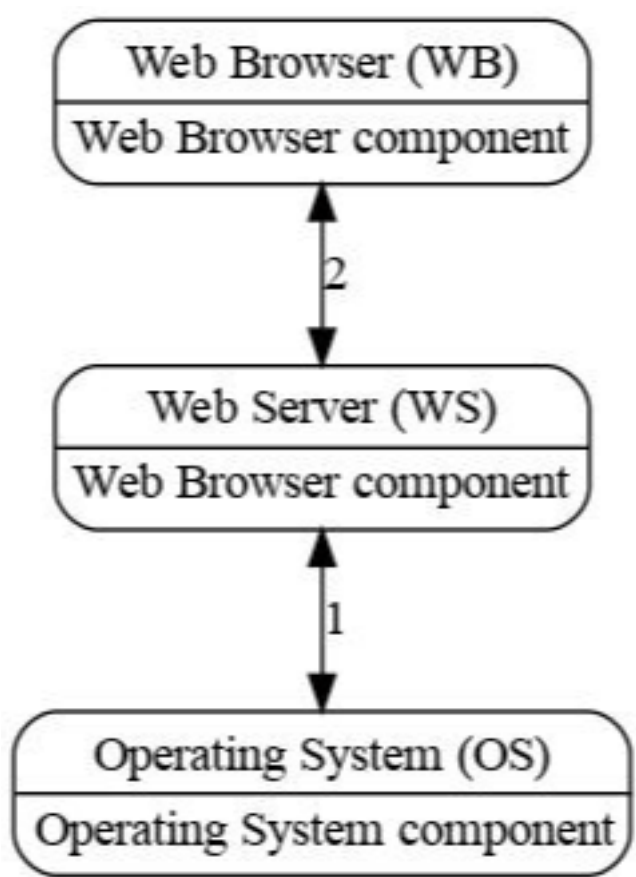
Customer



Vending Machine



Quale pattern architetturale meglio descrive l'architettura in figura ?



1.Layred architecture.

2.Model View Controller.

3.Pipe and filter architecture.

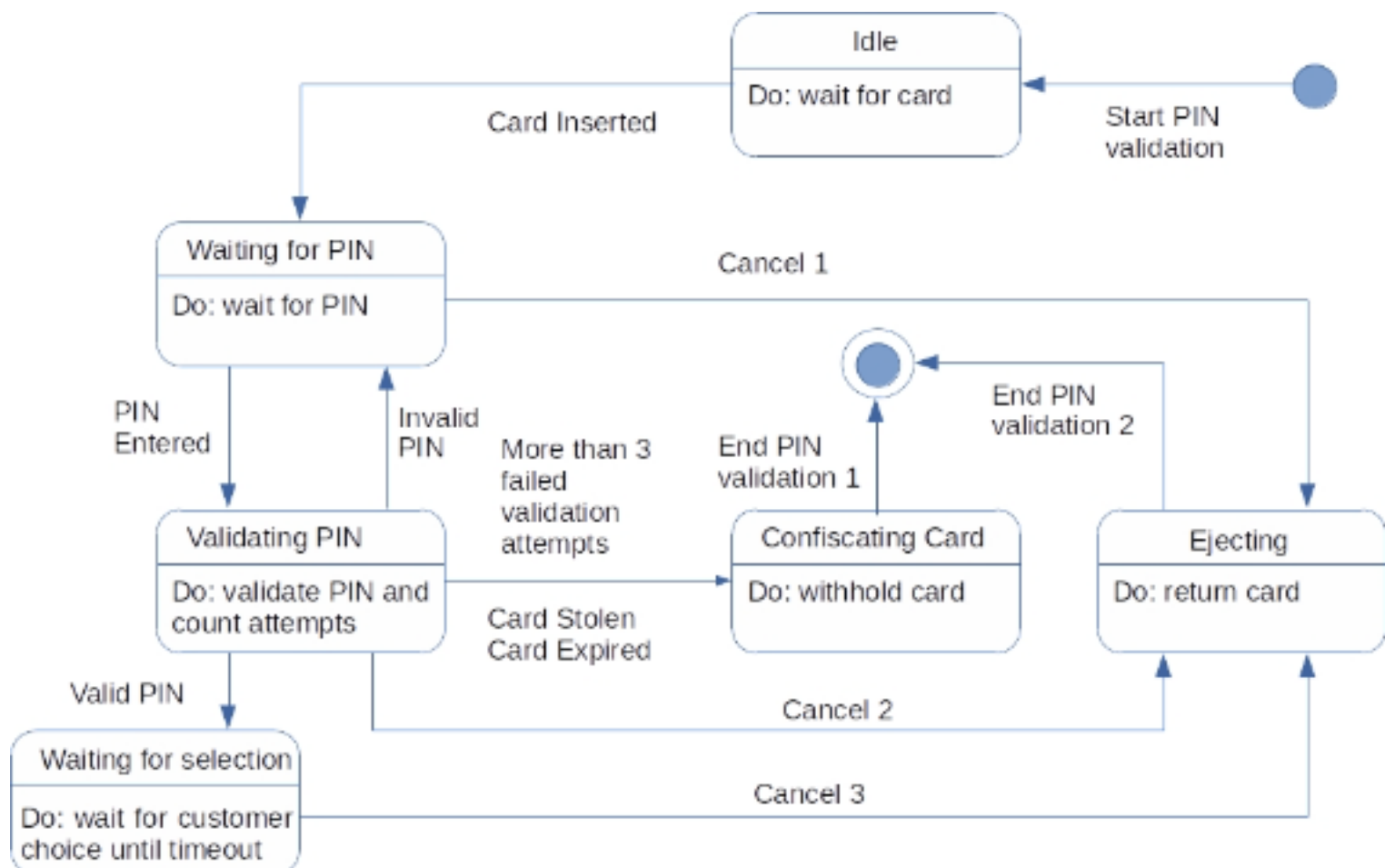
Risposta : 1

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, 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



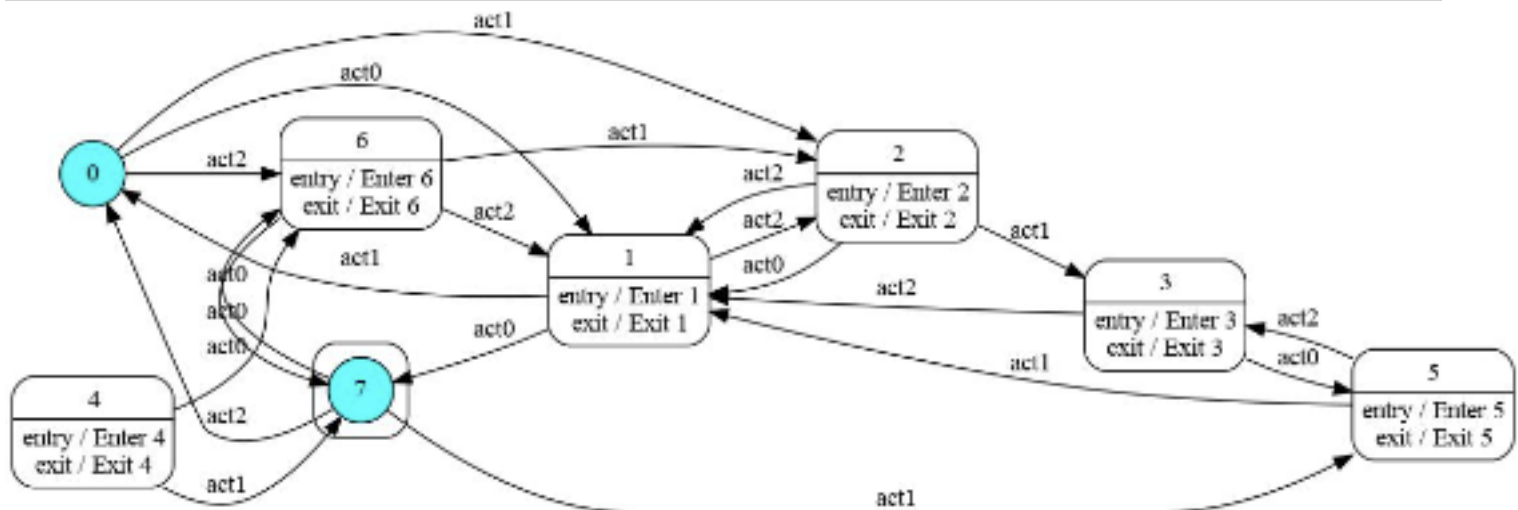
1.90%

2.80%

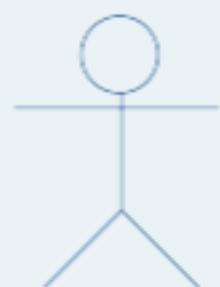
3.100%

Risposta : 3

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 ?



Customer



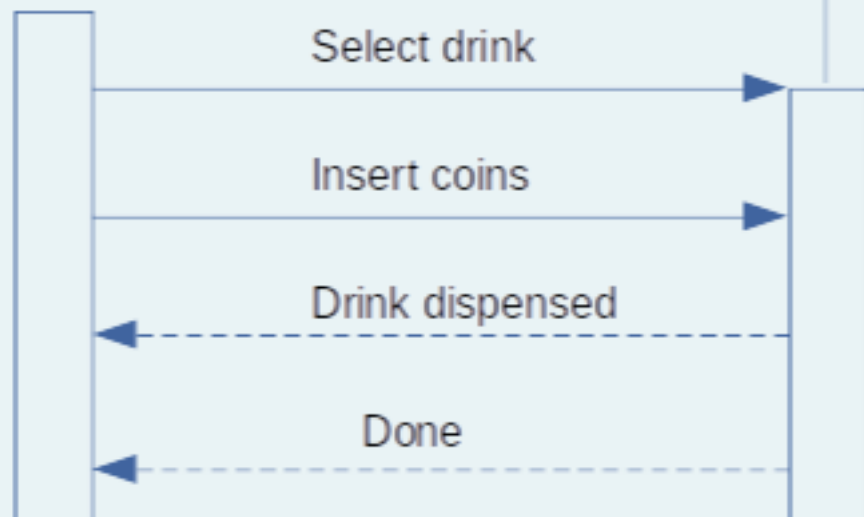
Vending Machine

Select drink

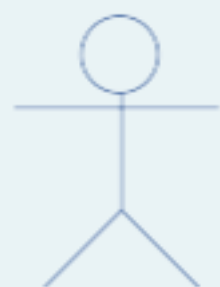
Insert coins

Drink dispensed

Done



Customer



Vending
Machine

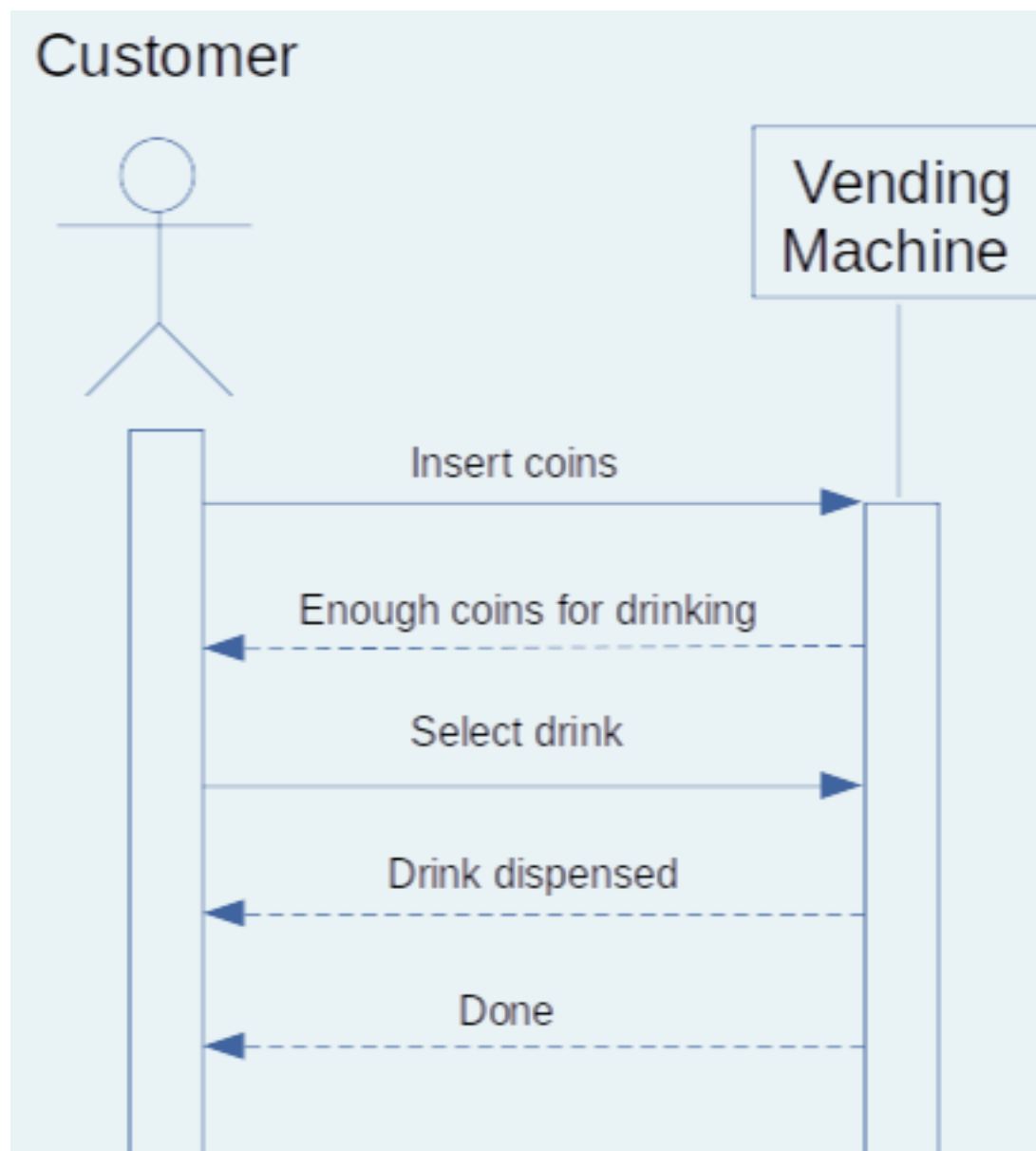
Insert coins

Enough coins for drinking

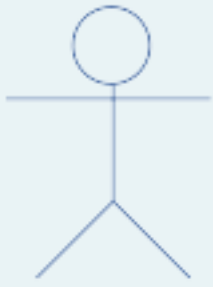
Select drink

Drink dispensed

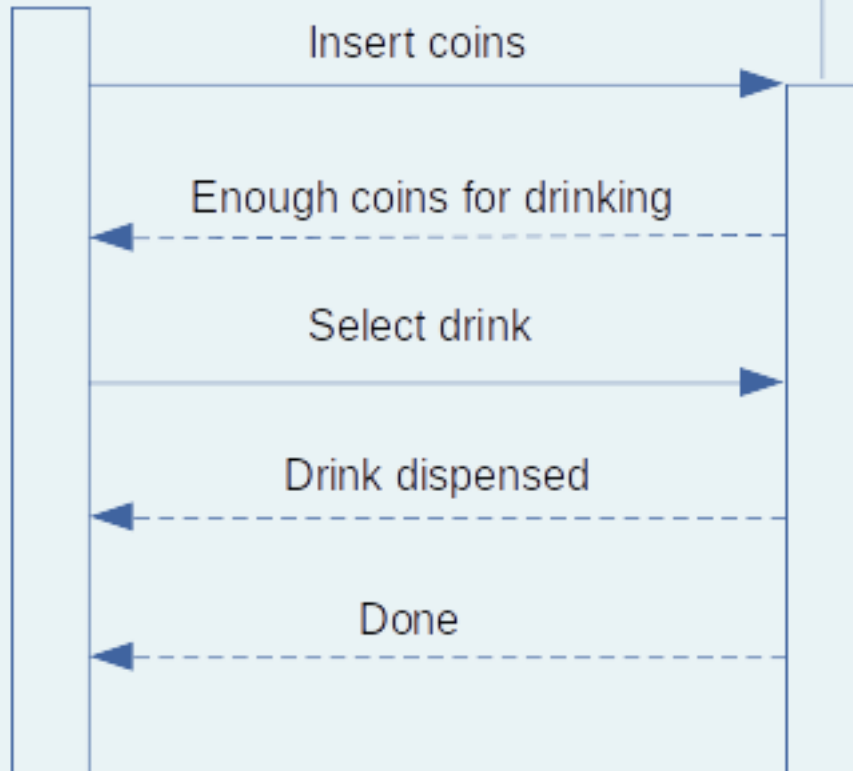
Done



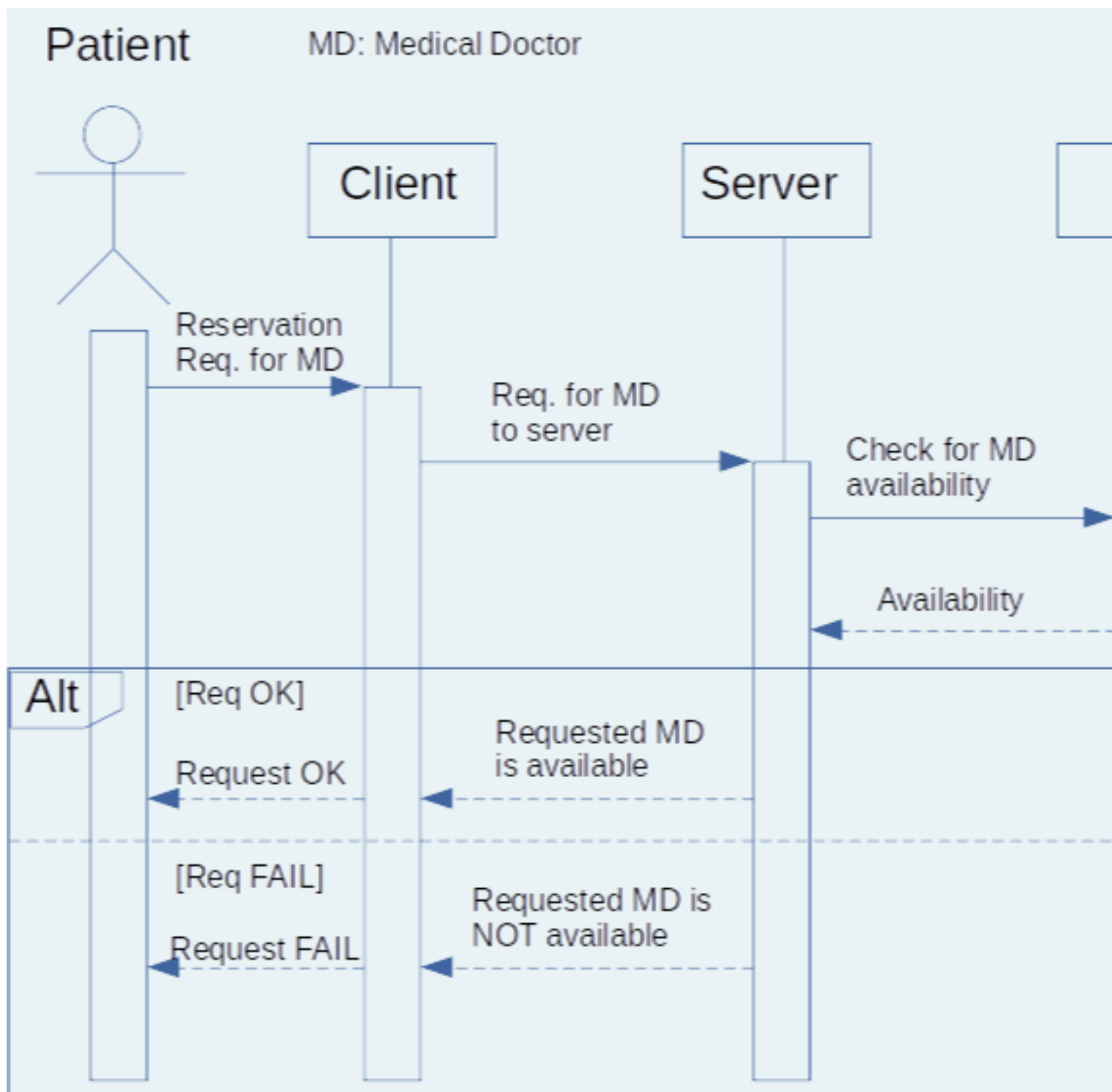
Customer



Vending Machine



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 ?



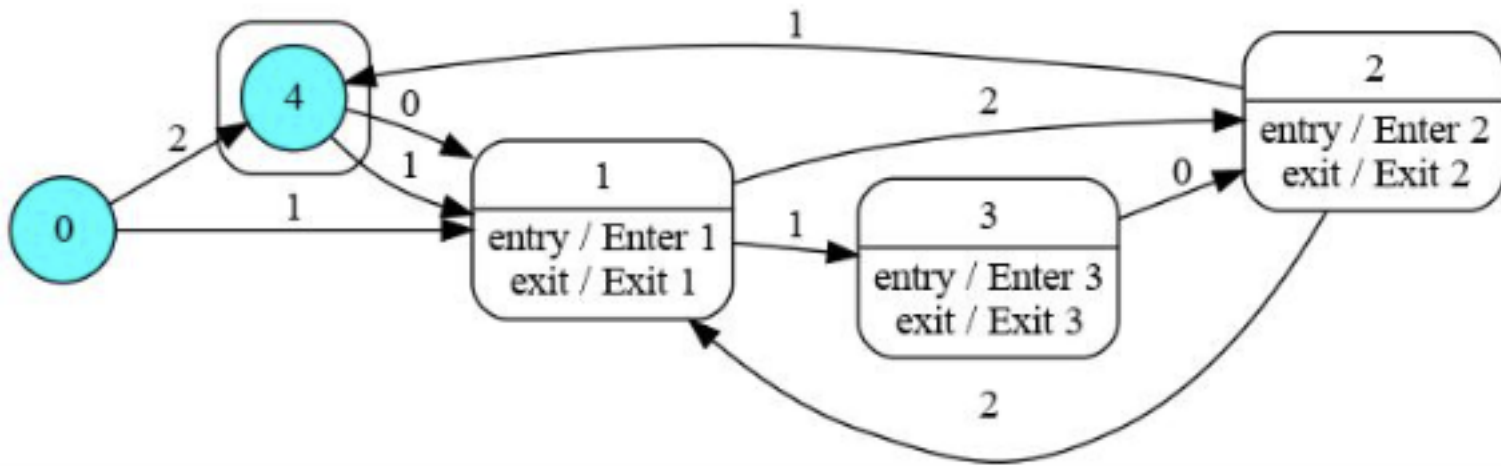
1. Anche se ci sono abbastanza monete potrebbe non essere possibile ottenere la bevanda selezionata.

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

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

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

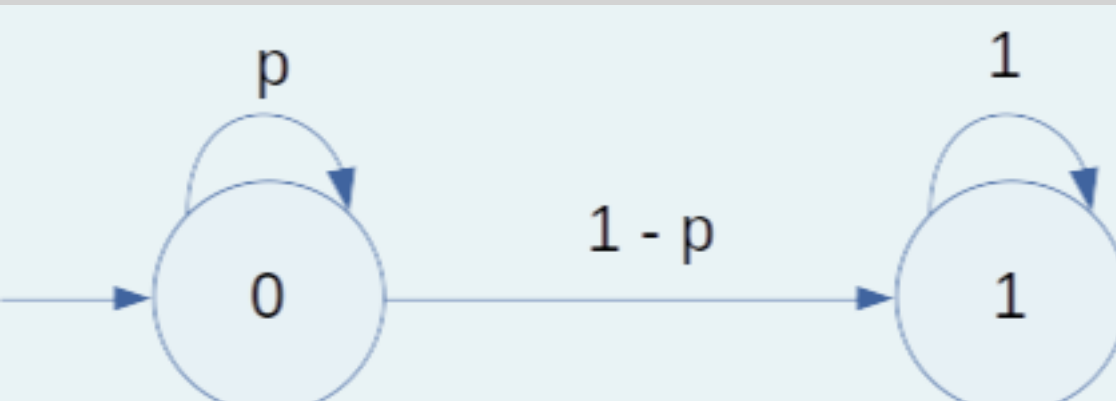
3.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0:
waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer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

Risposta : 2

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



1.60%

2.40%

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.

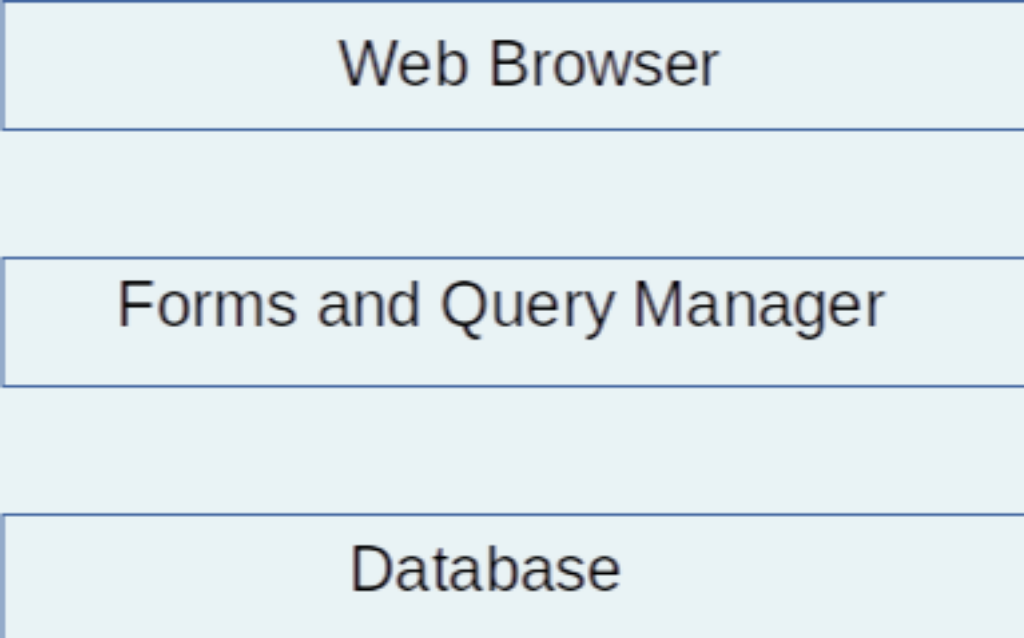
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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



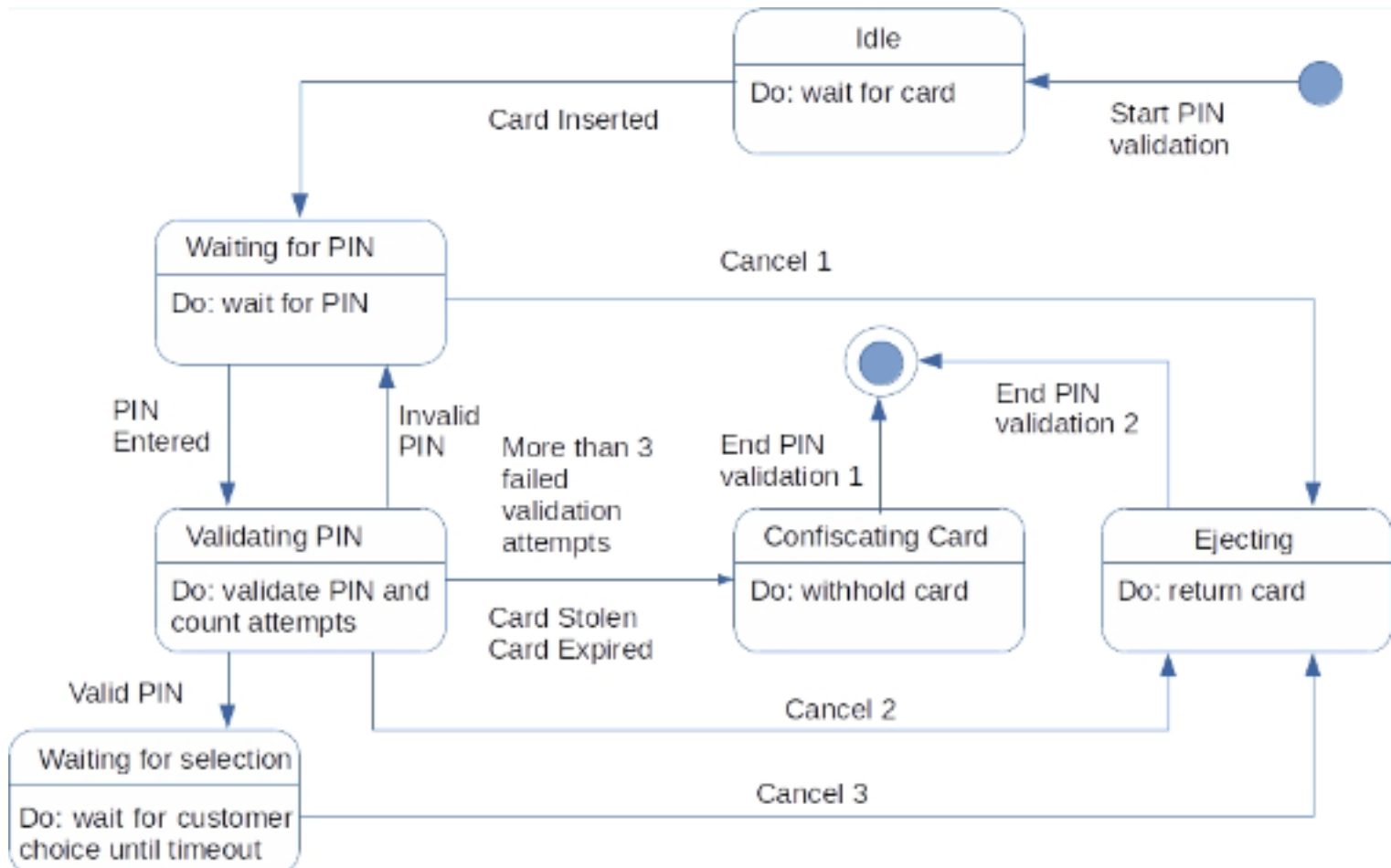
1.60%

2.90%

3.80%

Risposta : 2

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 : 2

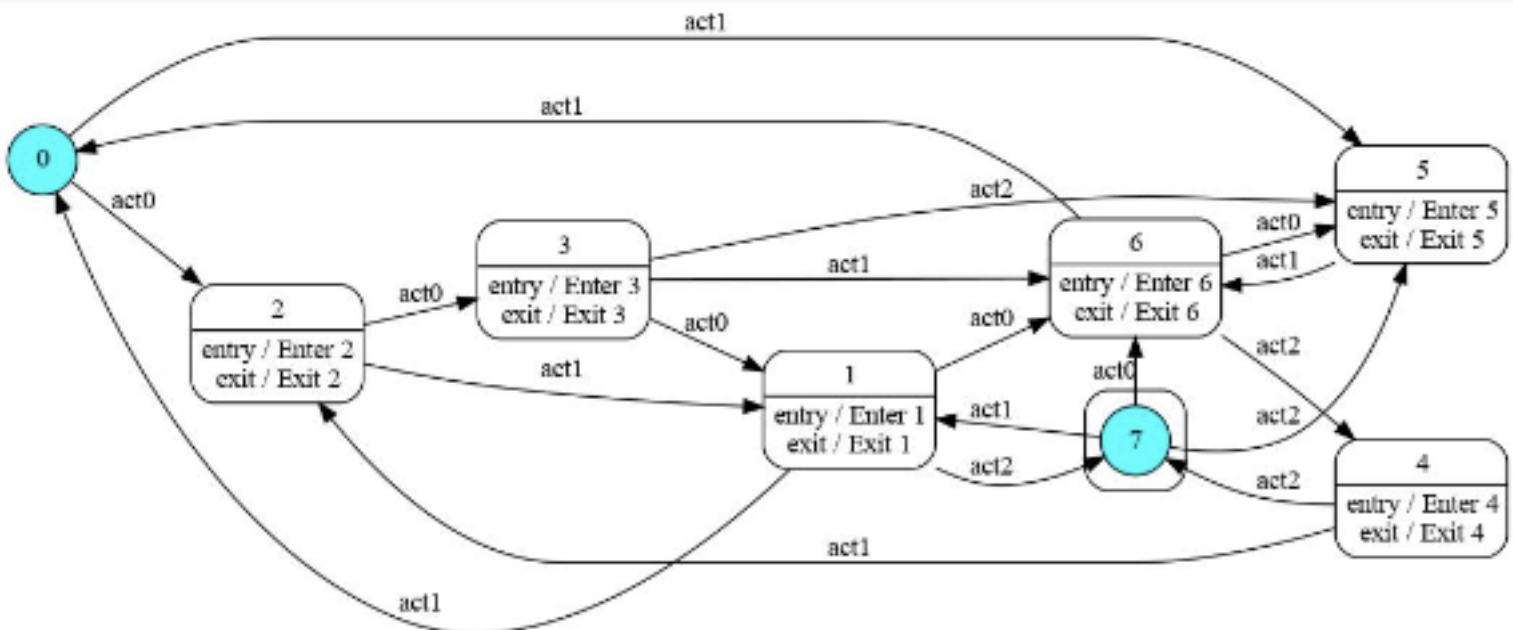
La transition coverage di un insieme di test cases (cioè sequeze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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



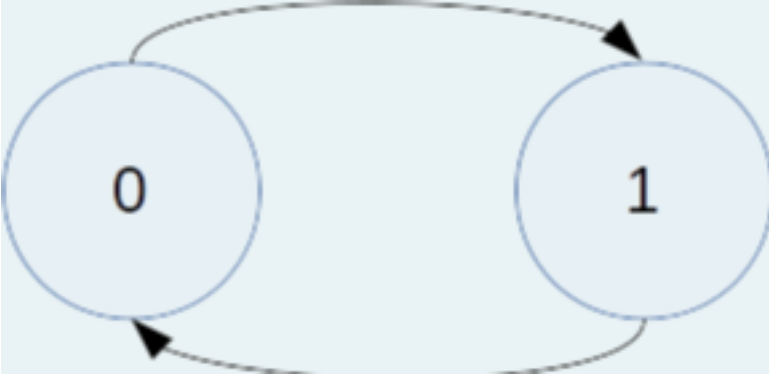
1.100%

2.80%
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:

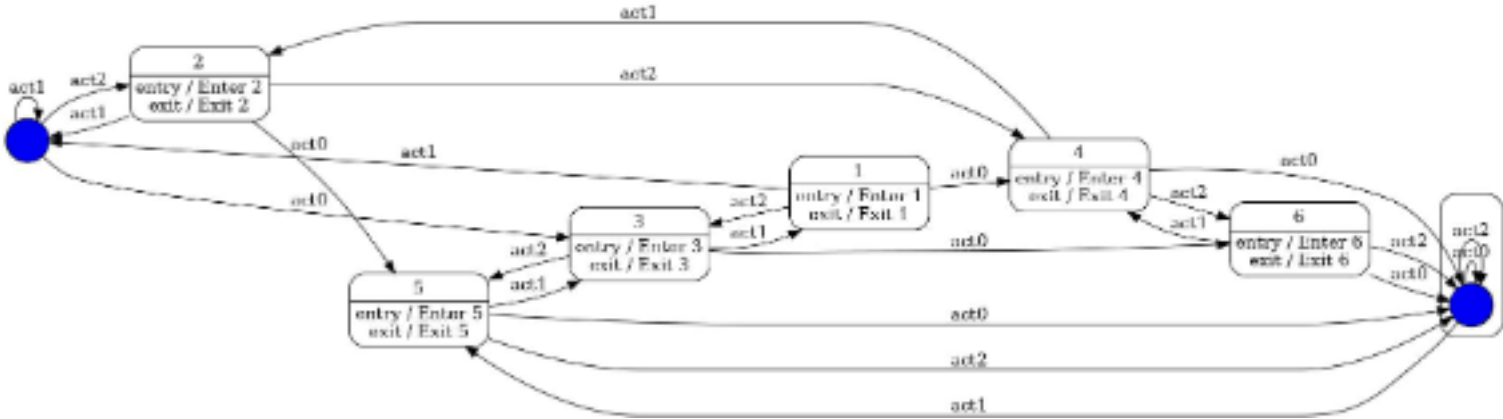
- 1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2
 - 2) Start PIN validation, card inserted, PIN Entered, Cancel 2, End PIN Validation 2
- Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra



1.60%
2.80%
3.40%

Risposta : 1

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

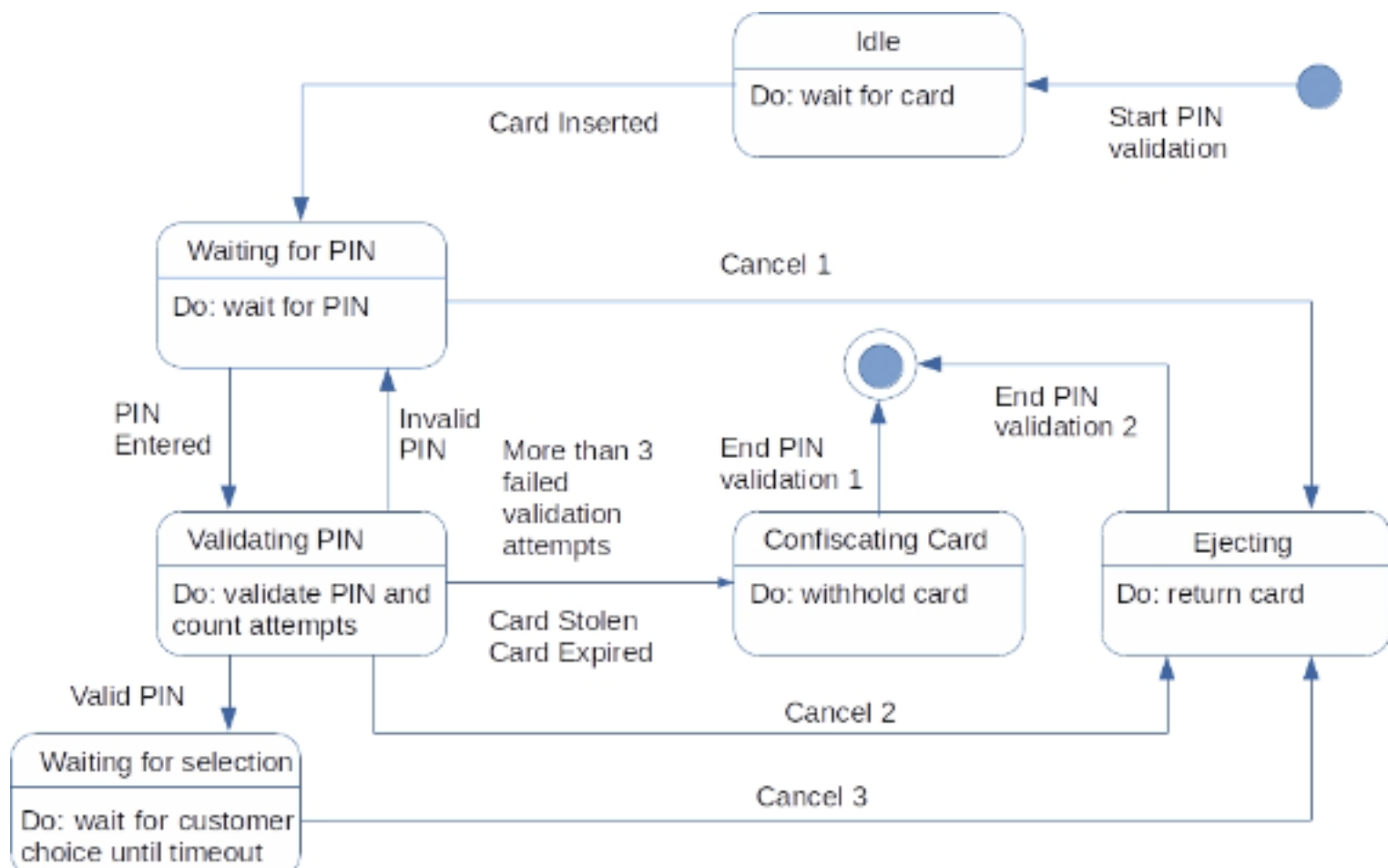


- 1. Una volta selezionato il piatto di mare da preparare, la preparazione del pesce e del contorno procedono in parallelo.
- 2. Una volta selezionato il piatto di mare da preparare, la stessa persona prepara prima il pesce e poi il contorno.
- 3. Una volta selezionato il piatto di mare da preparare, 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) raggiunti almeno una volta.
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

- 1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2
 - 2) Start PIN validation, card inserted, PIN Entered, Cancel 2, End PIN Validation 2
- Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



1.60%

2.80%

3.90%

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 */

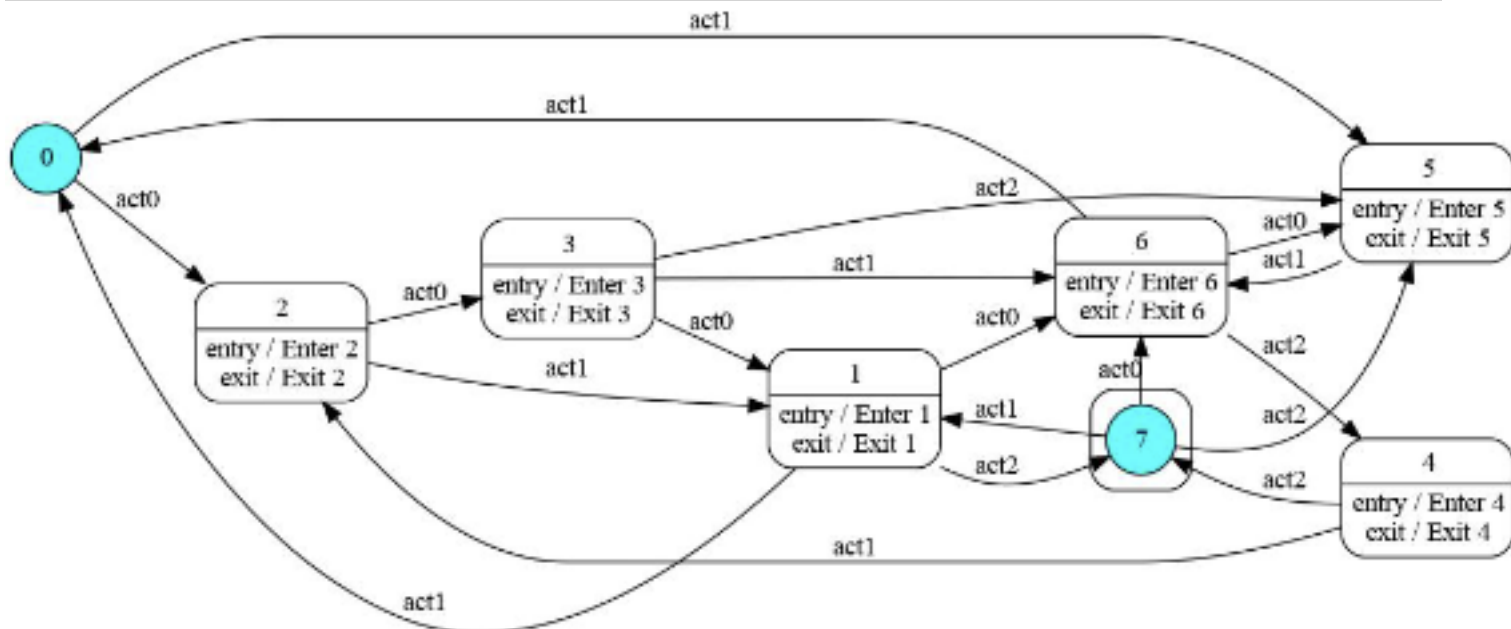
int f(int x) { int y = 0;
    LOOP: if (abs(x) - y <= 2)
        {return ;}
    else {y = y + 1; goto LOOP;}
} /* f() */

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

```

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

Quale delle seguenti è la branch coverage conseguita?



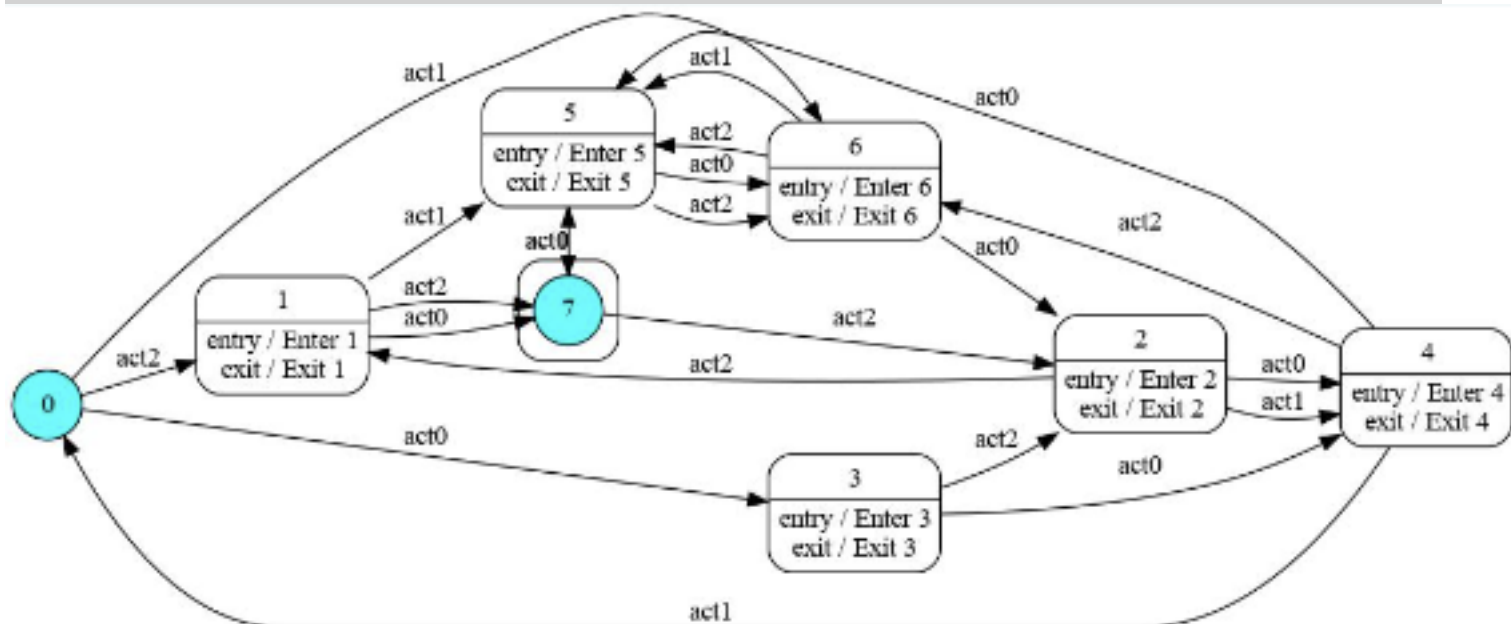
1.100%

2.80%

3.50%

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. 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.

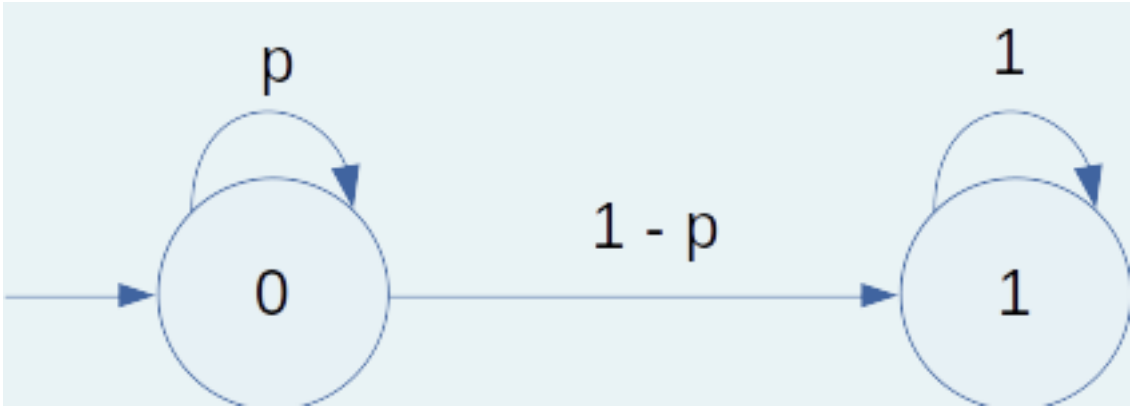
Risposta : 2

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

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

Uno scenario è una sequenza di stati.

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



- 1.0.32
- 2.0.12
- 3.0.08

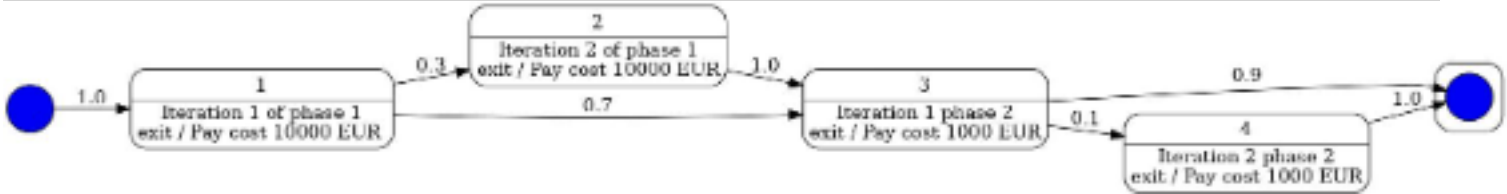
Risposta : 2

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.

Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:Test case 1: act0 act0 act2 act0 act0 act0 act2 act1 act2 act0 act2 act2 act2 act2 act0 act0 act1 act2 act2 act0 act2 act0 act2 act1 act0 act2 act1 act2 act2 act0 act2

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

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



- 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.

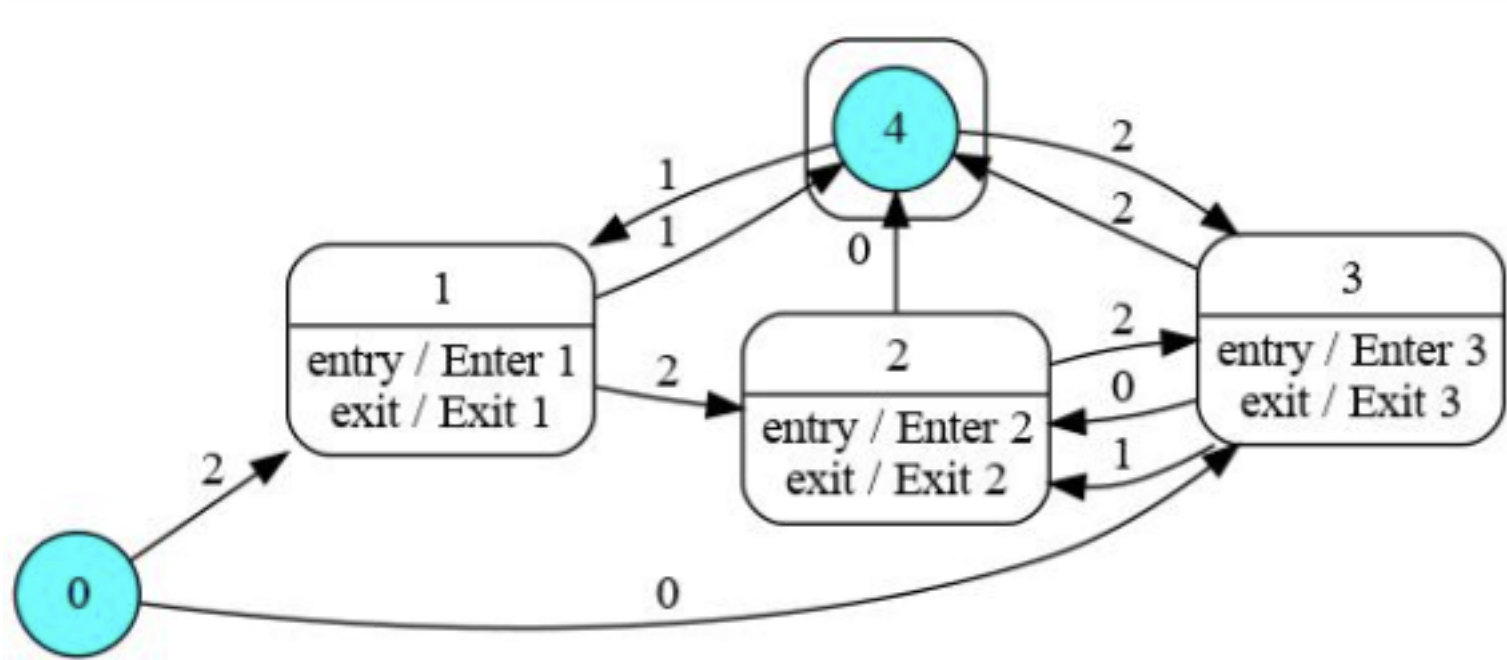
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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%

- 2.State coverage: 100%
- 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.

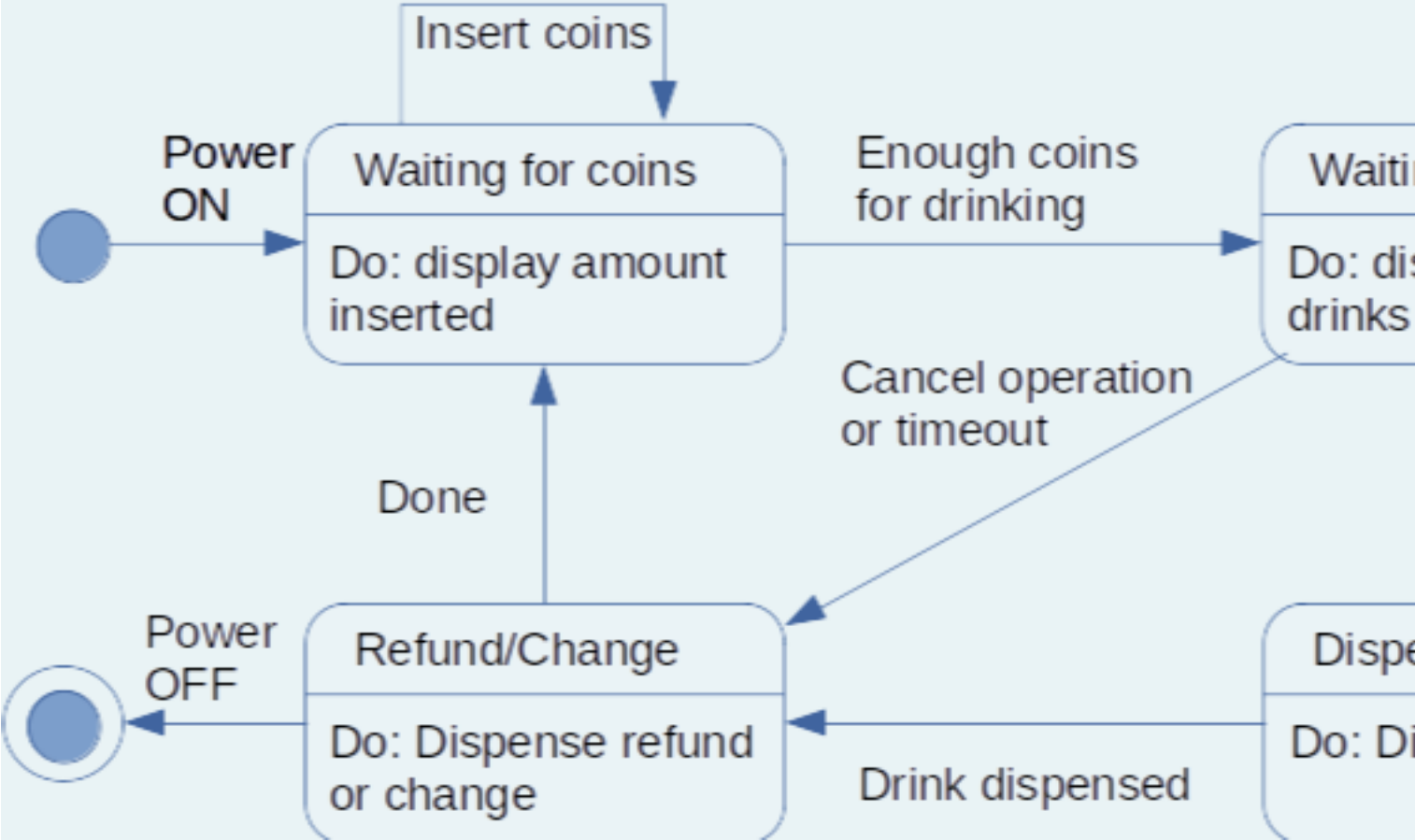
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

Test case 1: act2 act1 act2 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

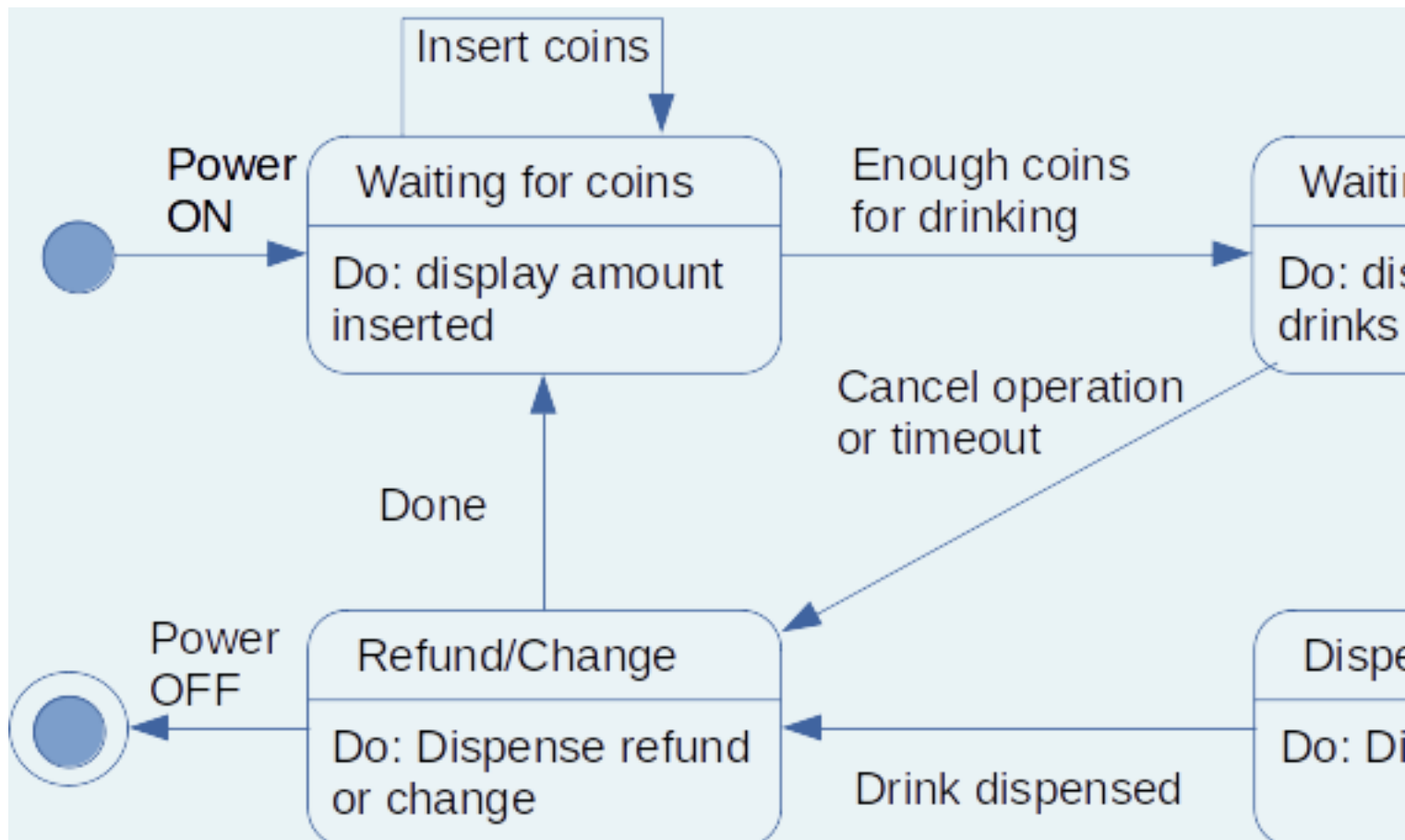
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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

Test case 1: act0 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%

2.45%

3.90%

Risposta : 2

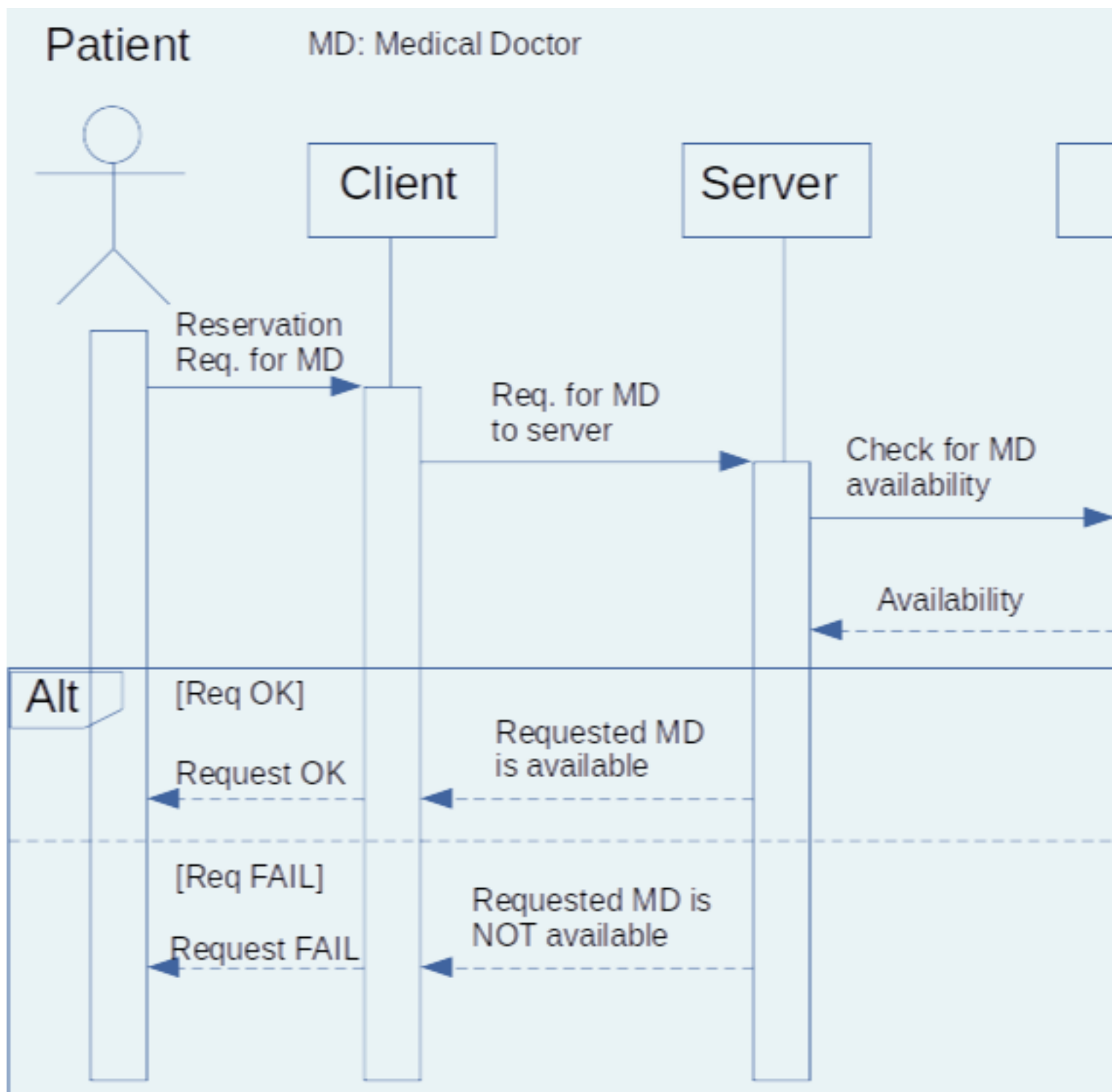
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il tempo necessario per completare la fase x è $\text{time}(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase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

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

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

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

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



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

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

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

Risposta : 1

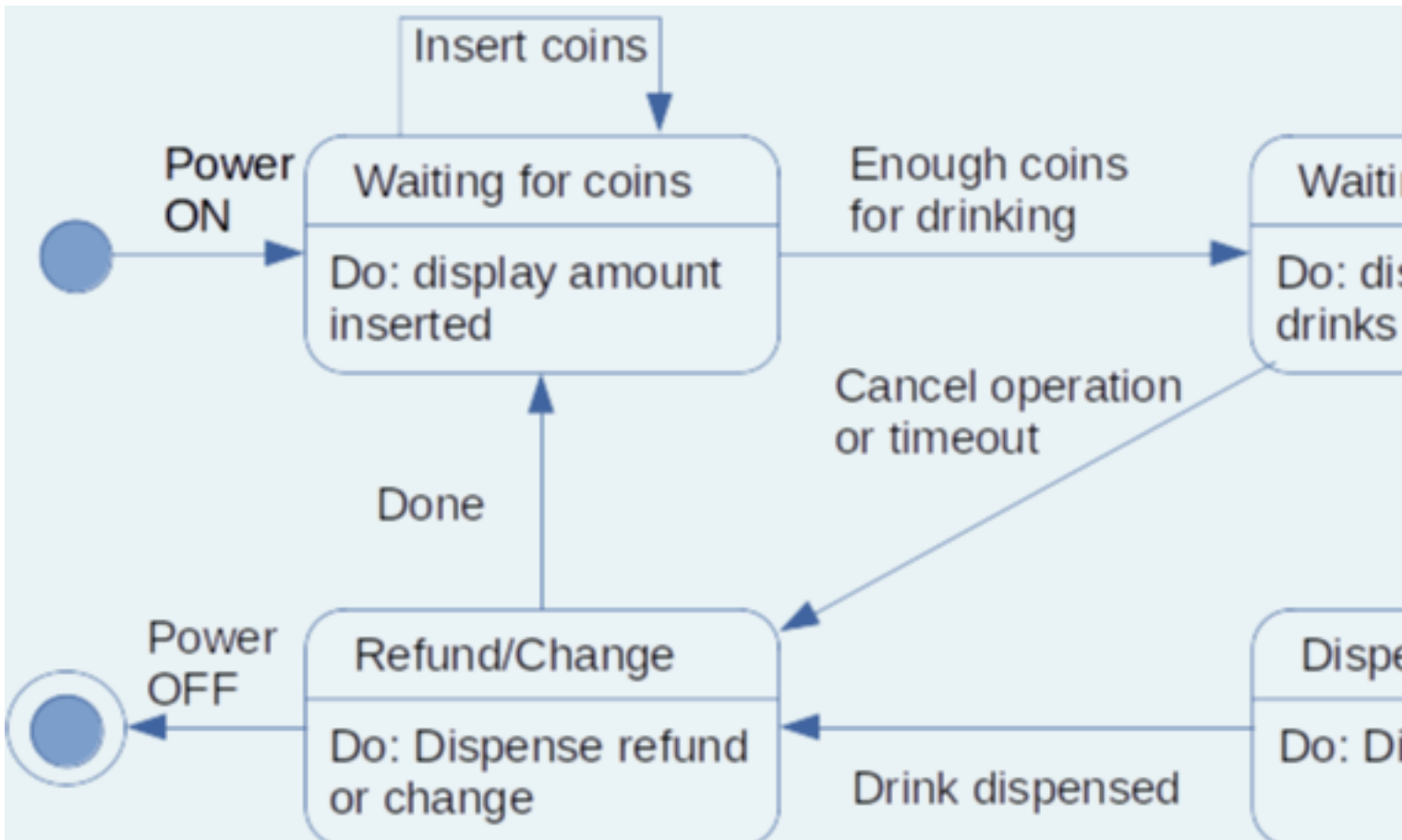
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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 act2 act1 act2 act1 act1 act1 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%

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

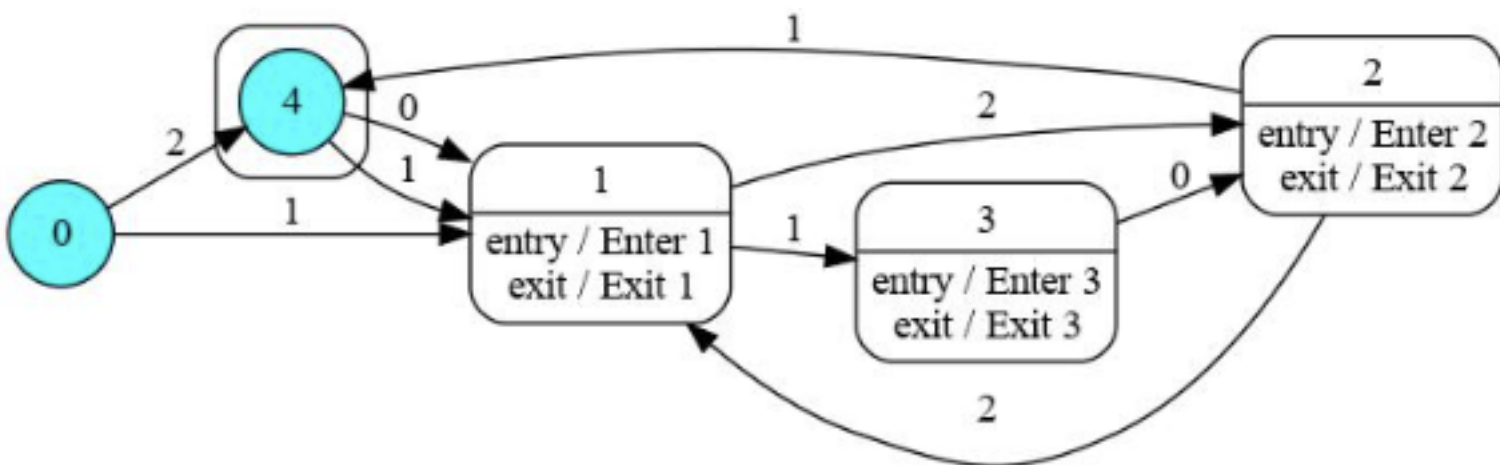
Si consideri il seguente insieme di test cases:

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

Risposta : 2

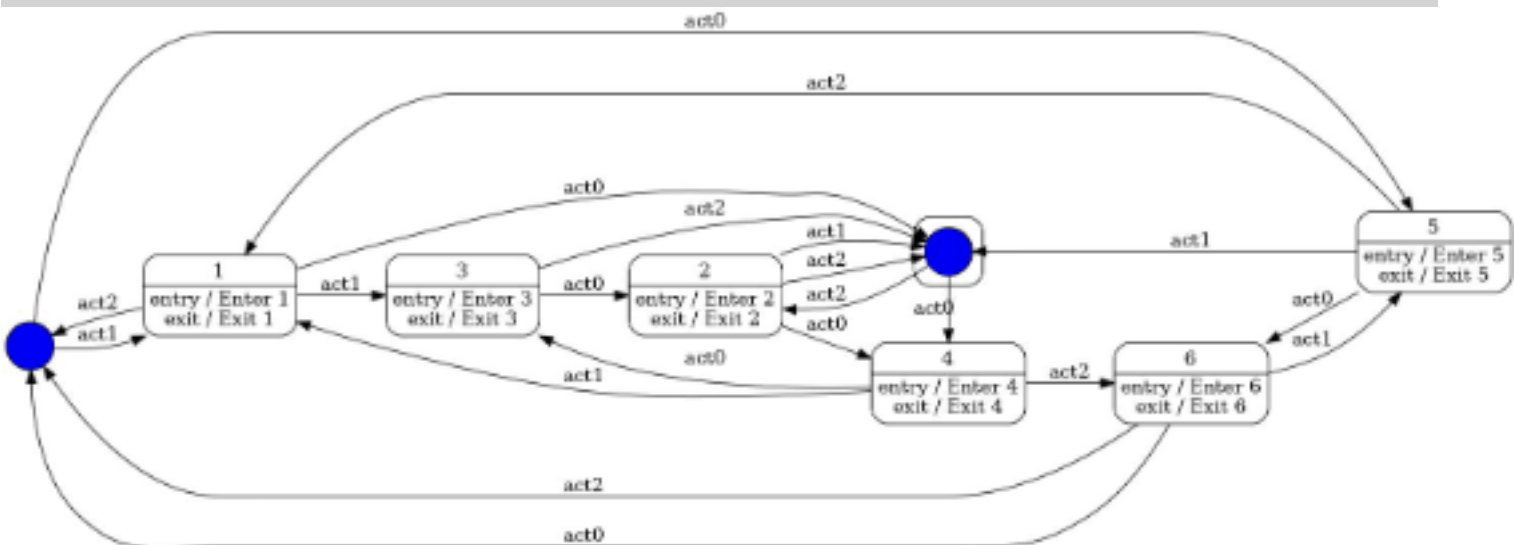
Un processo software può essere rappresentato con uno state diagram in cui gli stati rappresentano le fasi (e loro iterazioni) del procoesso software e gli archi le transizioni da una fase all'altra. Gli archi

sono etichettati con le probabilità della transizione e gli stati sono etichettati con il costo per lasciare lo stato.

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

Uno scenario è una sequenza di stati.

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



1.0.56

2.0.14

3.0.24

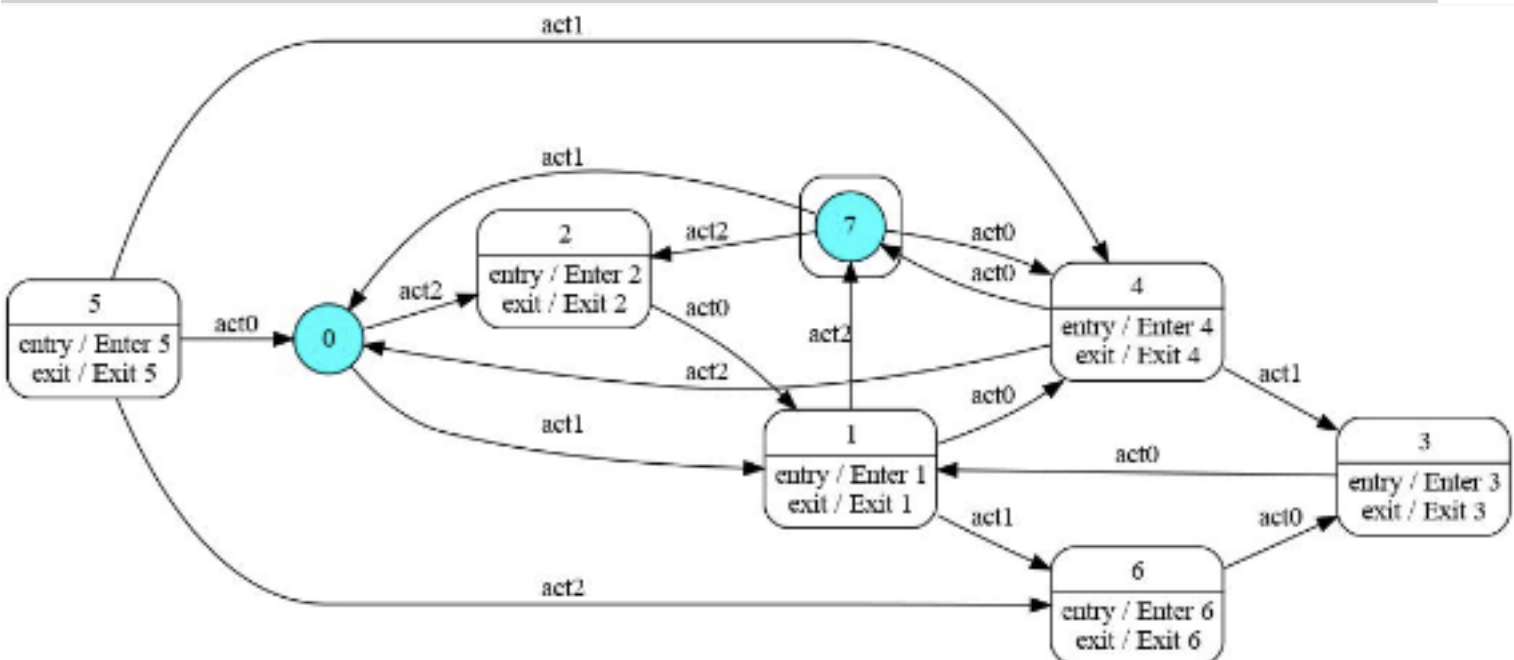
Risposta : 1

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

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

Uno scenario è una sequenza di stati.

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



1.0.28

2.0.12

3.0.42

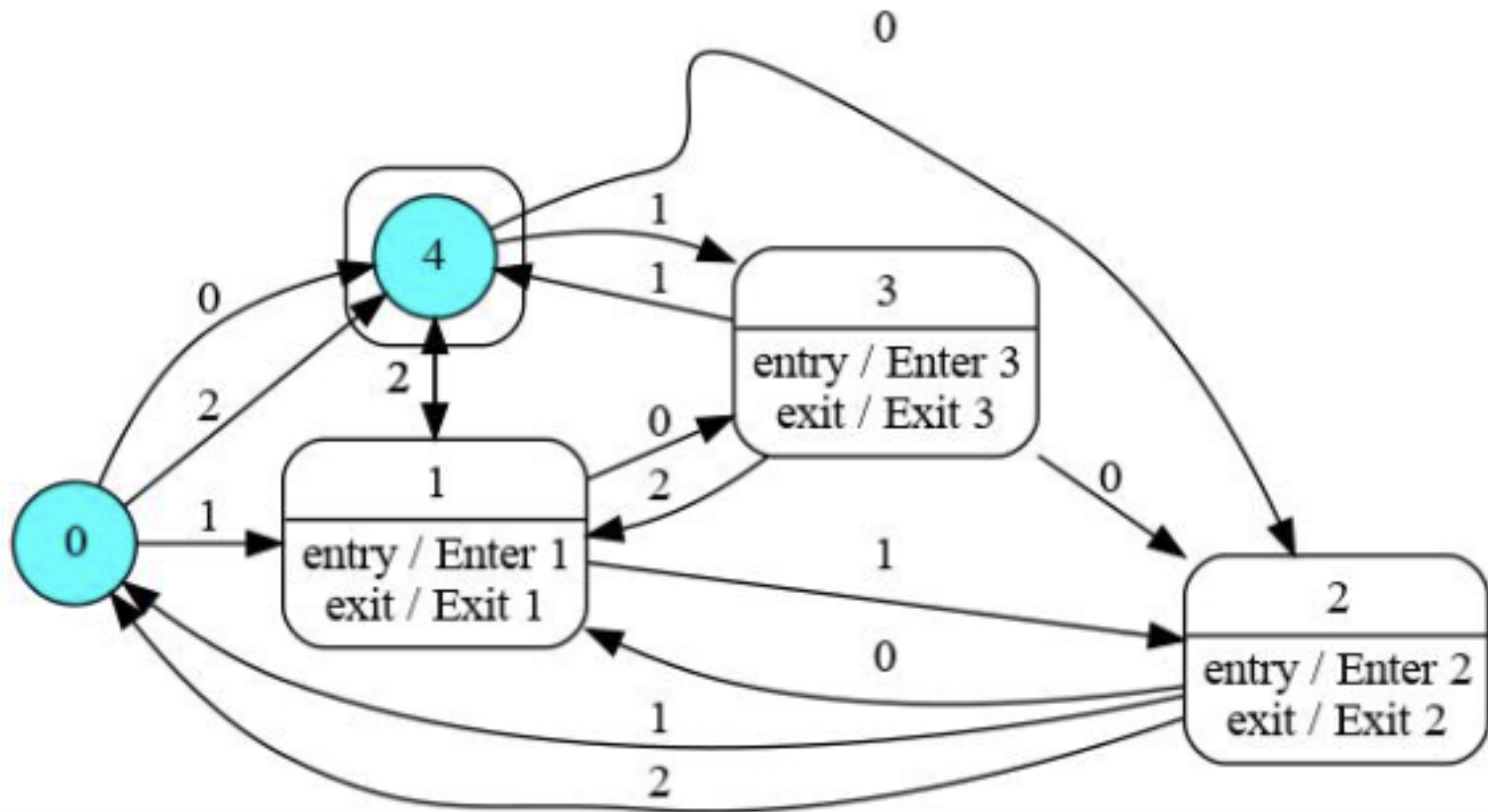
Risposta : 1

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 act1 act2 act1 act2 act2 act1

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

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



- 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.

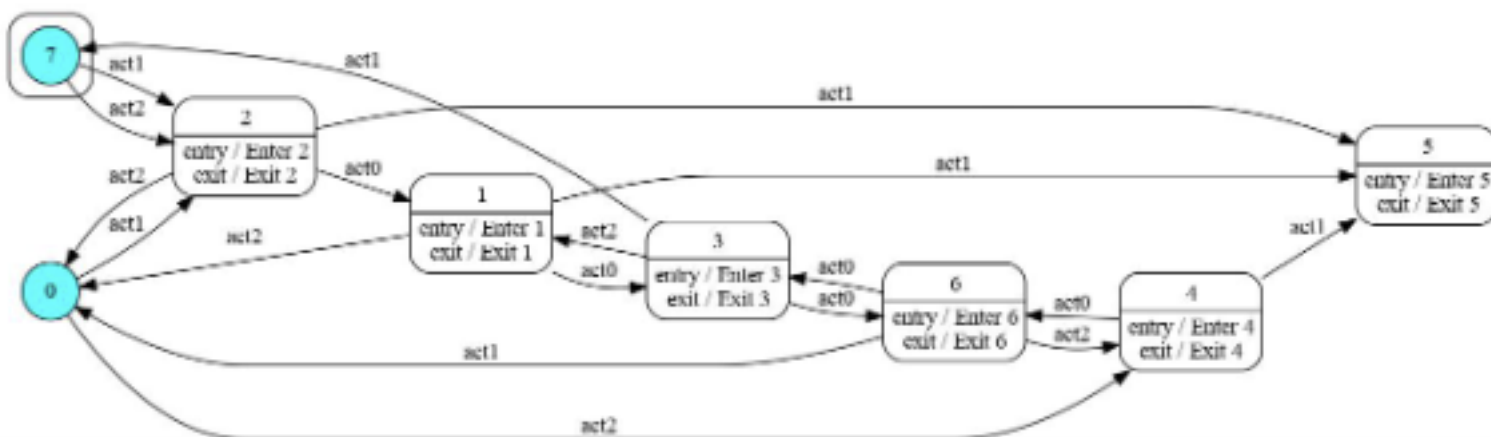
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

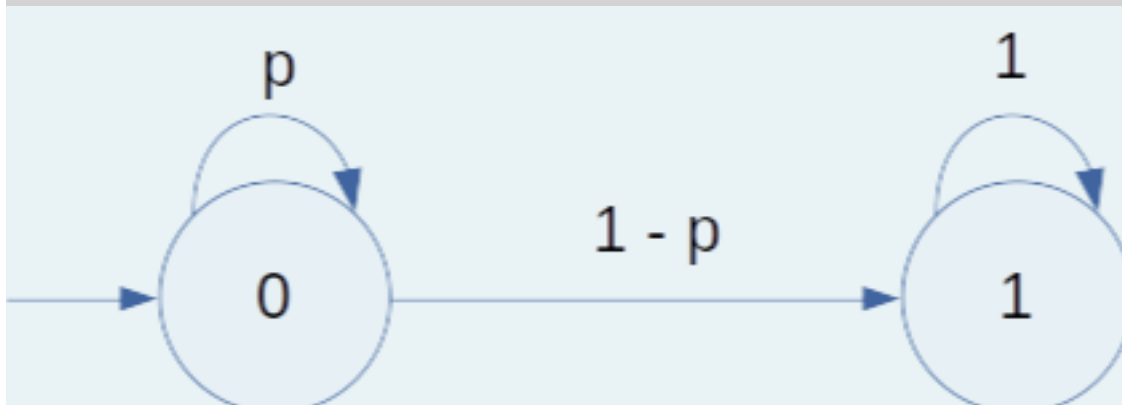
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il costo dello stato (fase) x è $c(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappresenta il completamento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

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

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

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

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



1. $c(0) \cdot (1 - p) / p$
2. $c(0) / (1 - p)$
3. $c(0) / (p \cdot (1 - p))$

Risposta : 2

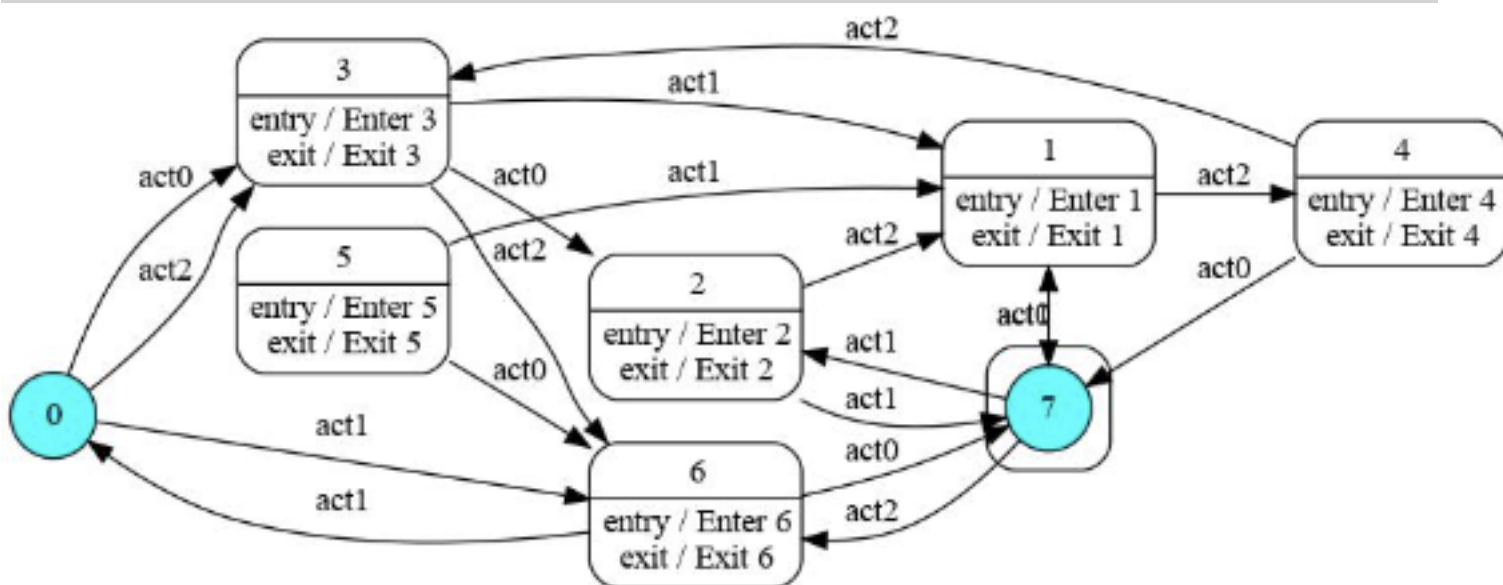
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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 act0



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.

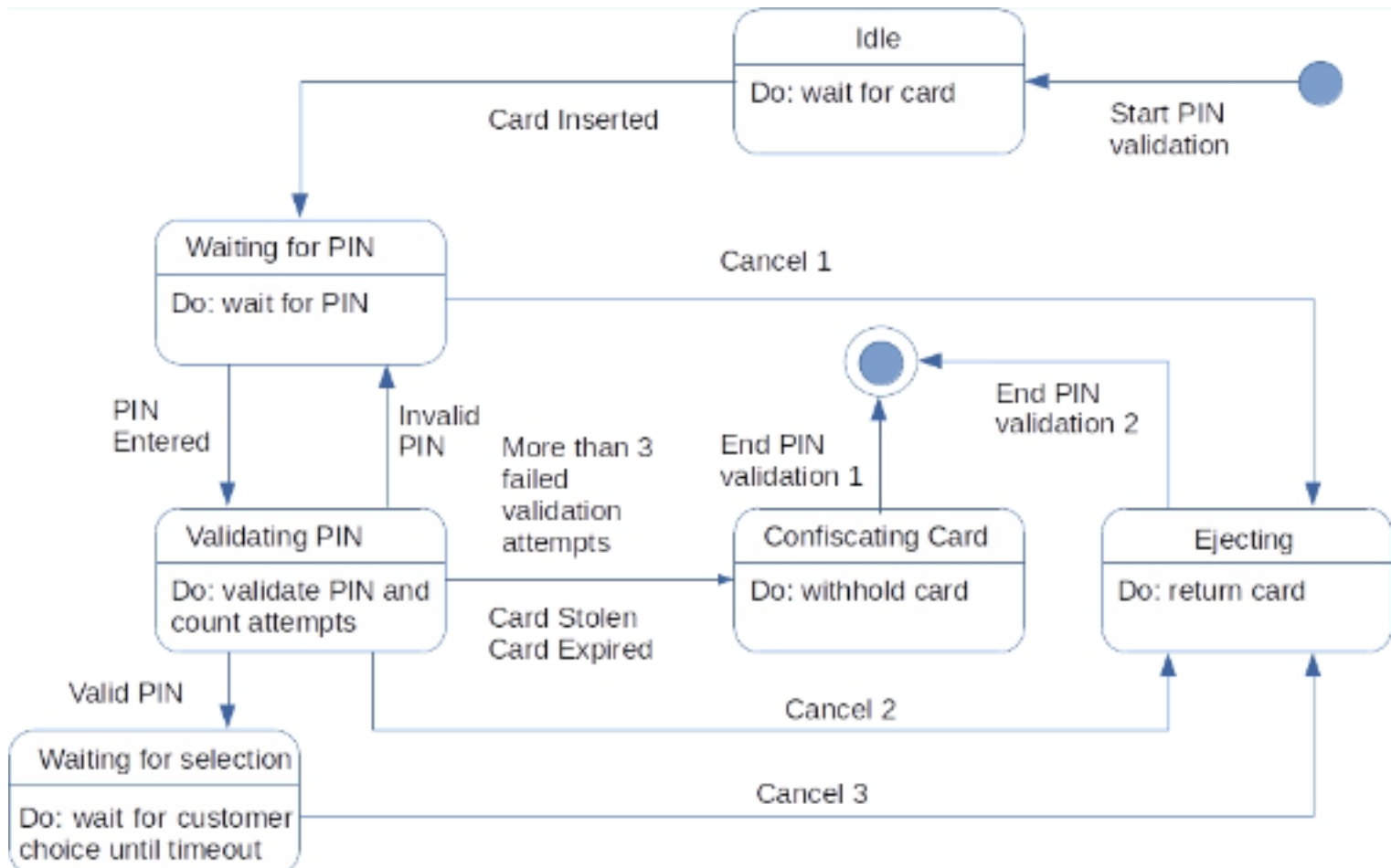
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

Test case 2: act0 act1 act2 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?



1.100%

2.80%

3.60%

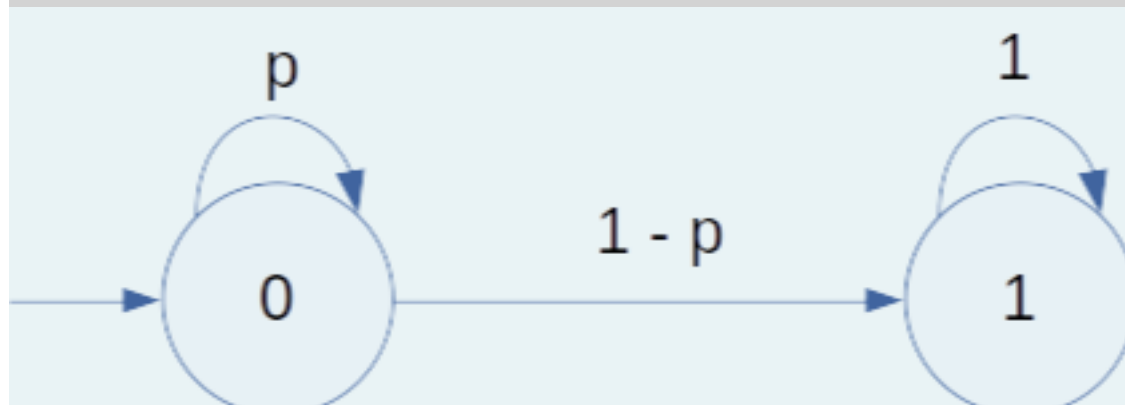
Risposta : 2

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: act2 act2

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

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



1.50%

2.70%

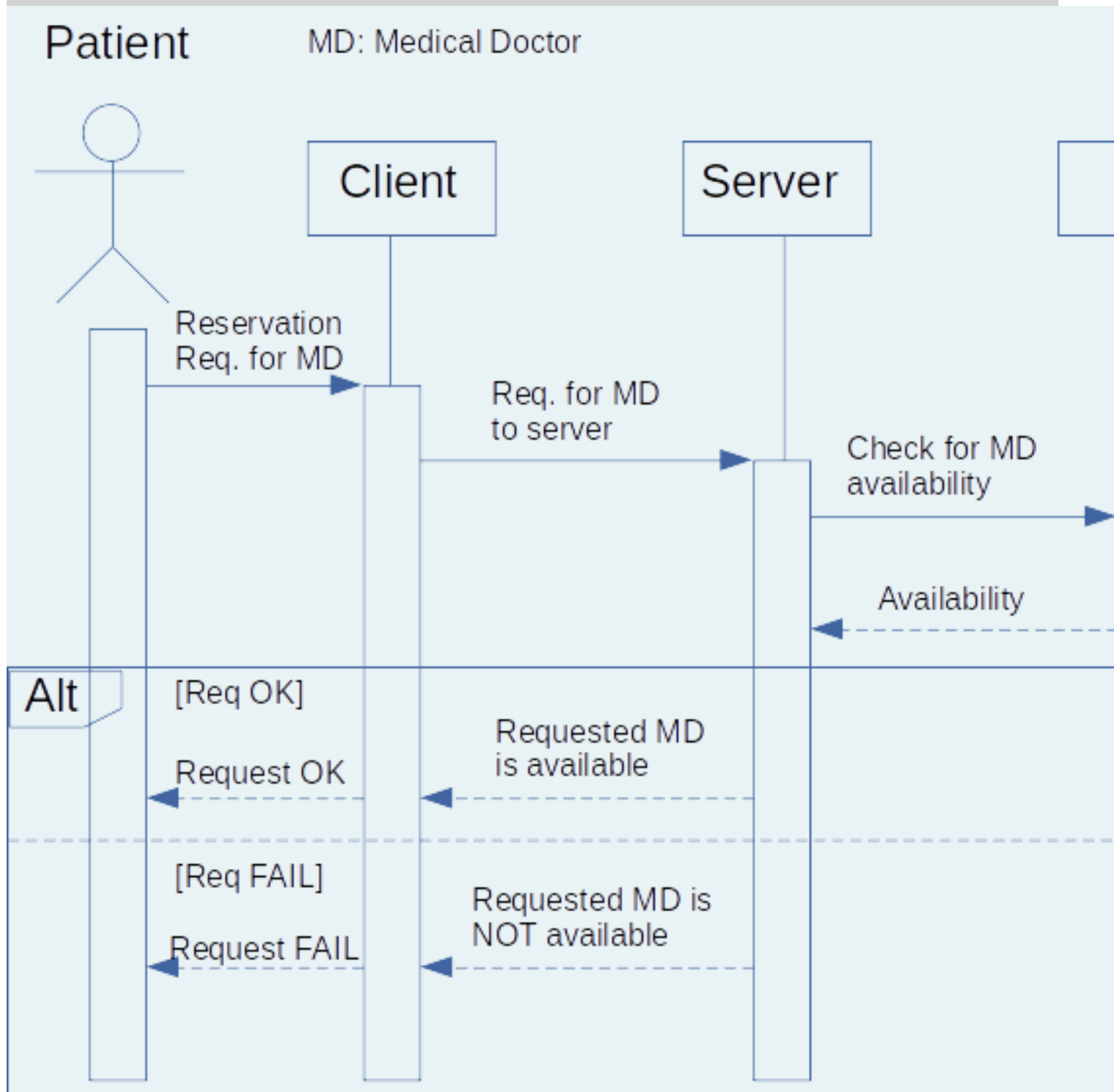
3.90%

Risposta : 1

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

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

Uno scenario è una sequenza di stati.



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.

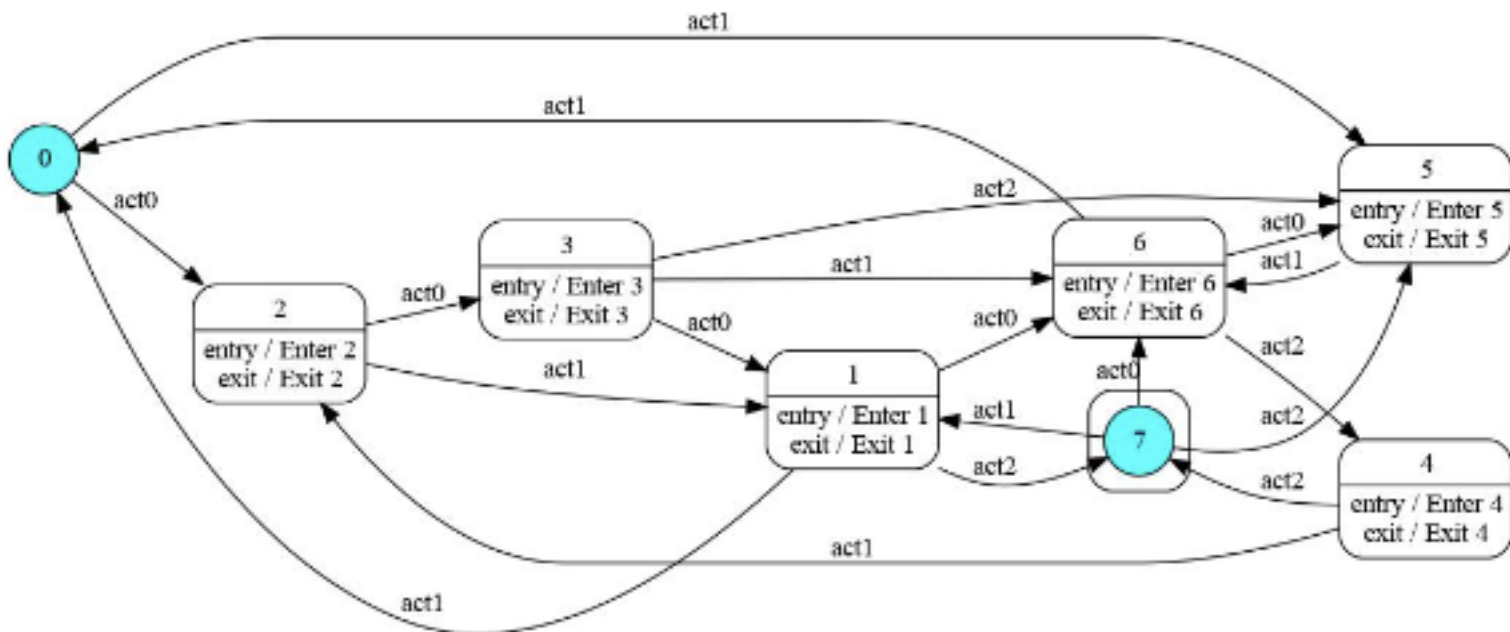
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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.

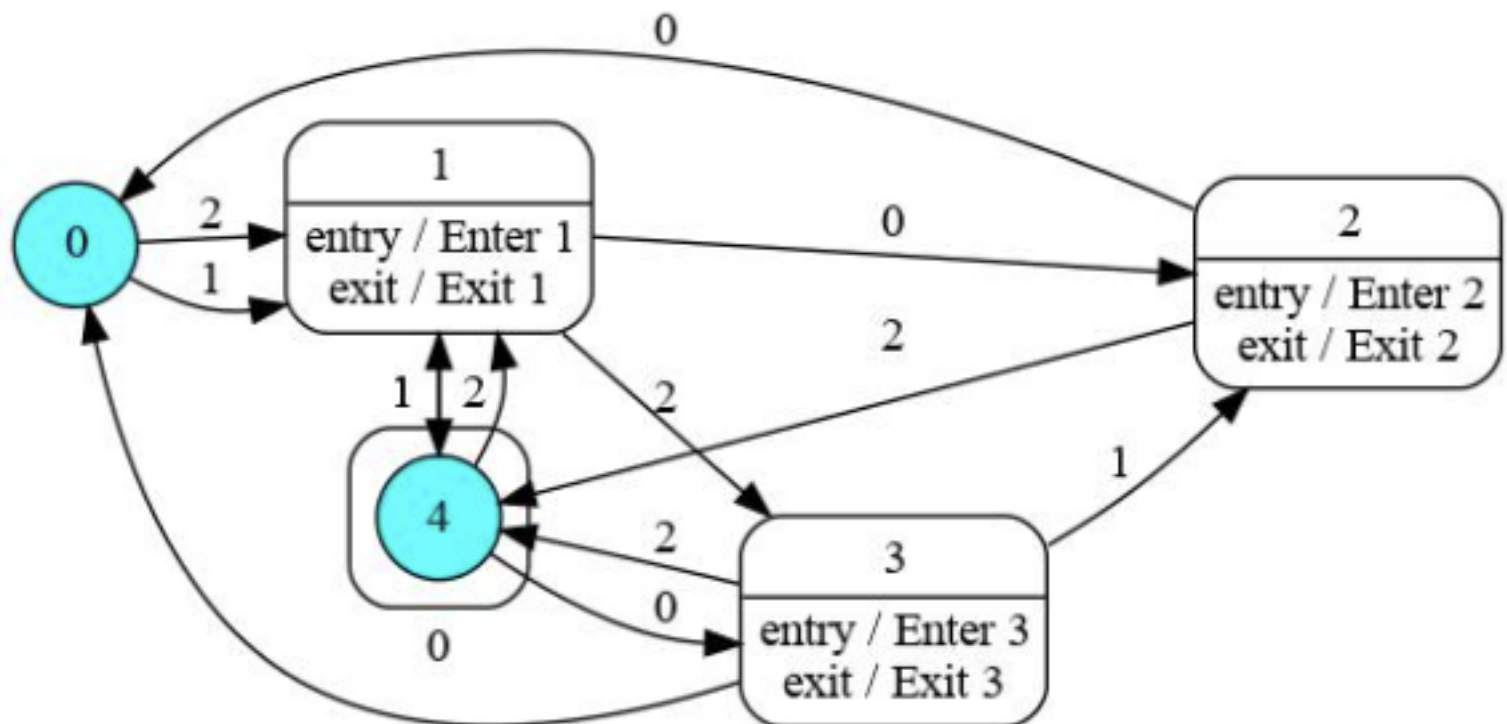
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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%

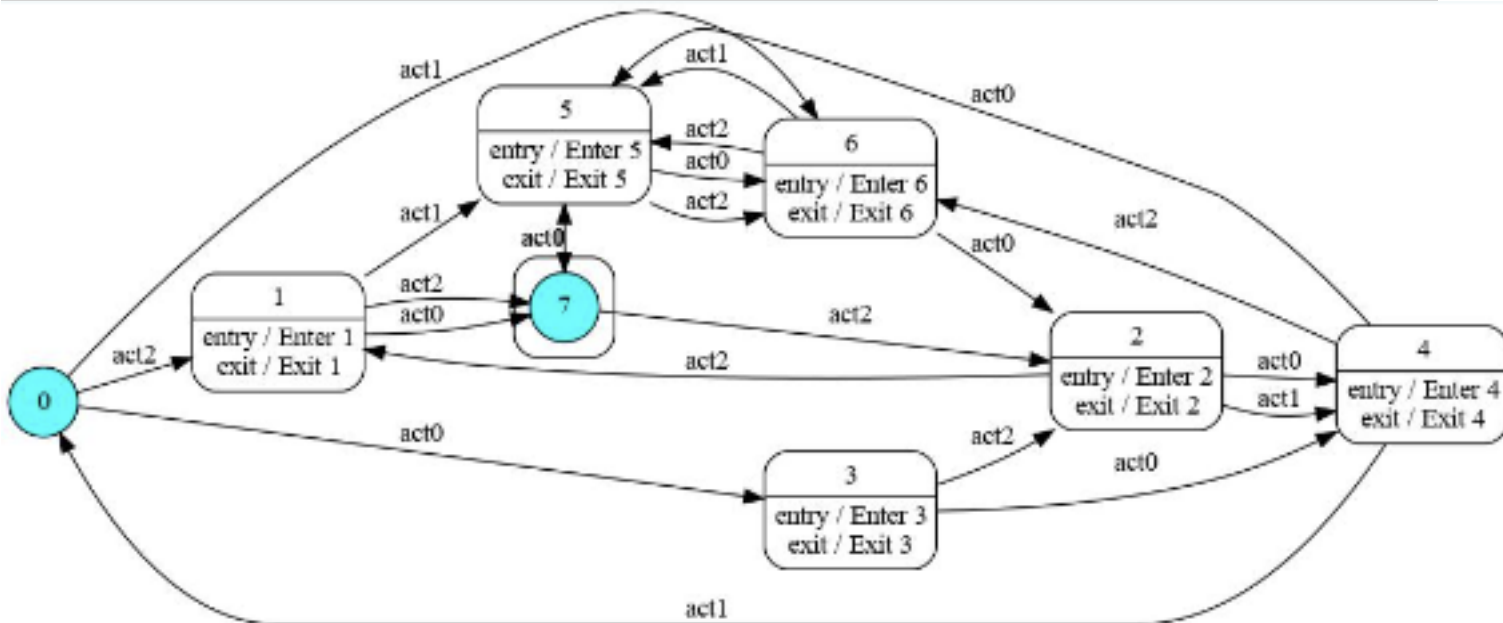
Risposta : 3

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 act2 act1 act2 act1 act2 act0 act1

Test case 2: act0 act2 act0

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



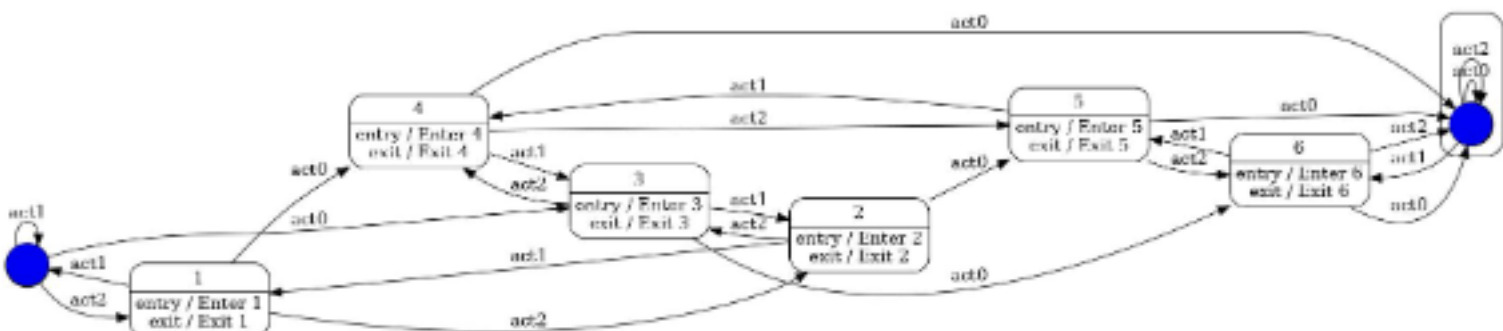
1.State coverage: 60%

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.



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.

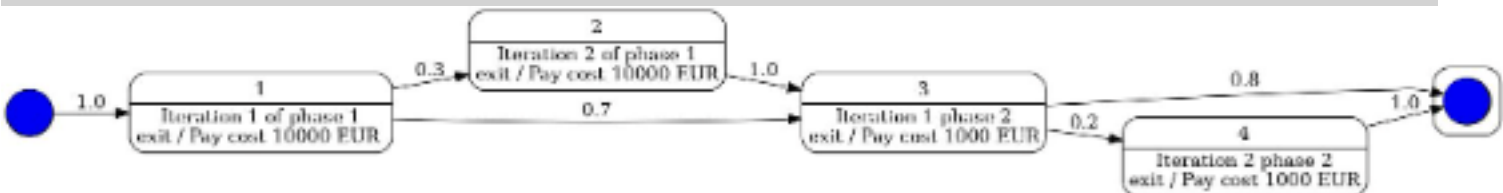
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: act1 act1 act1

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

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

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



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.

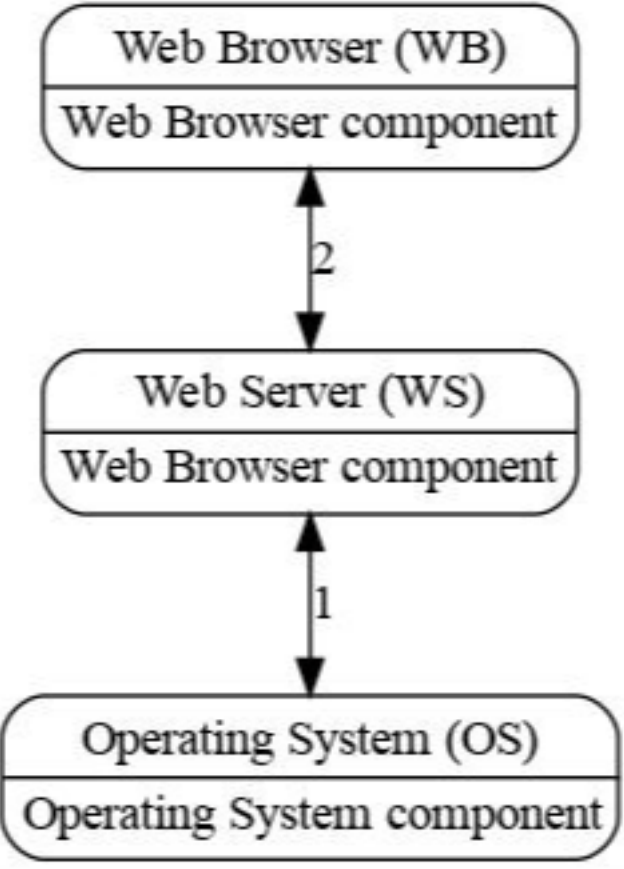
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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

Test case 2: act1

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

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



1.State coverage: 60%

2.State coverage: 75%

3.State coverage: 100%

Risposta : 2

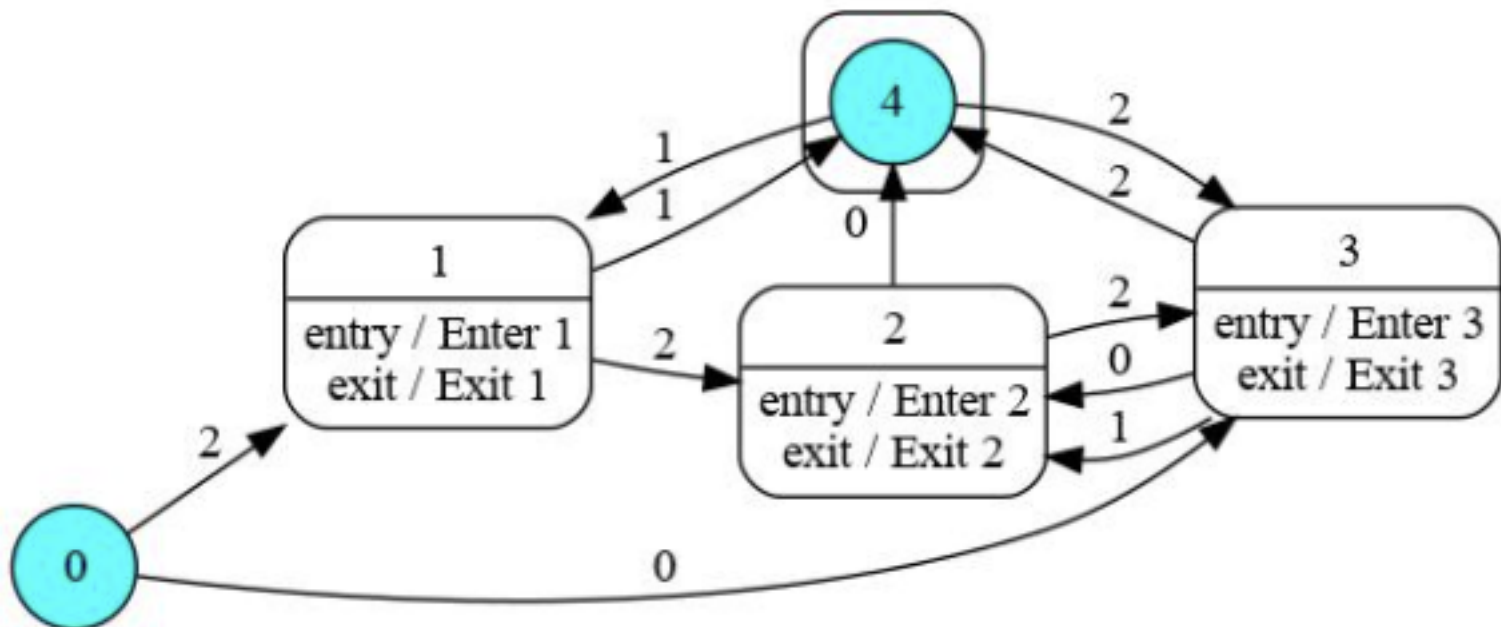
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in (0, 1). Il costo dello stato (fase) x è c(x). La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappresenta il completamento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

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

Quindi il costo C(X) della sequenza di stati X = x(0), x(1), x(2), 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. $c(0) \cdot (1 - p) / p$
2. $c(0) / (p \cdot (1 - p))$
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.

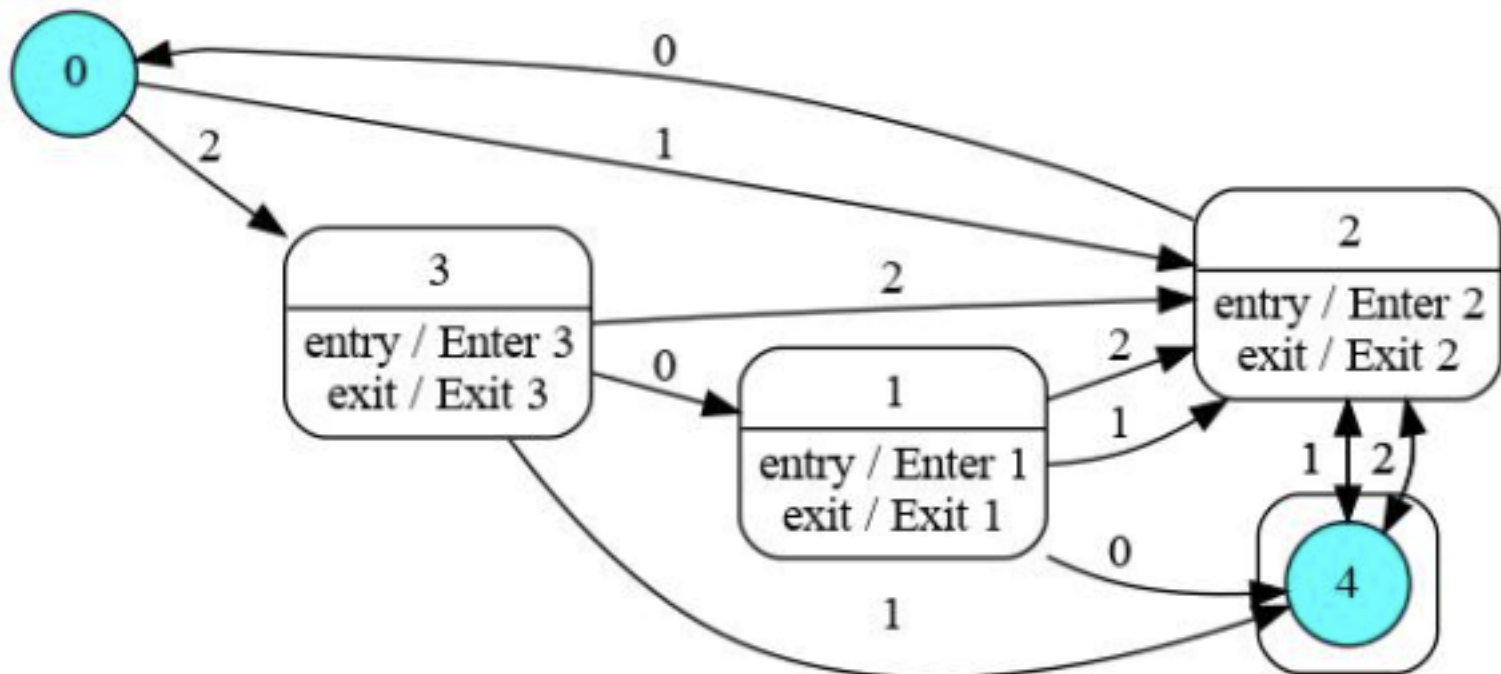
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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%

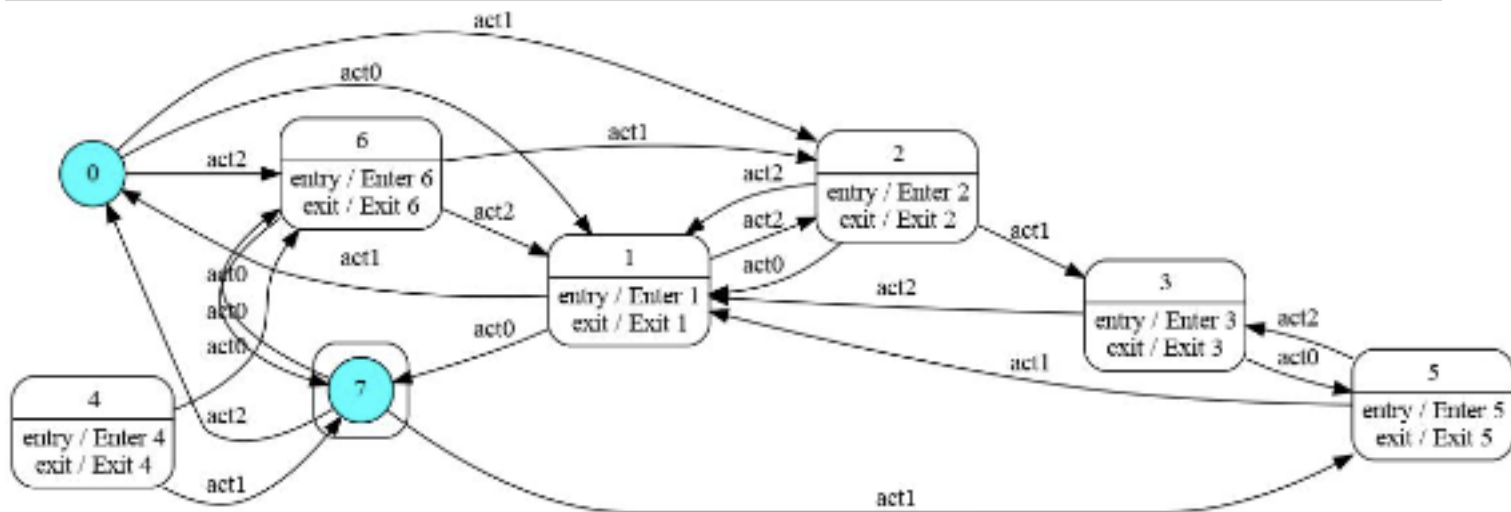
Risposta : 2

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 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: 60%

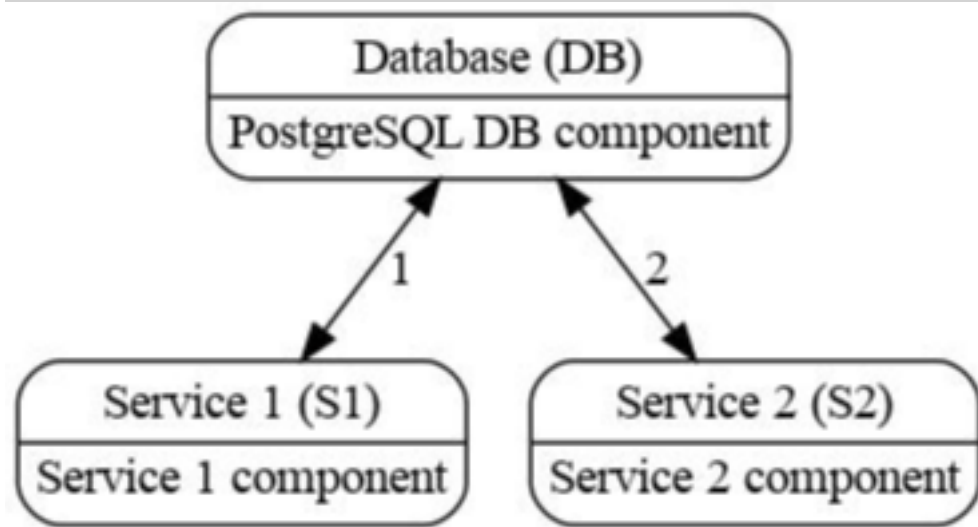
Risposta : 1

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

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

Uno scenario è una sequenza di stati.

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



1.0.27

2.0.03

3.0.07

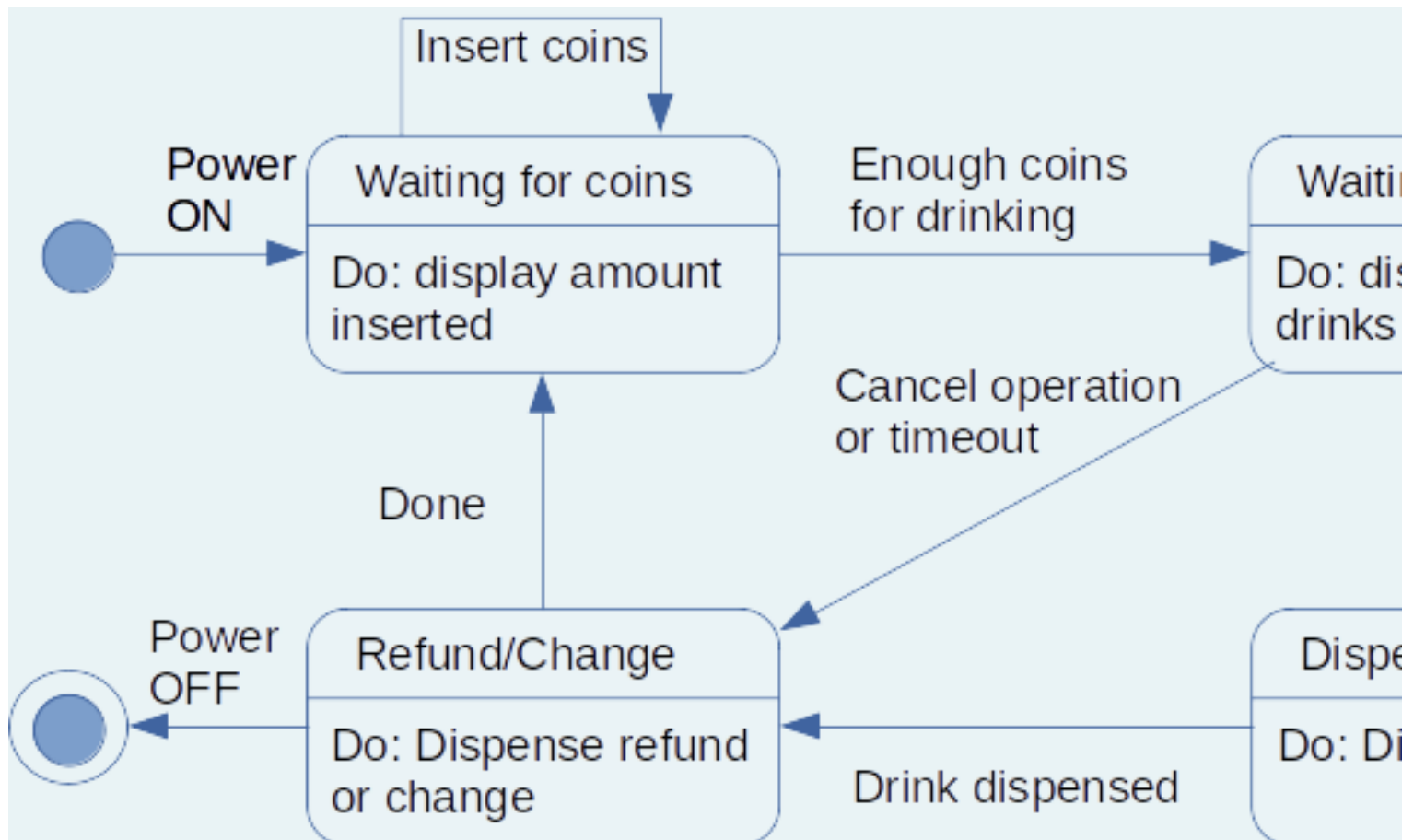
Risposta : 2

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

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

Uno scenario è una sequenza di stati.

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



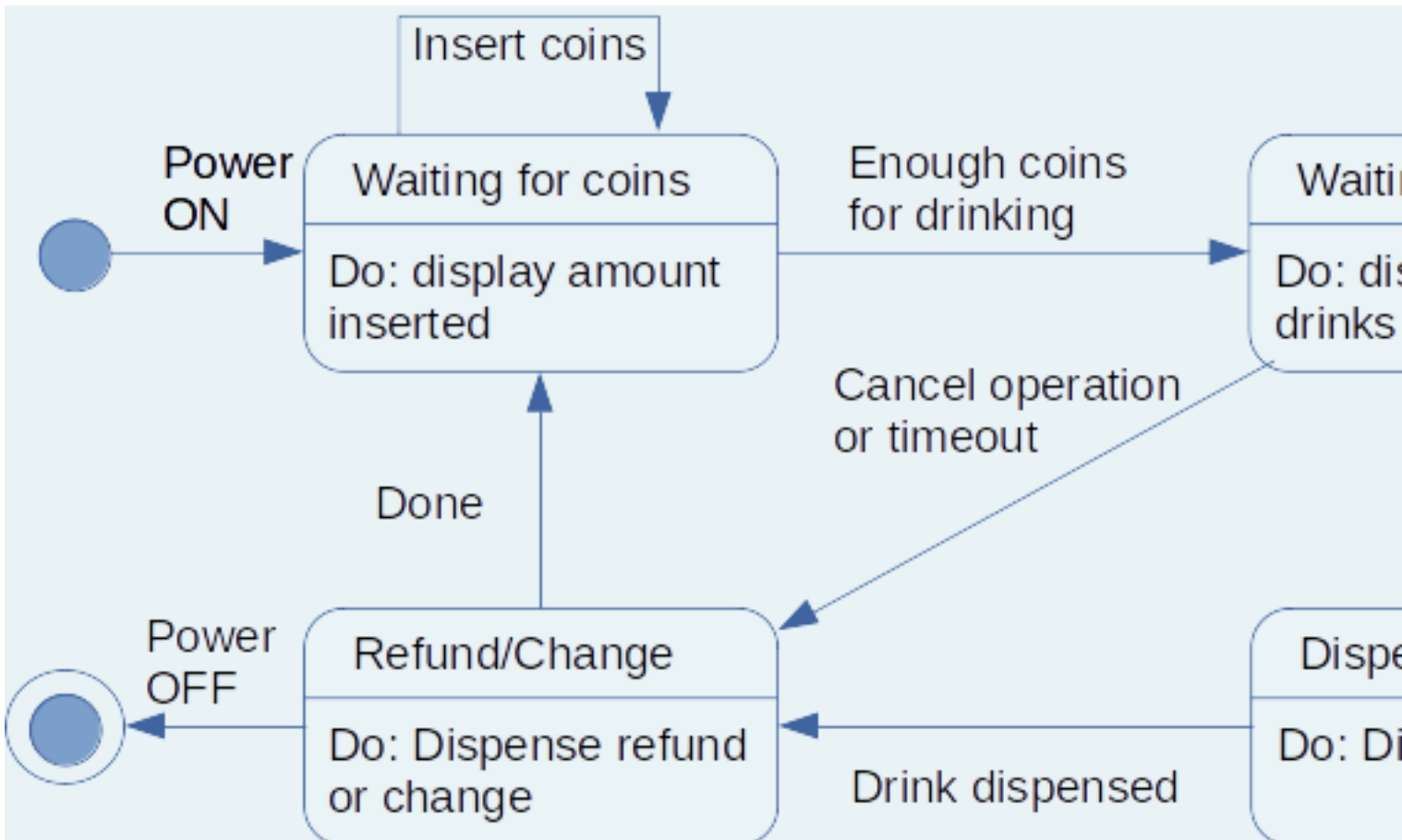
1.0.56

2.0.14

3.0.24

Risposta : 1

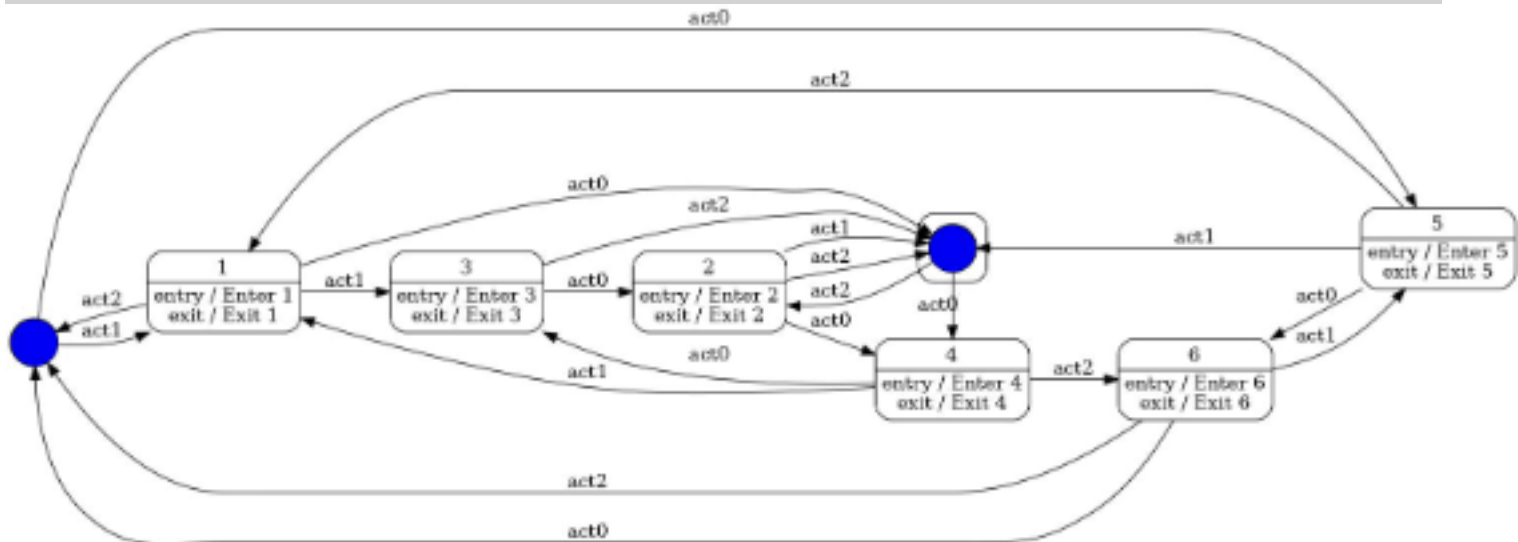
Si consideri la Markov chain in figura con stato iniziale 0 e p in $(0, 1)$. Quale delle seguenti formule calcola il valore atteso del numero di transizioni necessarie per lasciare lo stato 0.



1. $(1 - p)/p$
2. $1/(1 - p)$
3. $1/(p \cdot (1 - p))$

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


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

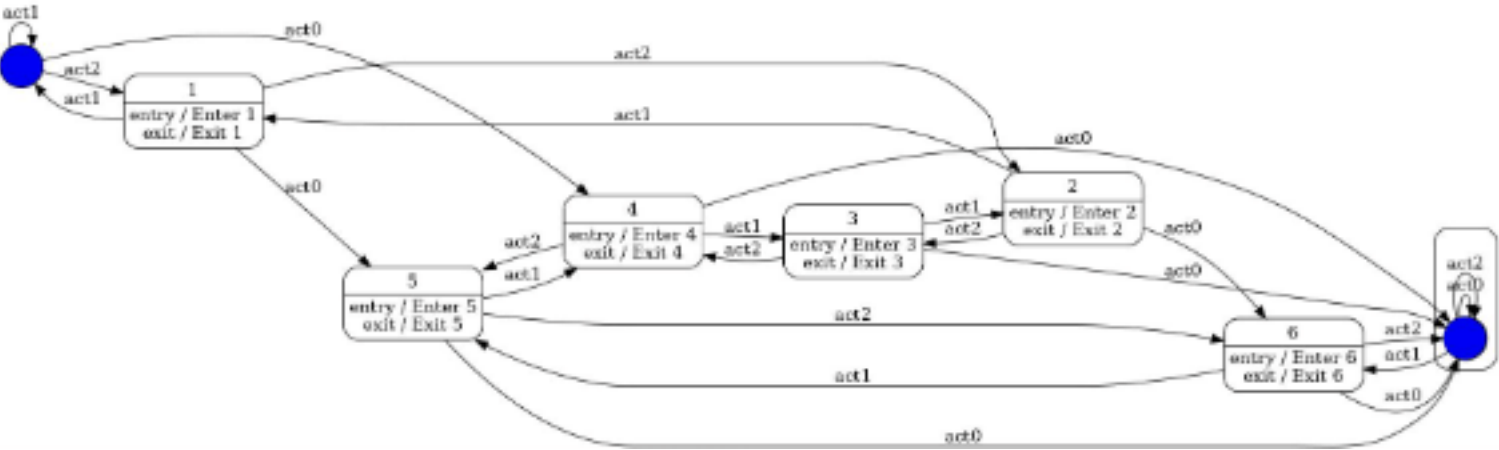
Risposta : 3

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

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

Uno scenario è una sequenza di stati.

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



- 1.0.12
- 2.0.32
- 3.0.08

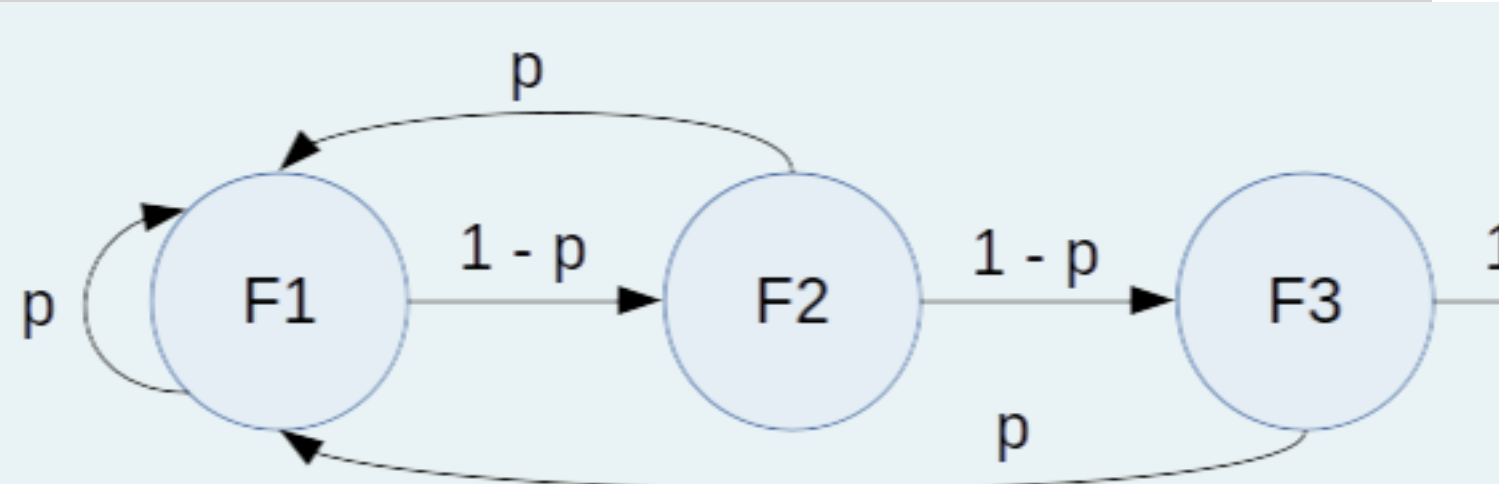
Risposta : 1

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 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.

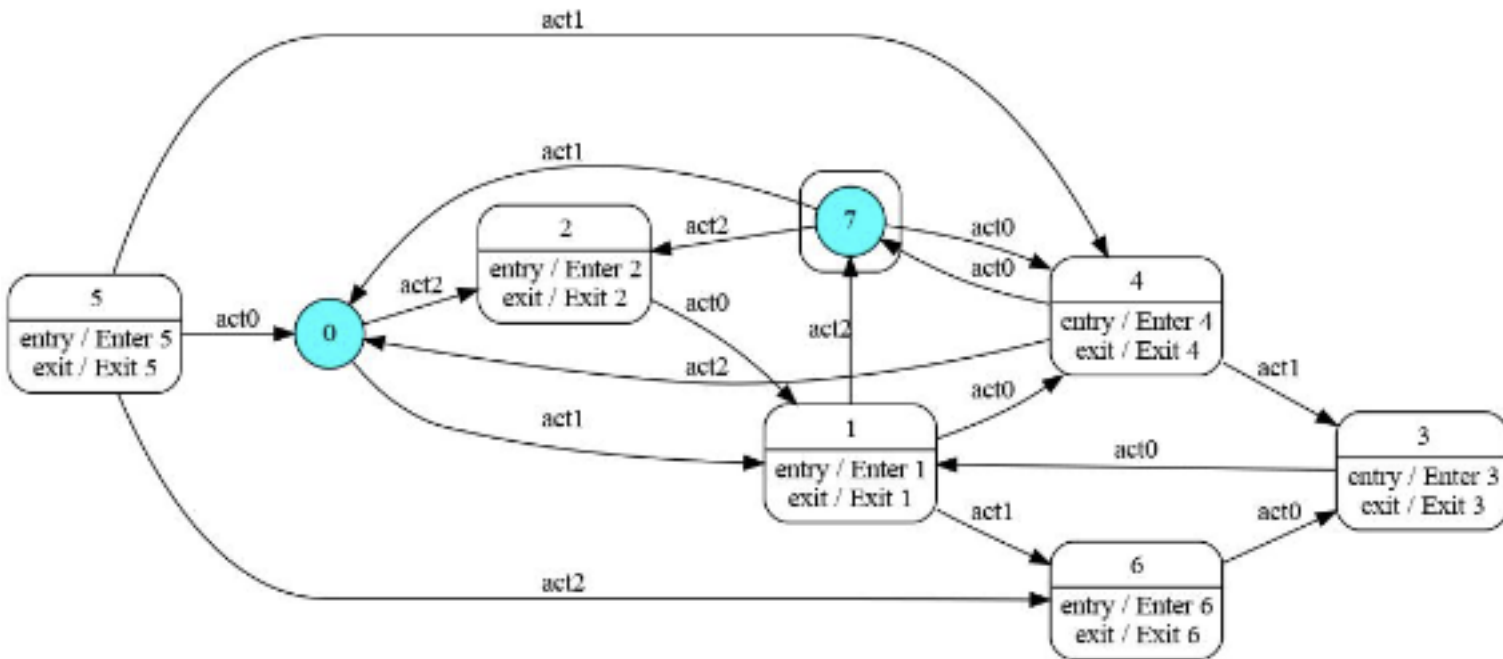
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

- Test case 1: act1 act2 act1 act2 act2

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: 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.

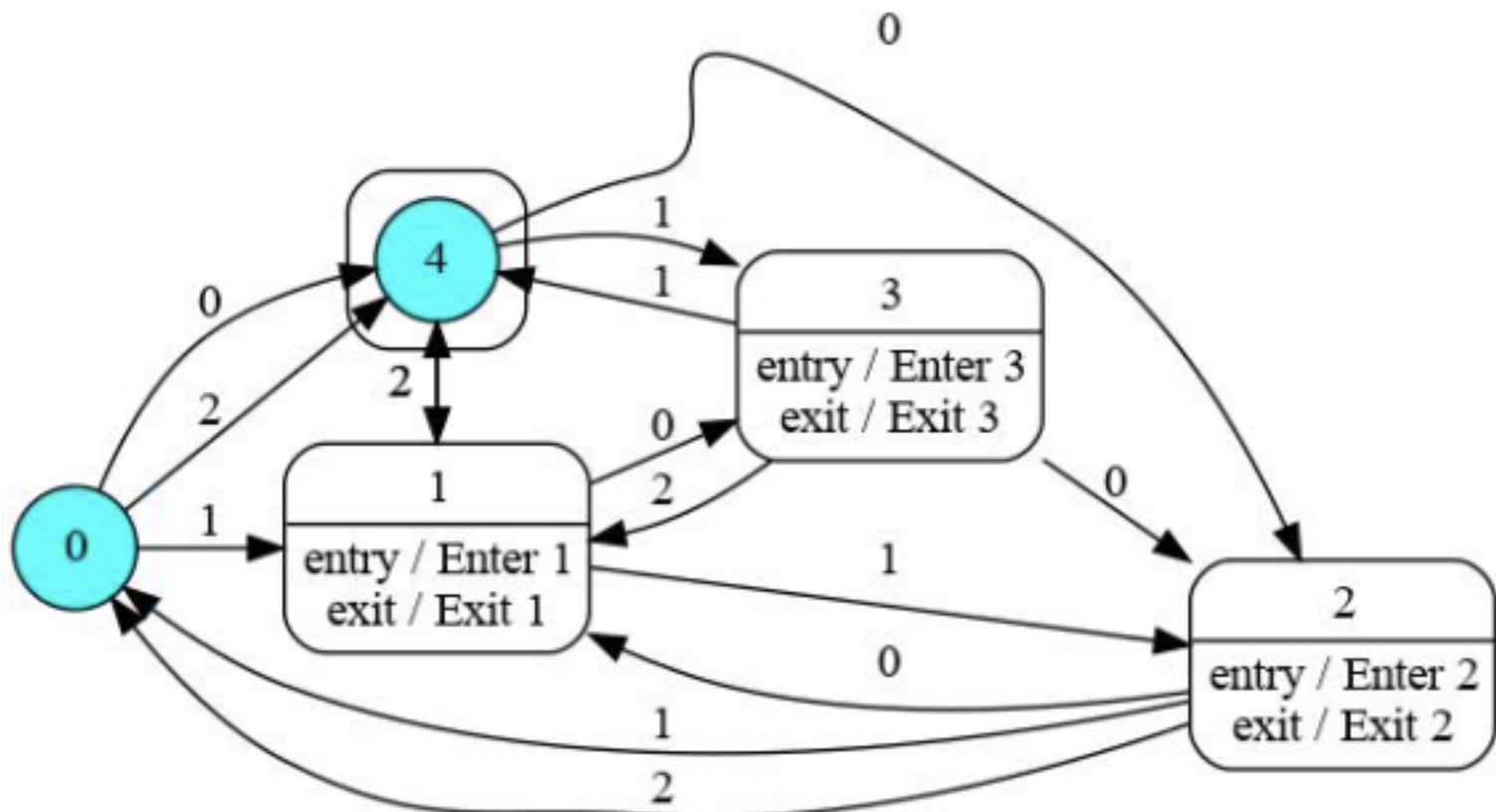
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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



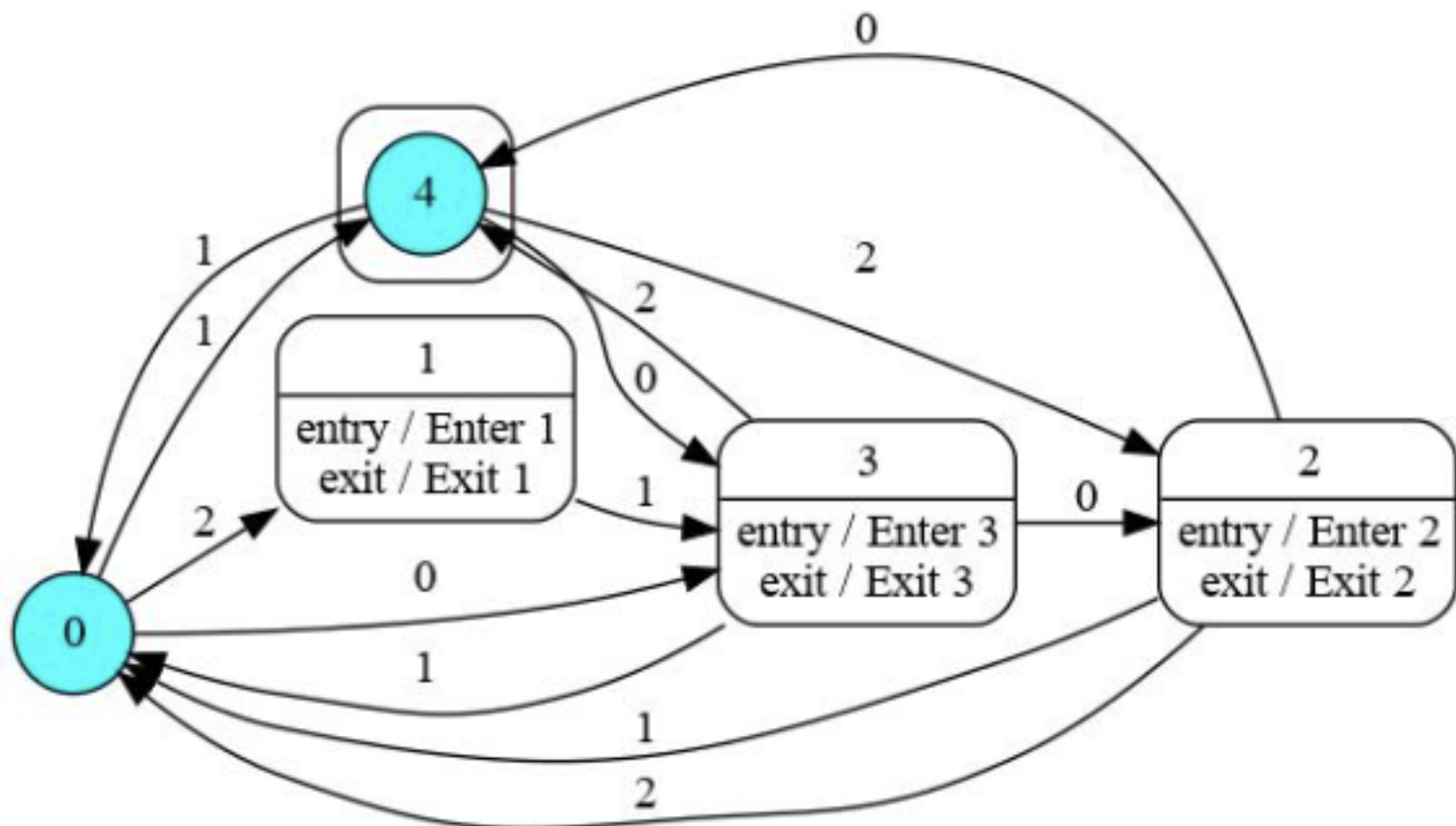
1.State coverage: 25%

2.State coverage: 80%

3.State coverage: 60%

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

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

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

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

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

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

Risposta : 3

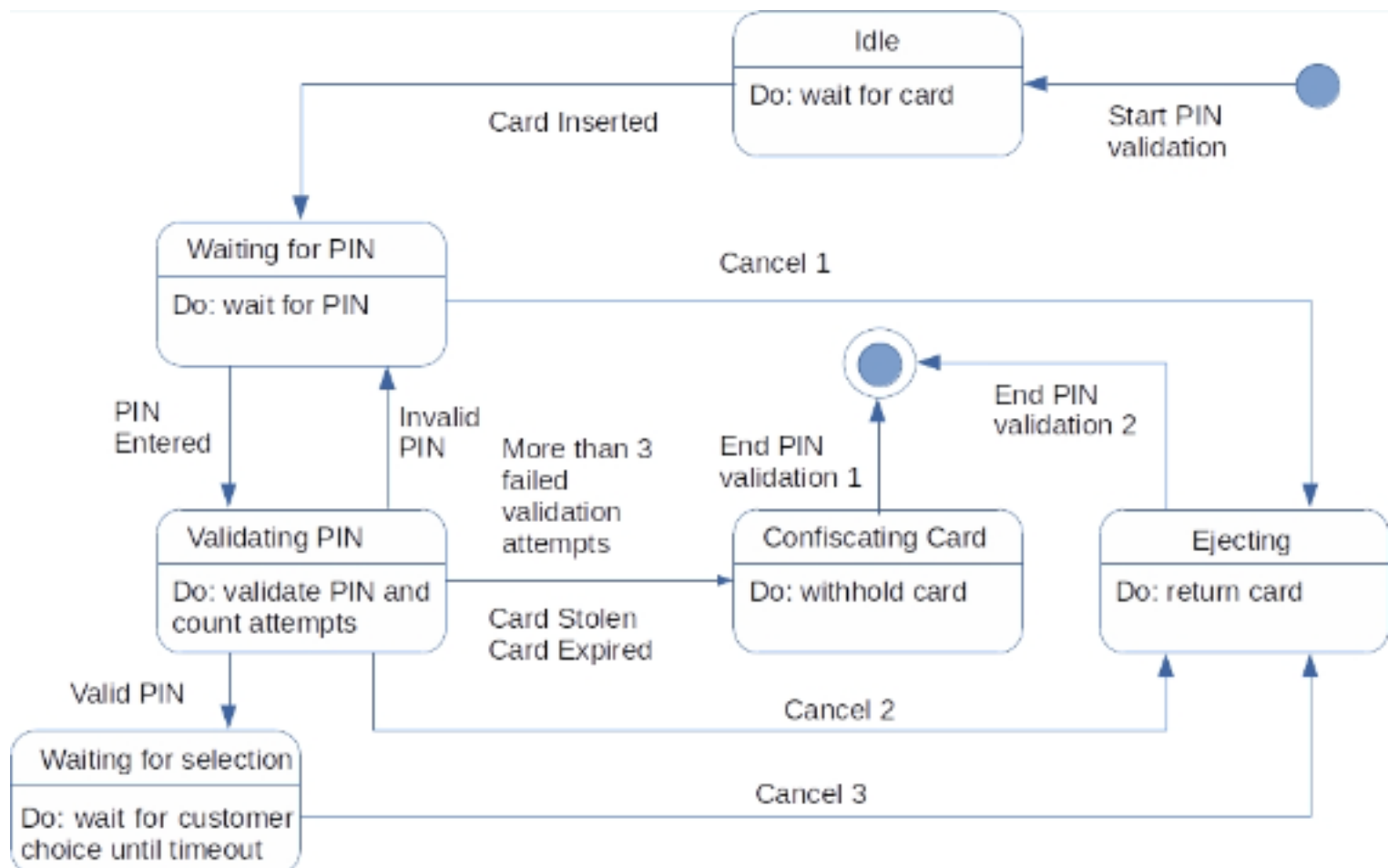
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il tempo necessario per completare la fase x è $\text{time}(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase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

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

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

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

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



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

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

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

Risposta : 3

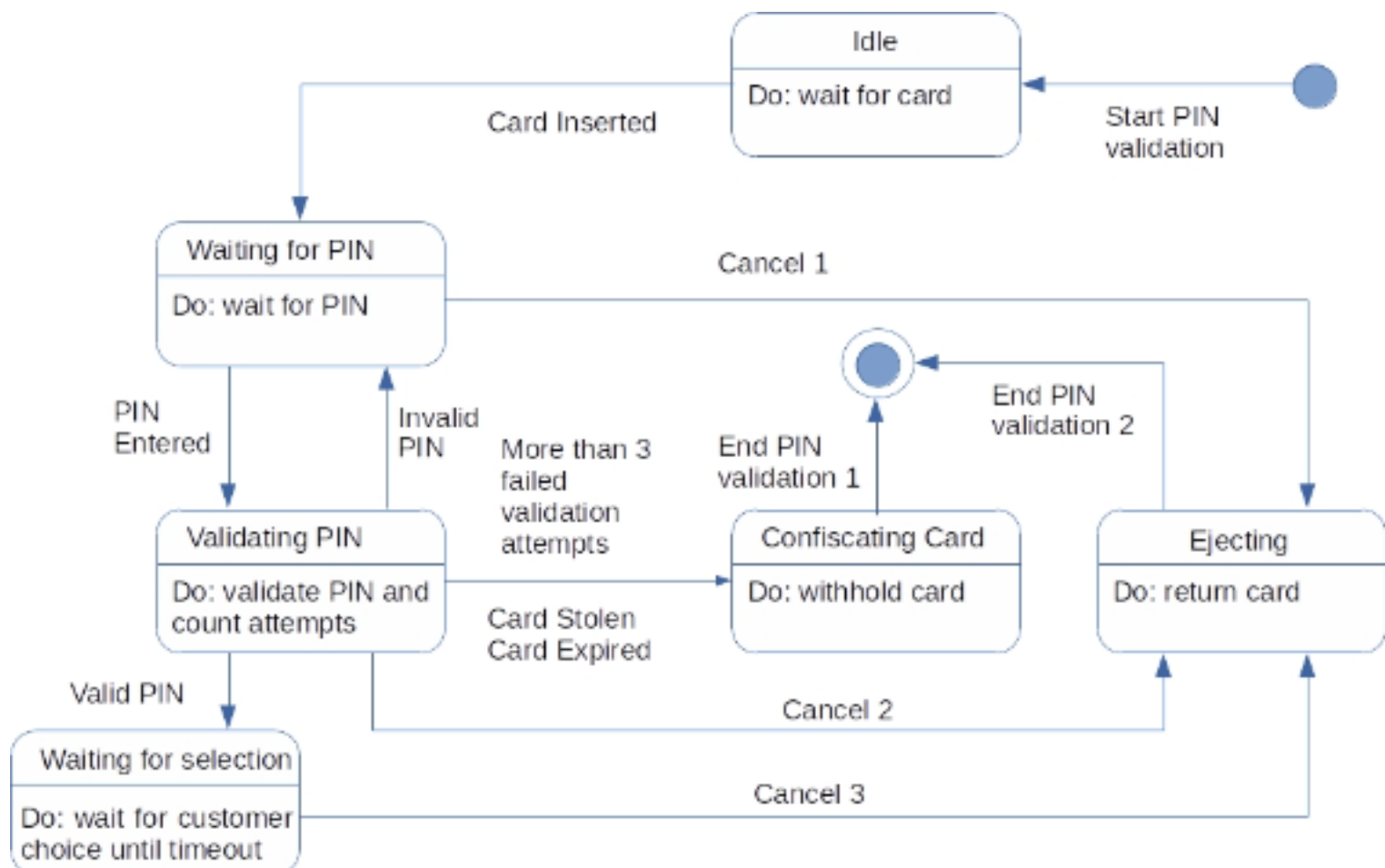
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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.

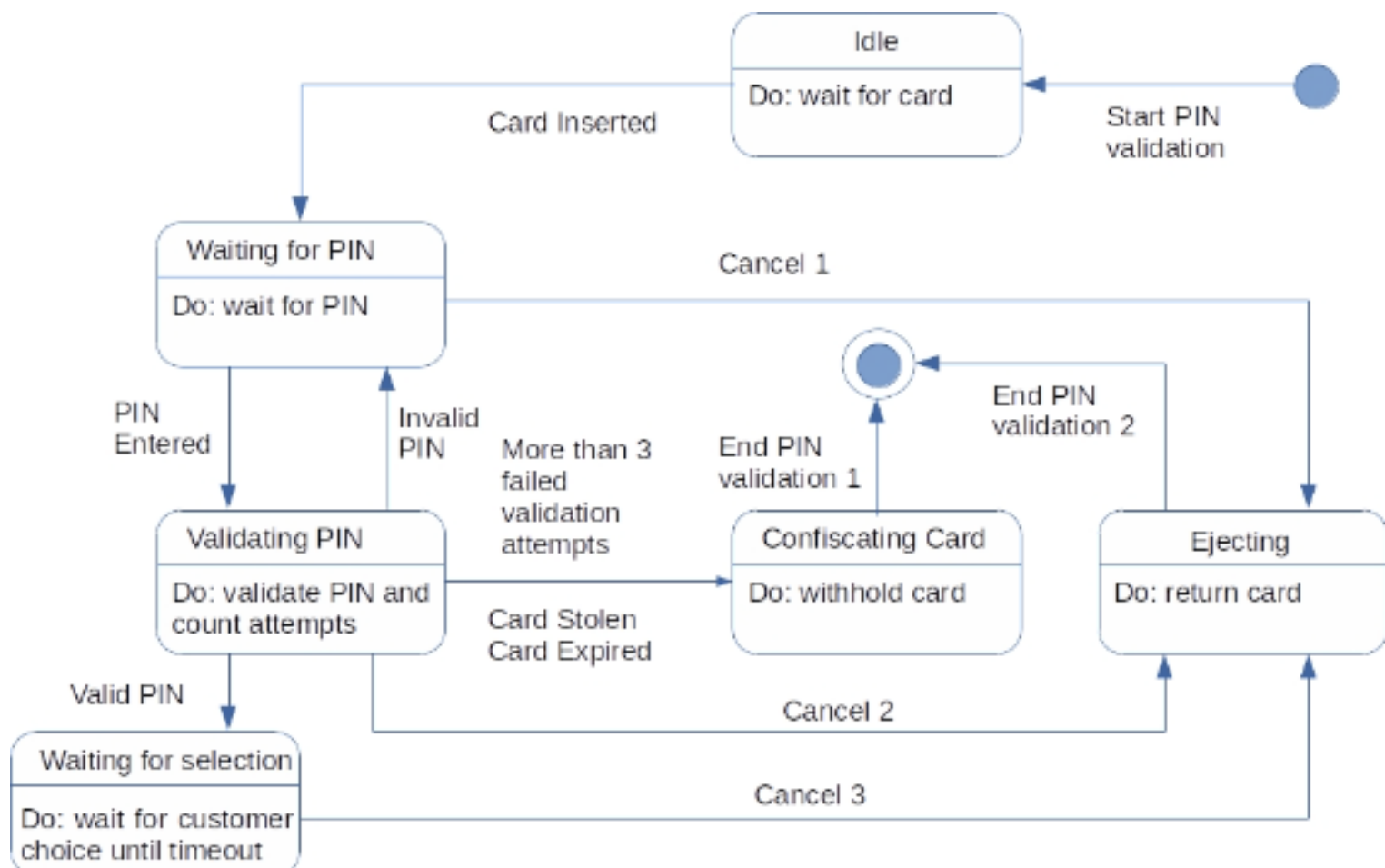
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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%

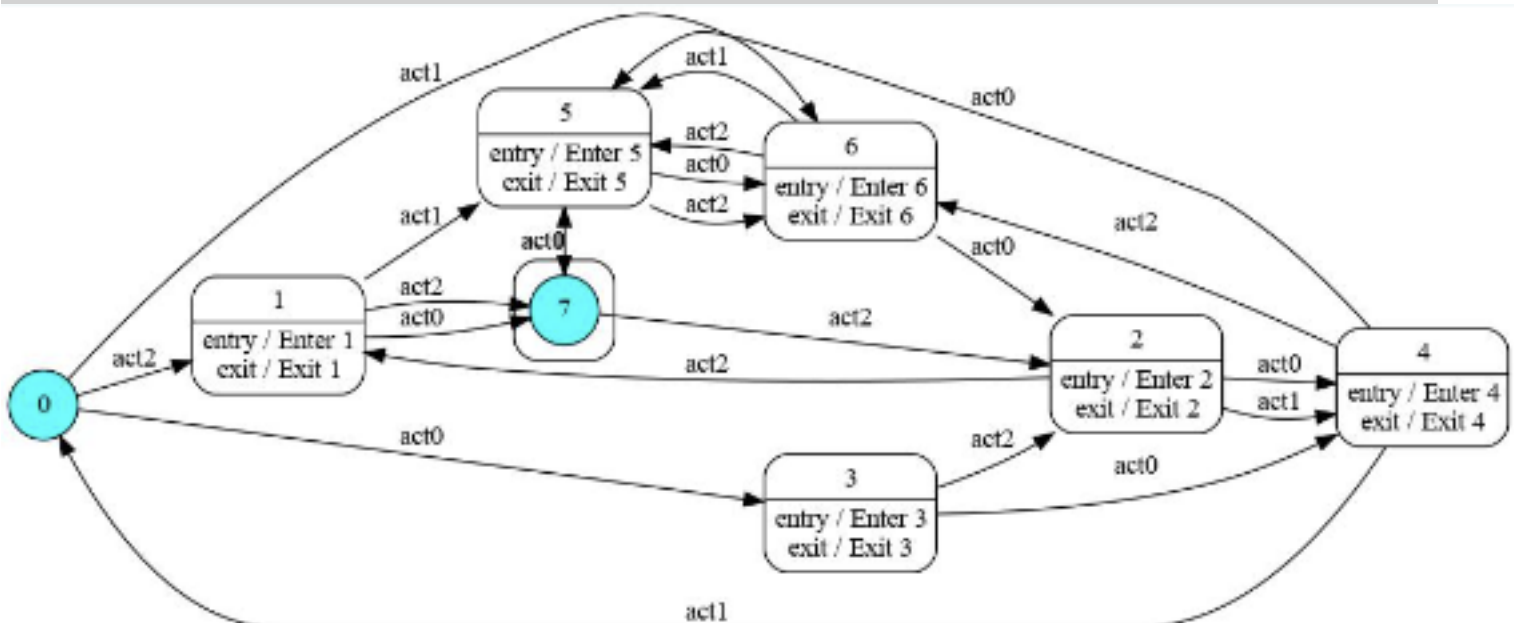
Risposta : 1

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

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

Uno scenario è una sequenza di stati.

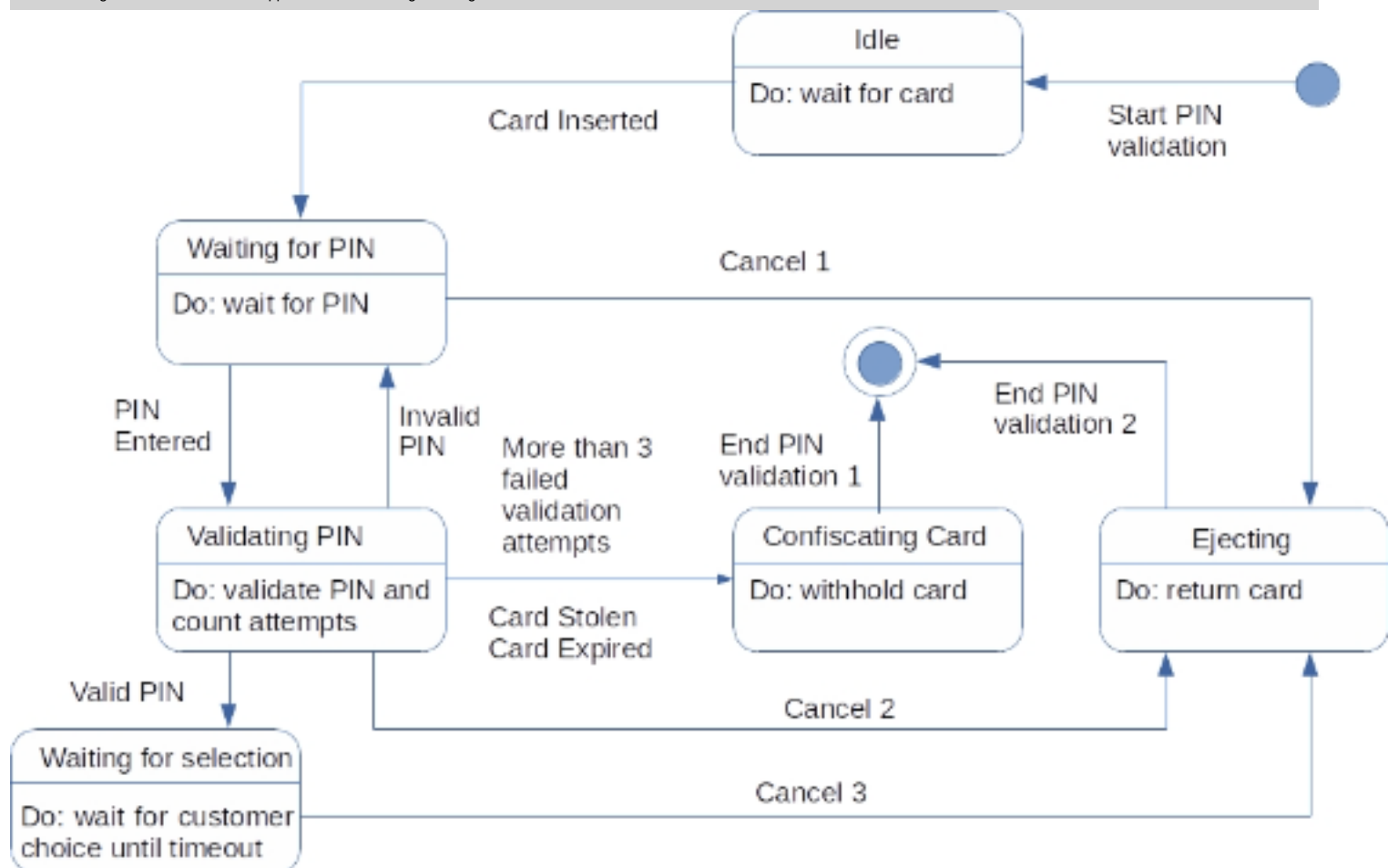
Qual'è la probabilità dello scenario: 1, 2, 3? In altri termini, qual'è la probabilità che non sia necessario ripetere la prima fase (ma non la seconda) ?



1.0.42

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

```

```

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

```

```

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

```

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.

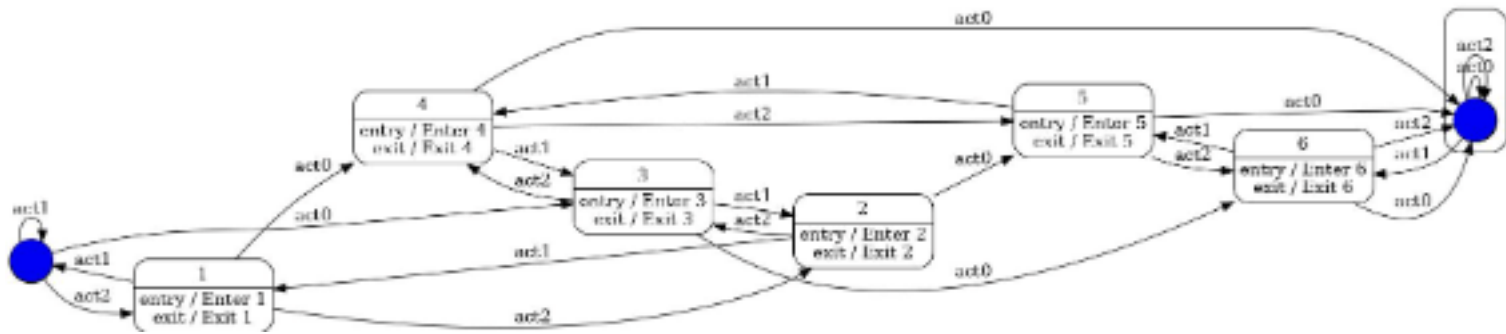
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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%

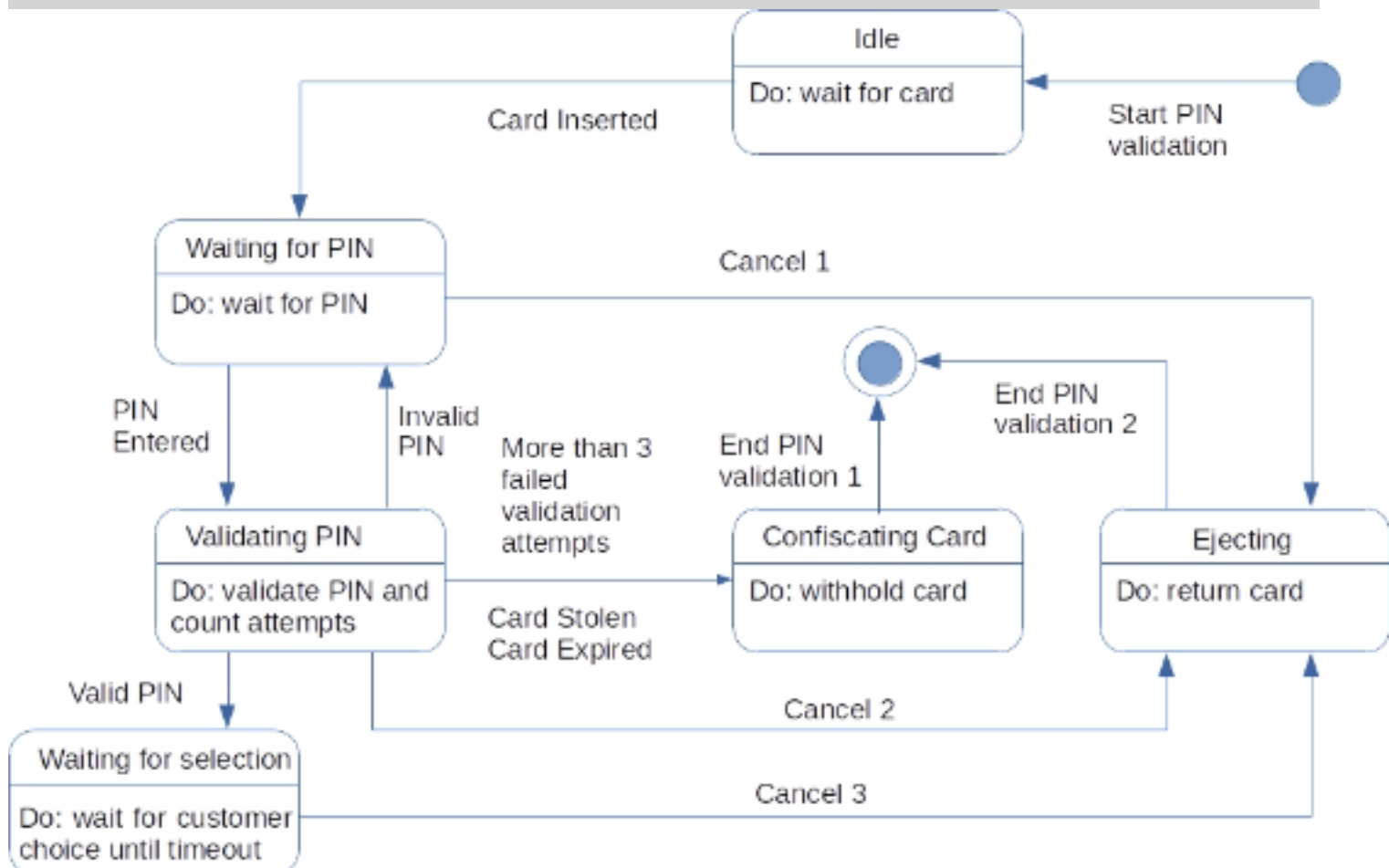
Risposta : 3

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.

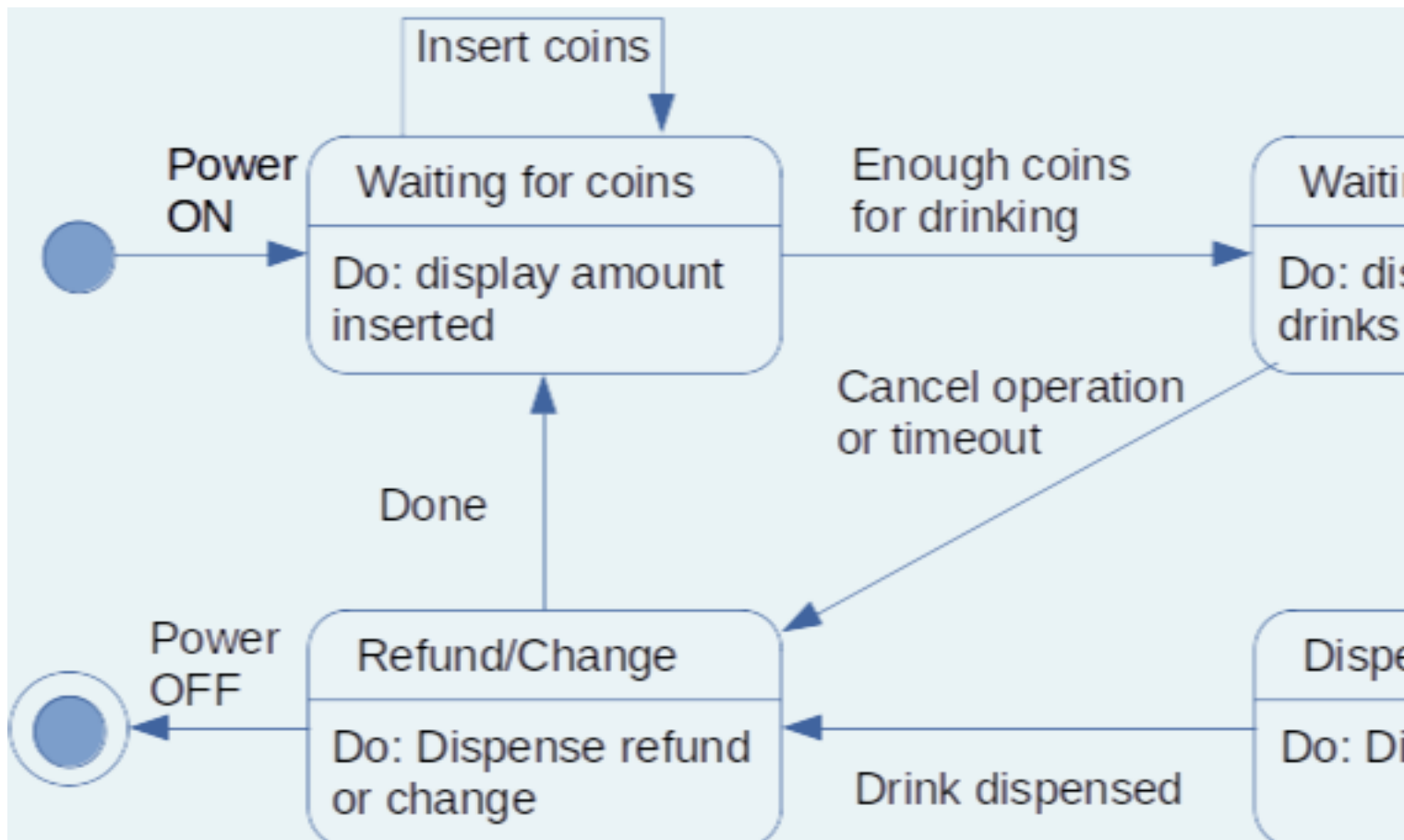
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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%

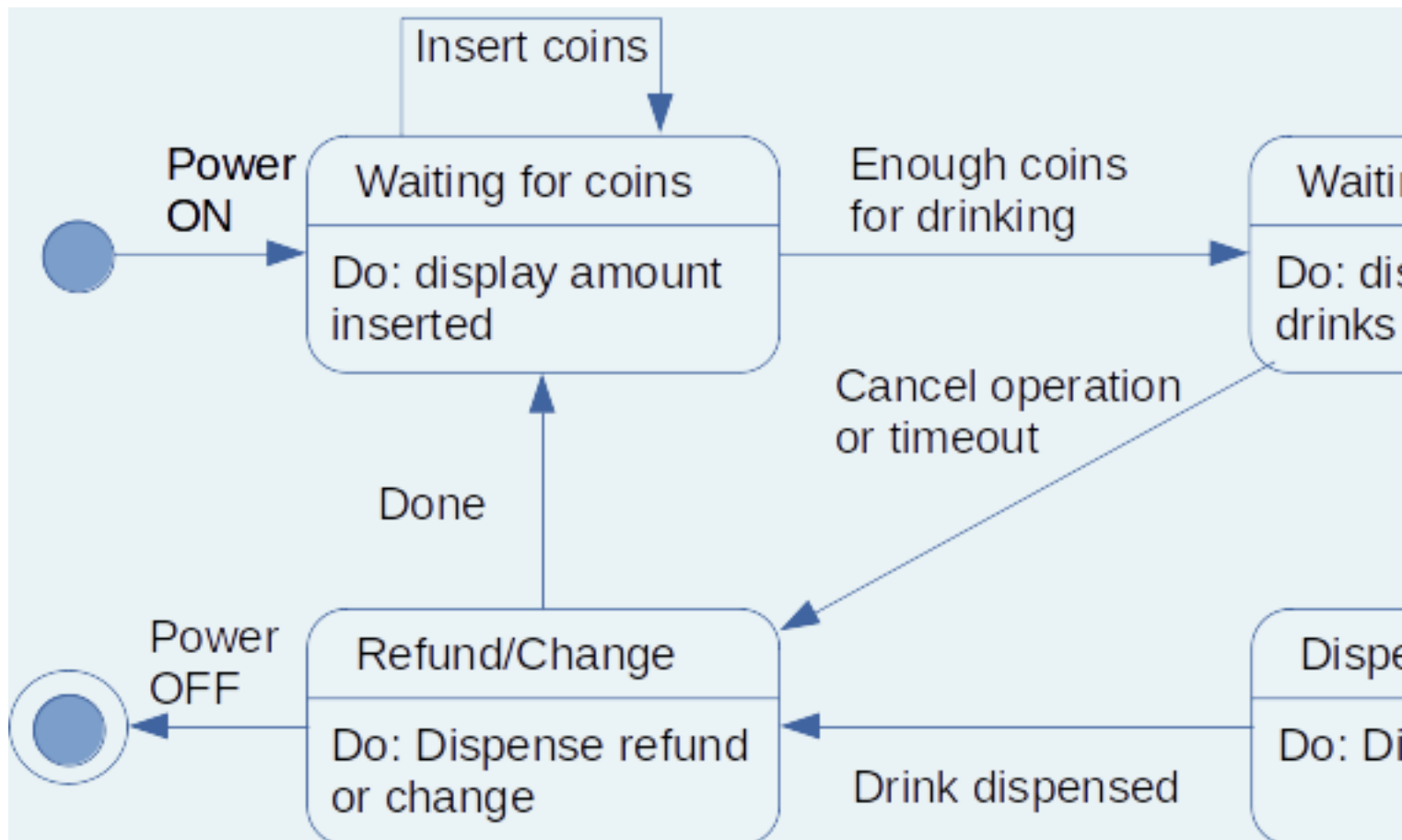
Risposta : 3

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

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

Uno scenario è una sequenza di stati.

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



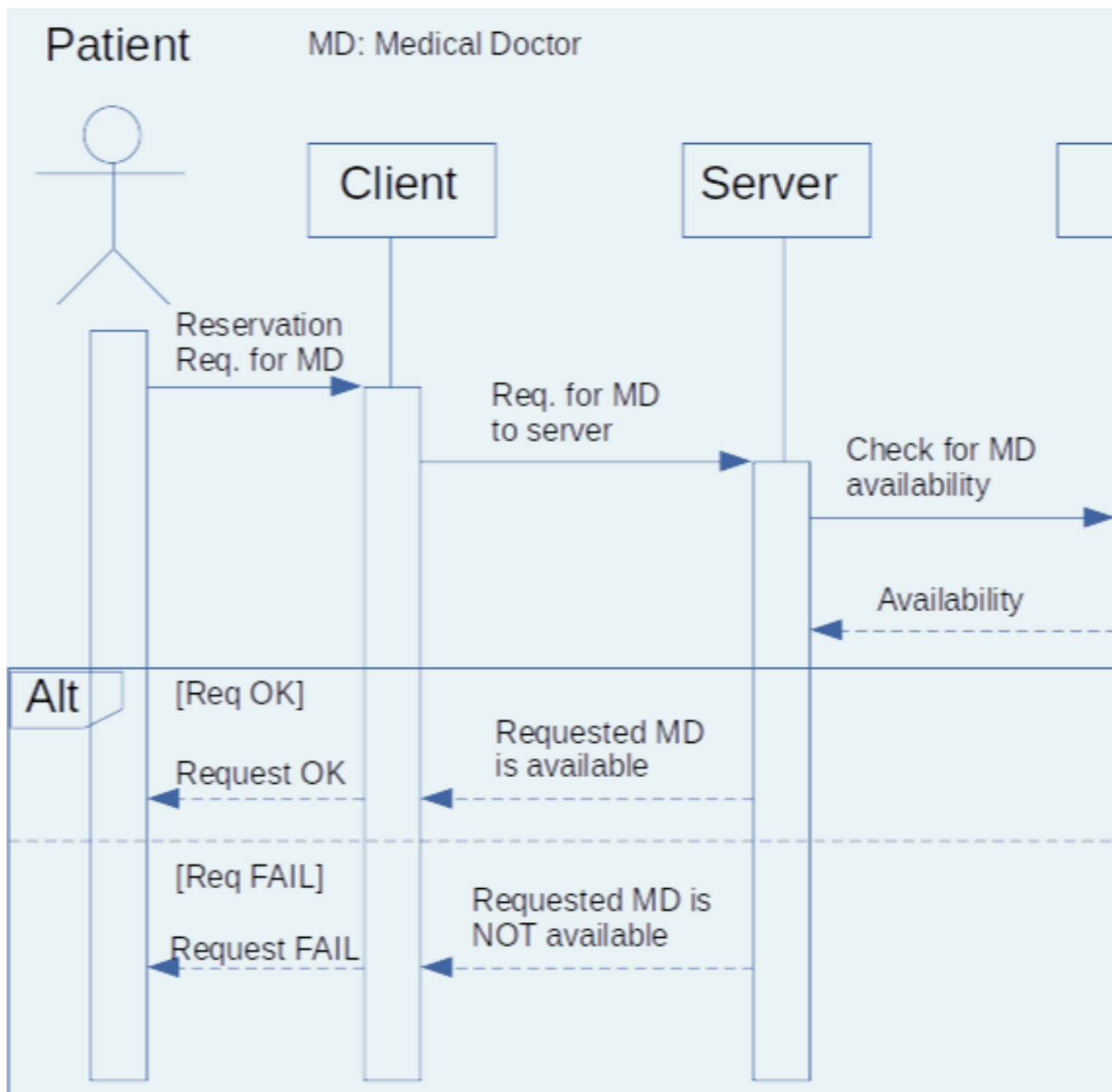
1.0.12

2.0.32

3.0.08

Risposta : 1

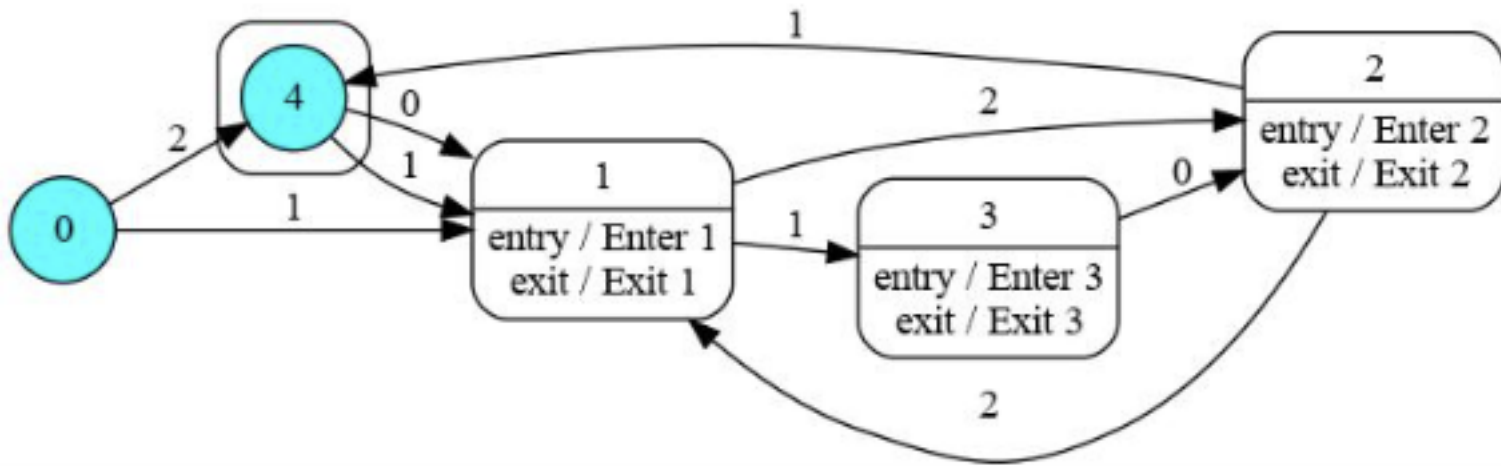
Si consideri la Markov chain in figura con stato iniziale 0 e p in $(0, 1)$. Quale delle seguenti formule calcola il valore atteso del numero di transizioni necessarie per lasciare lo stato 0.



1. $(1 - p)/p$
2. $1/(1 - p)$
3. $1/(p \cdot (1 - p))$

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

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

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

Risposta : 1

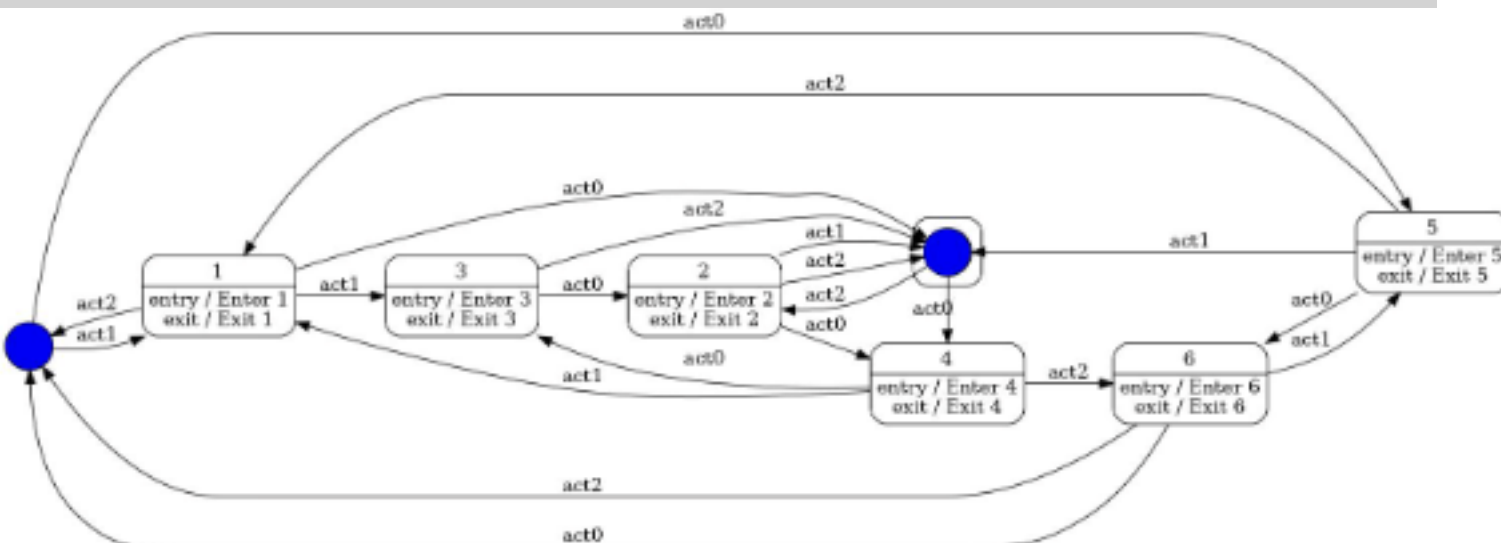
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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.

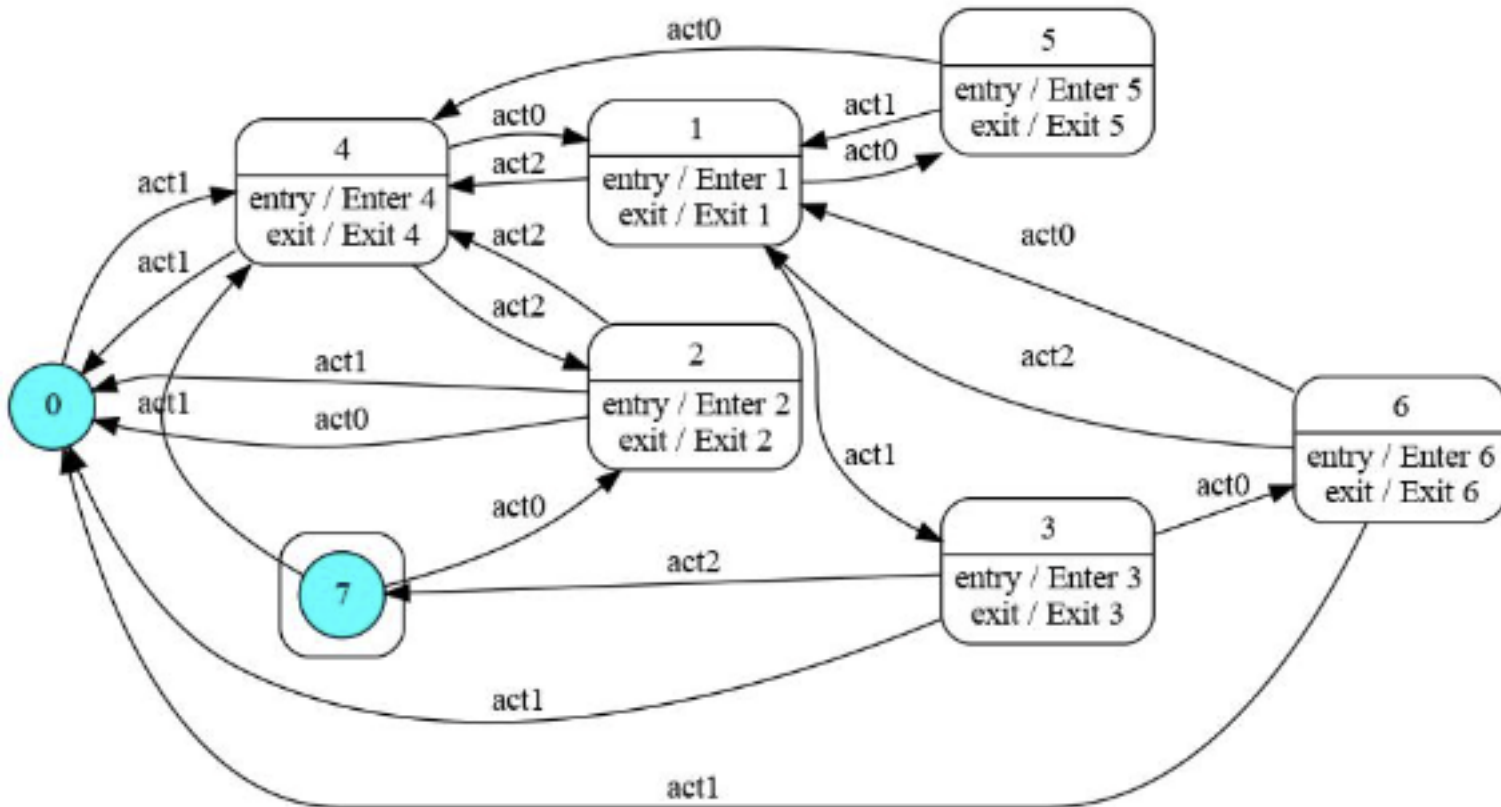
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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%

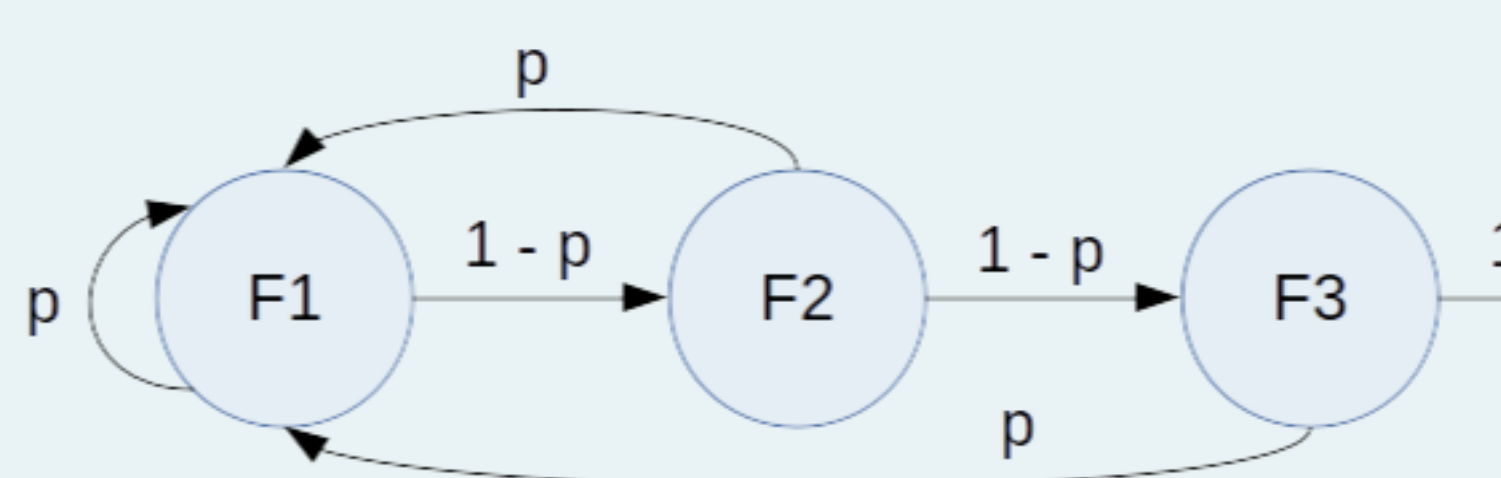
Risposta : 3

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

Ad esempio lo state diagram in 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 termini, 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

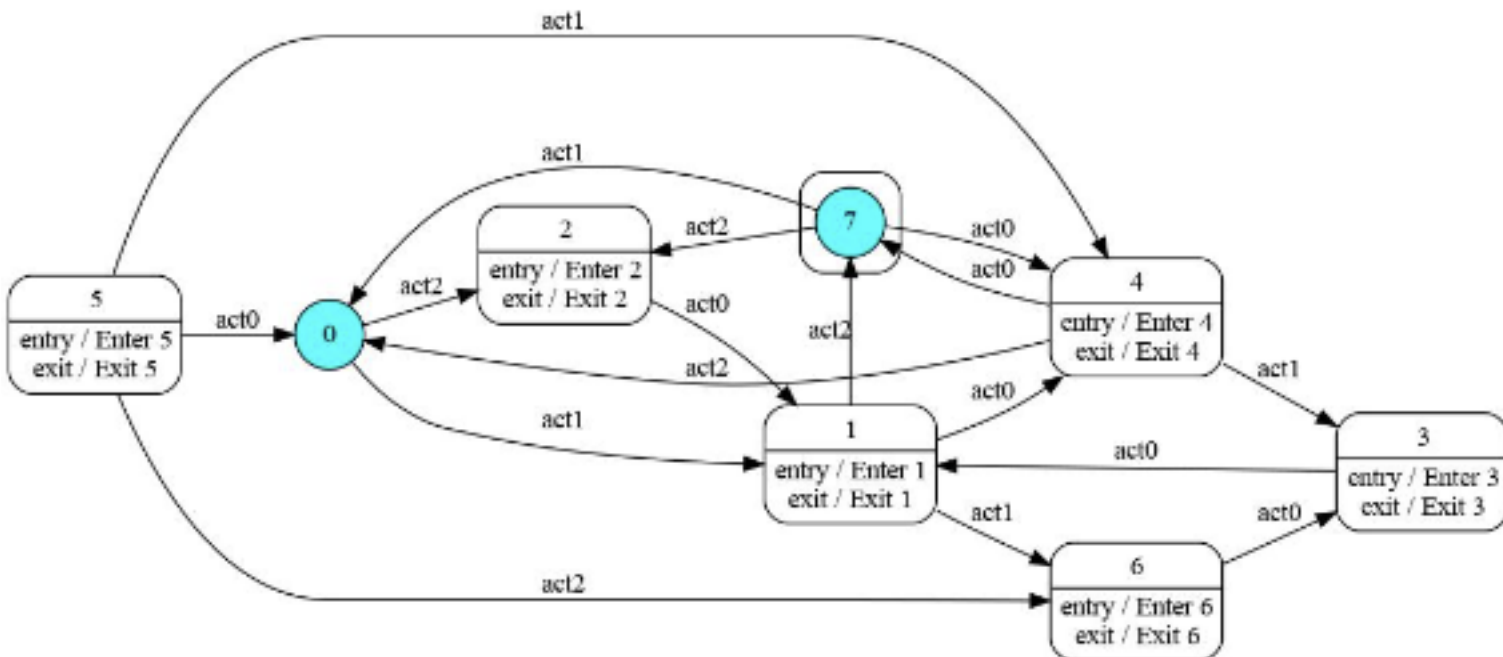
Risposta : 1

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

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

Uno scenario è una sequenza di stati.

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



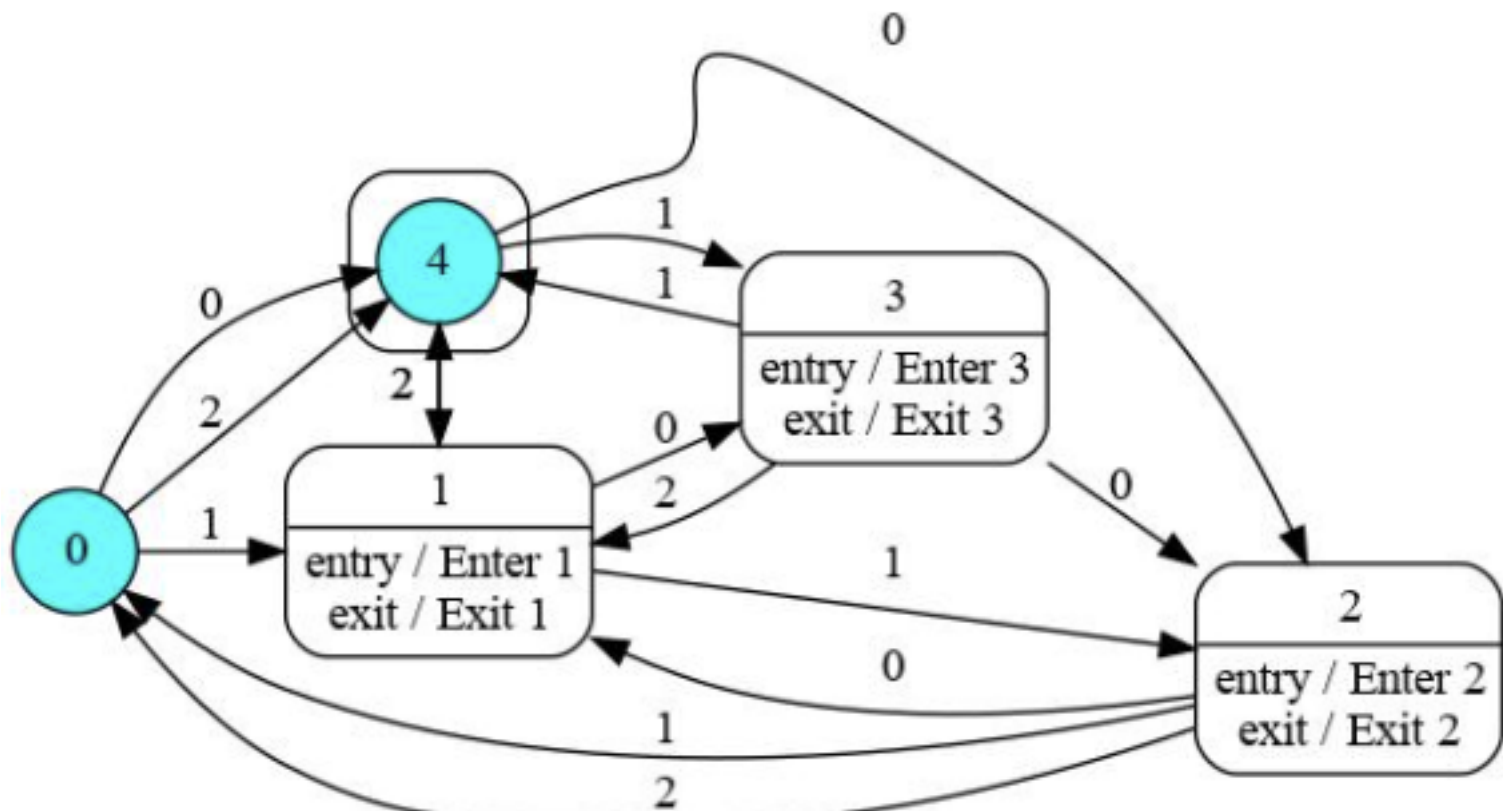
1.0.27

2.0.07

3.0.03

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

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

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

```

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.

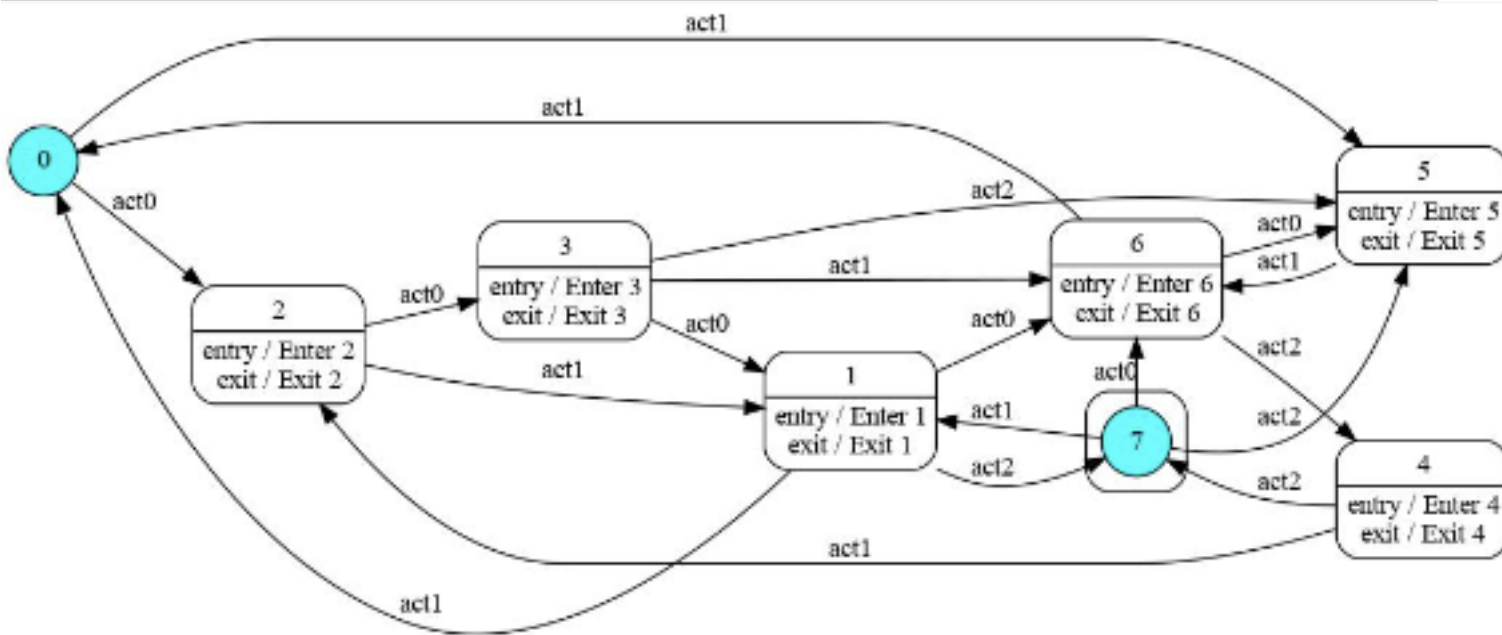
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

Test case 1: act1 act2 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



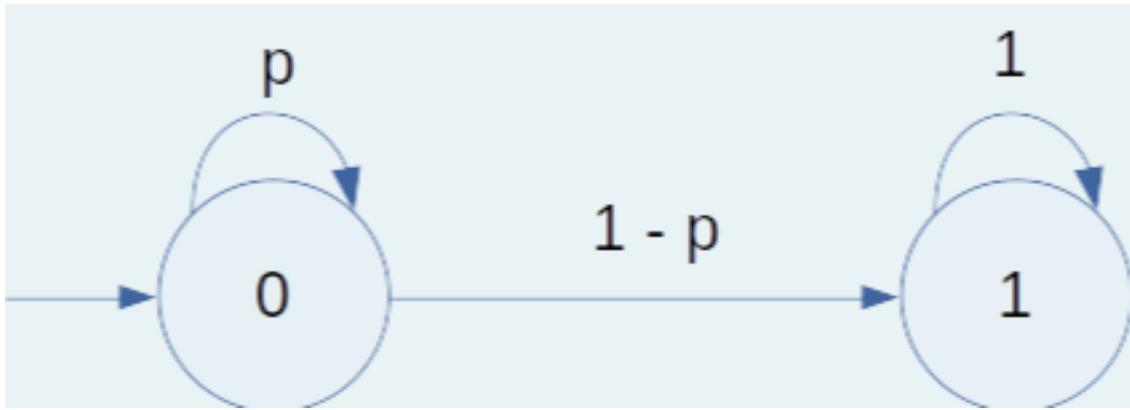
1.State coverage: 70%

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

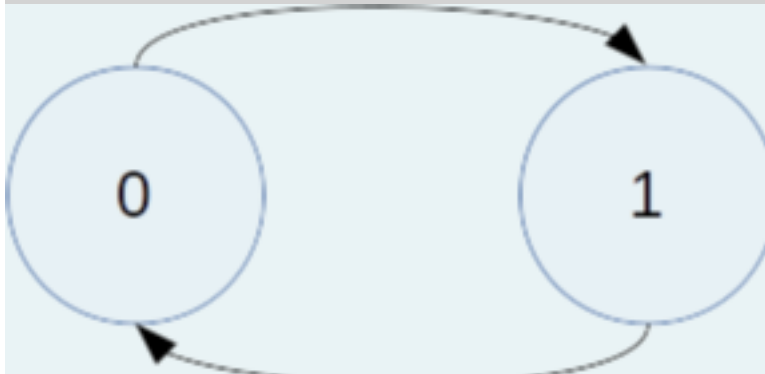
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: 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

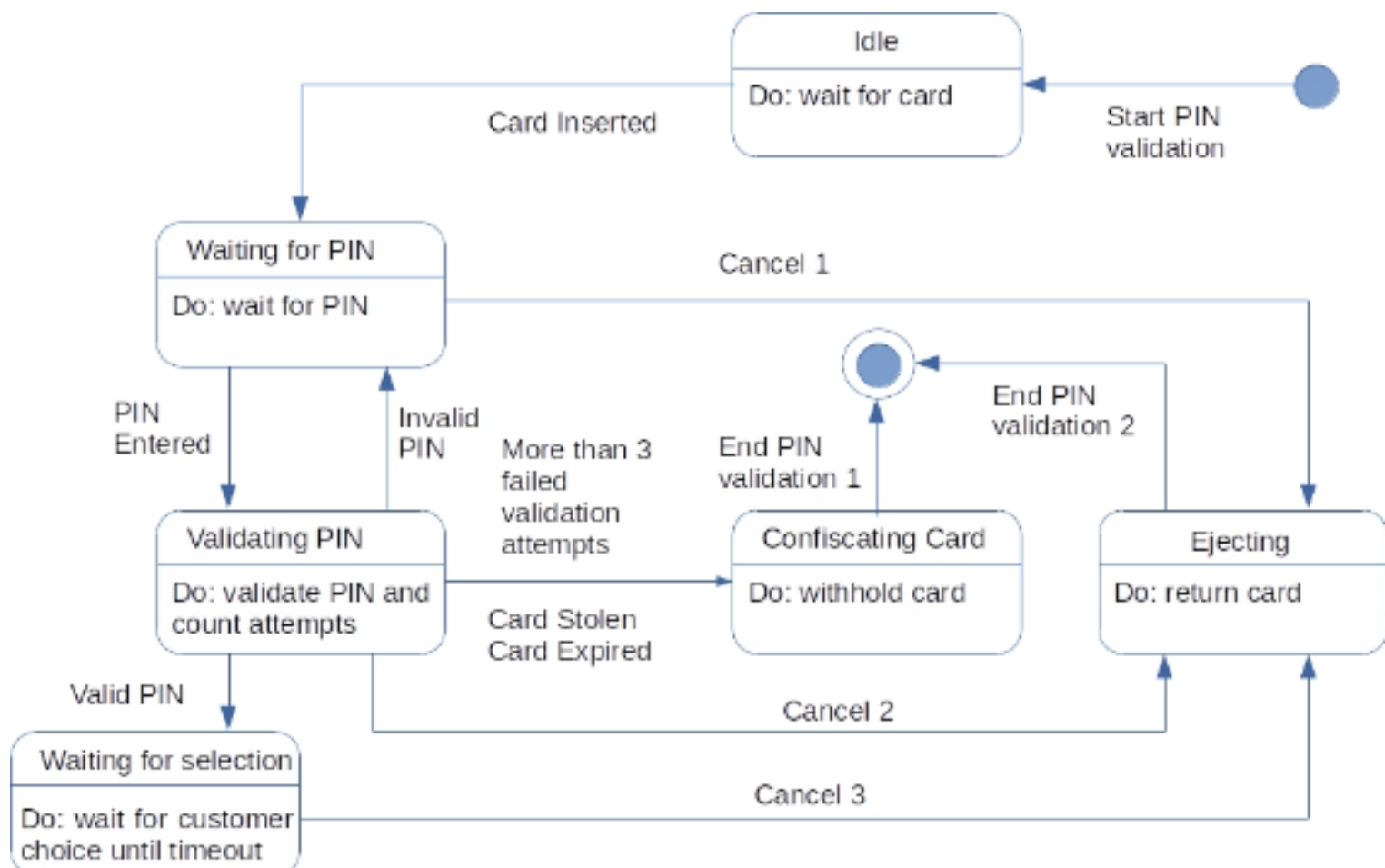
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il tempo necessario per completare la fase x è $\text{time}(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase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

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

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

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

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



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

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

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

Risposta : 3

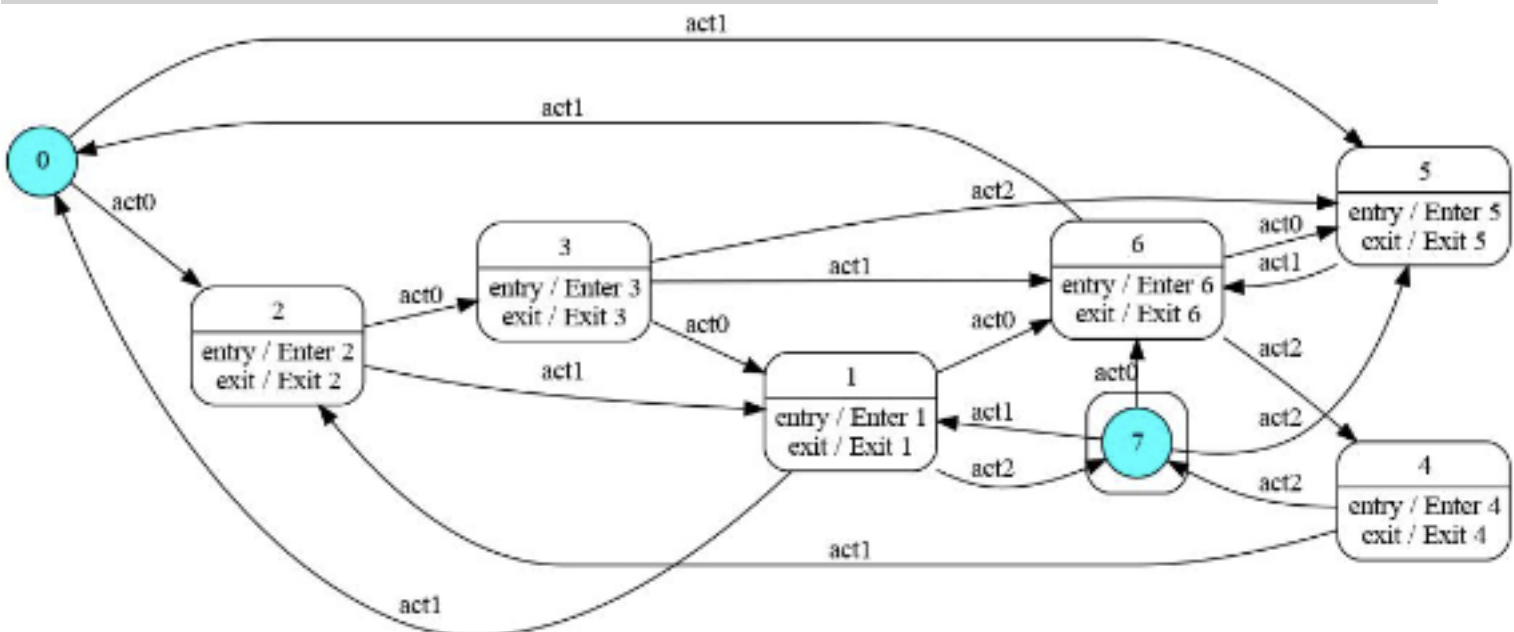
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il costo dello stato (fase) x è $c(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappresenta il completamento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

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

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

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

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



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

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

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

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.

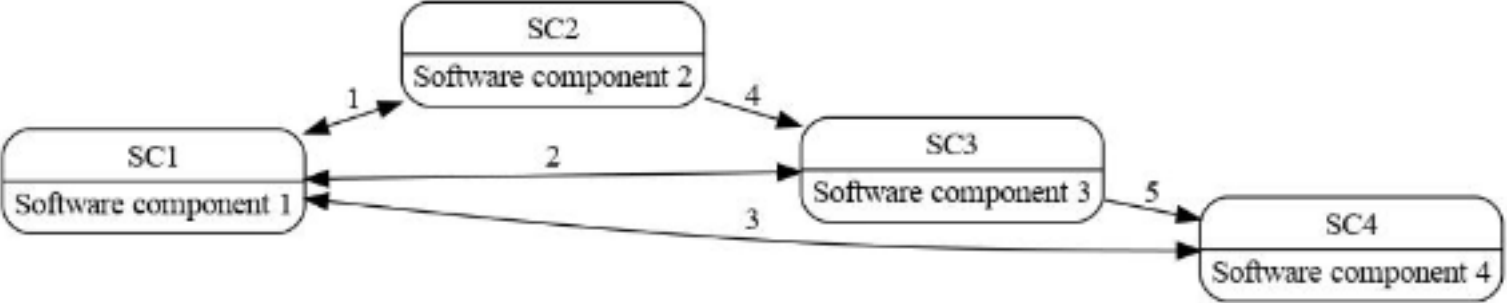
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

Test case 1: 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%

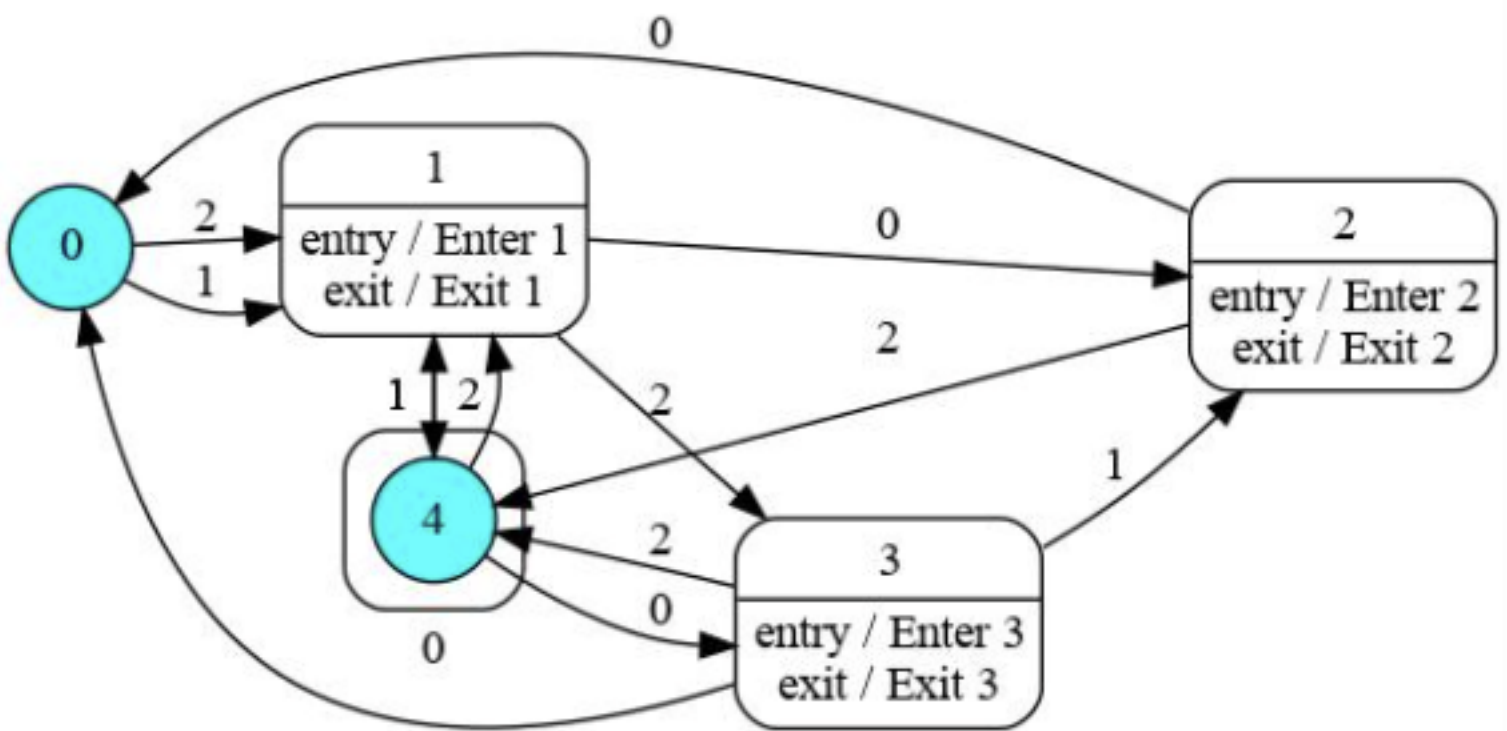
Risposta : 3

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

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

Uno scenario è una sequenza di stati.

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



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.

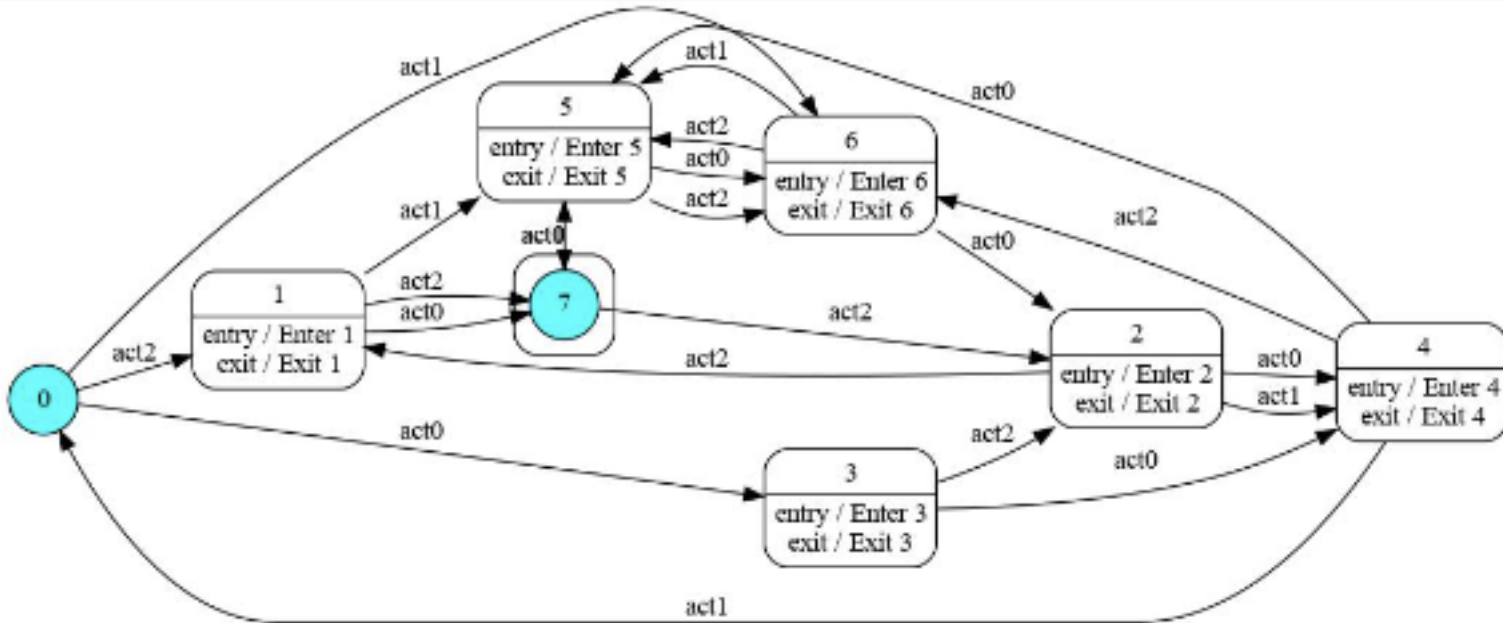
Si consideri lo state diagram in figura Si consideri il seguente insieme di test cases:

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



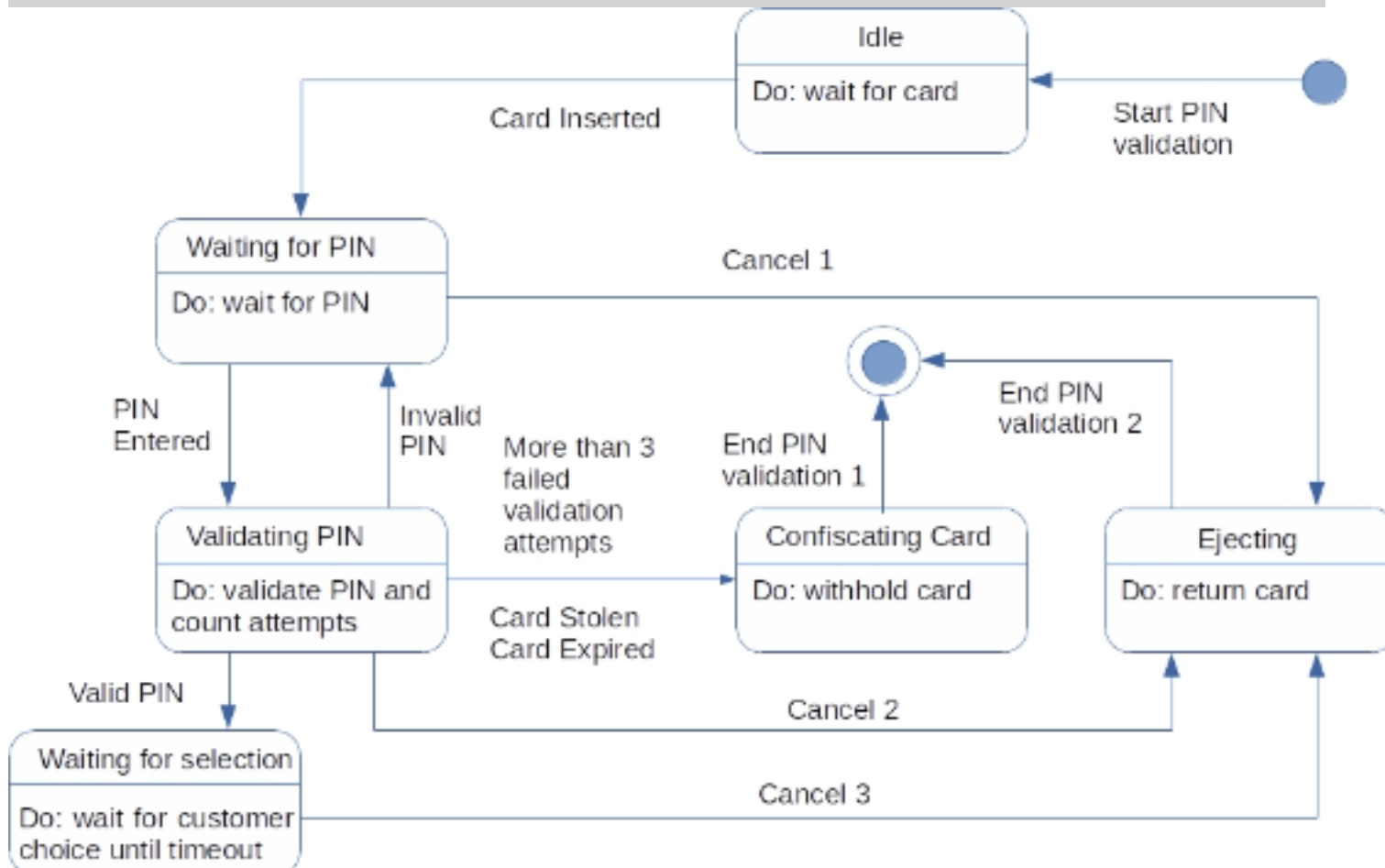
1.State coverage: 80%

2.State coverage: 100%

3.State coverage: 50%

Risposta : 2

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



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 :=

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

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

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

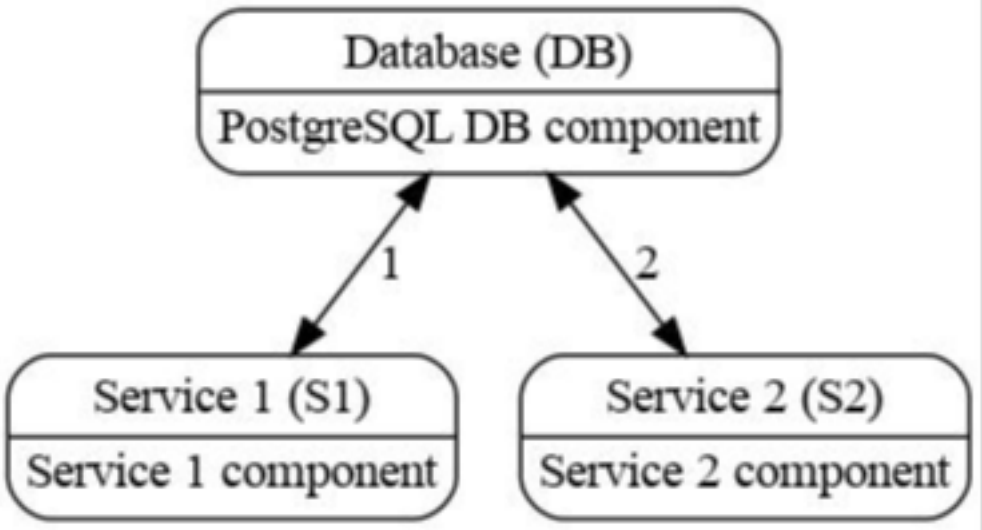
Risposta : 2

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

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

Uno scenario è una sequenza di stati.

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



- 1.0.32
- 2.0.12
- 3.0.08

Risposta : 2

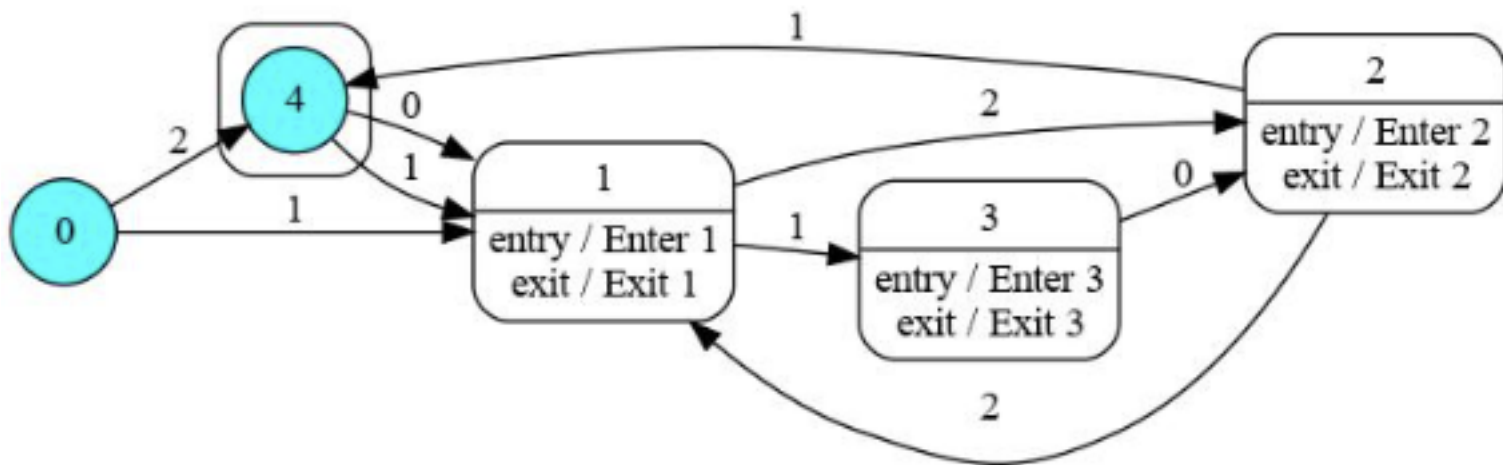
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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%

Risposta : 3

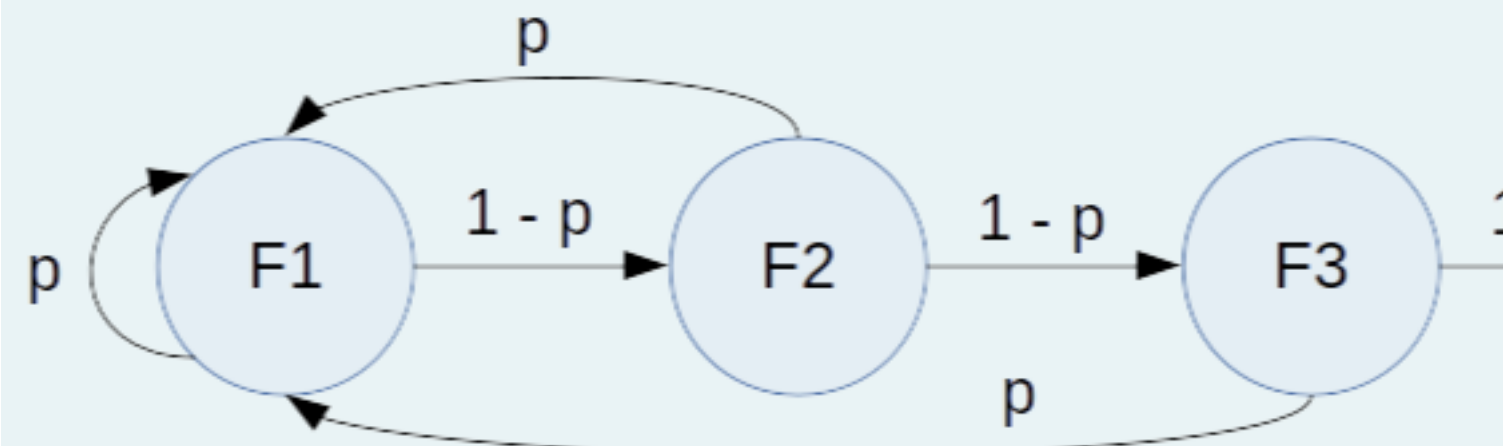
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il costo dello stato (fase) x è $c(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase 1 rappresenta il completamento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

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

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

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

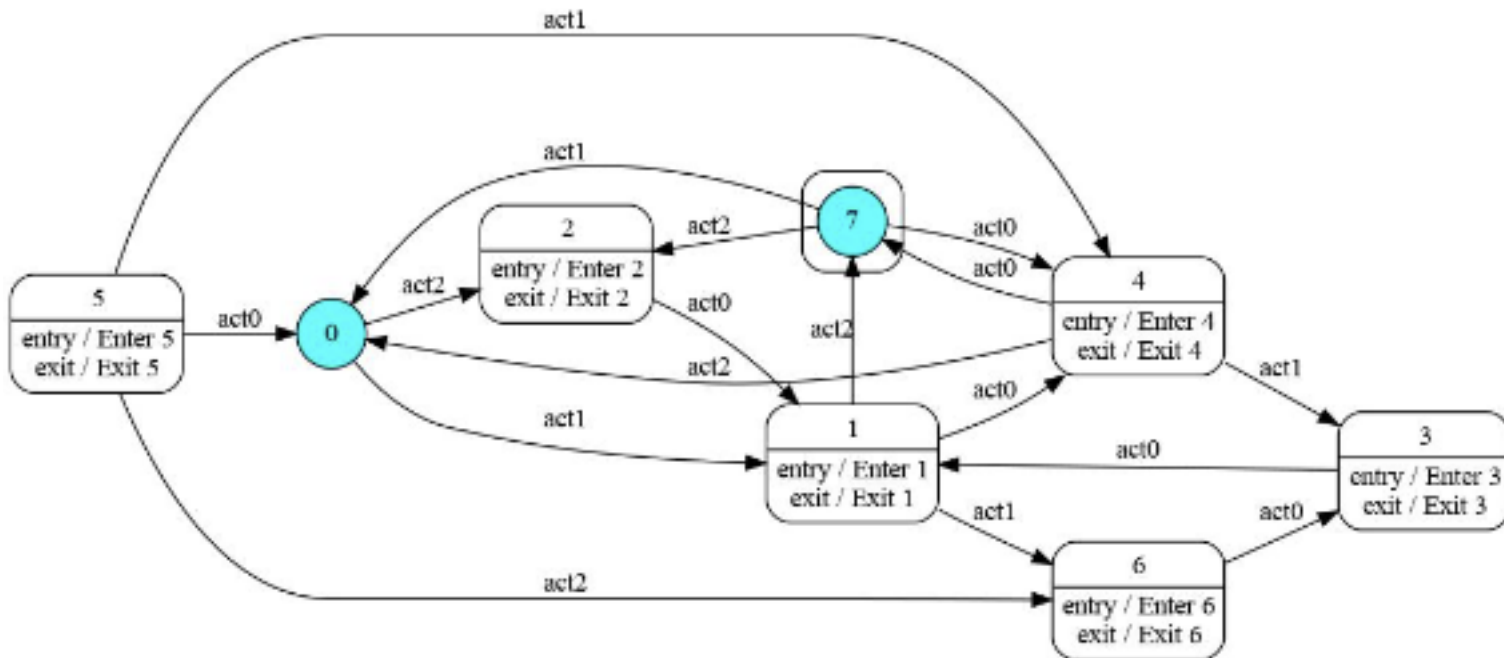
Quale delle seguenti formule calcola il valore atteso del costo per completare il processo software di cui sopra



1. $c(0) \cdot (1 - p) / p$
2. $c(0) / (p \cdot (1 - p))$
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() then x := 0;elsewhen sample(0,T) then if (pre(x) == 0) and (pre(u) == 1) then x := 3;elseif (pre(x) == 0) and (pre(u) == 2) then x := 3;elseif (pre(x) == 1) and (pre(u) == 1) then x := 4;elseif (pre(x) == 2) and (pre(u) == 1) then x := 4;elseif (pre(x) == 3) and (pre(u) == 0) then x := 2;elseif (pre(x) == 3) and (pre(u) == 1) then x := 1;elseif (pre(x) == 3) and (pre(u) == 2) then x := 2;elseif (pre(x) == 4) and (pre(u) == 0) then x := 2;elseif (pre(x) == 4) and (pre(u) == 1) then x := 0;elseif (pre(x) == 4) and (pre(u) == 2) then x := 2;else x := pre(x); // defaultend if;end when;end FSA;

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

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

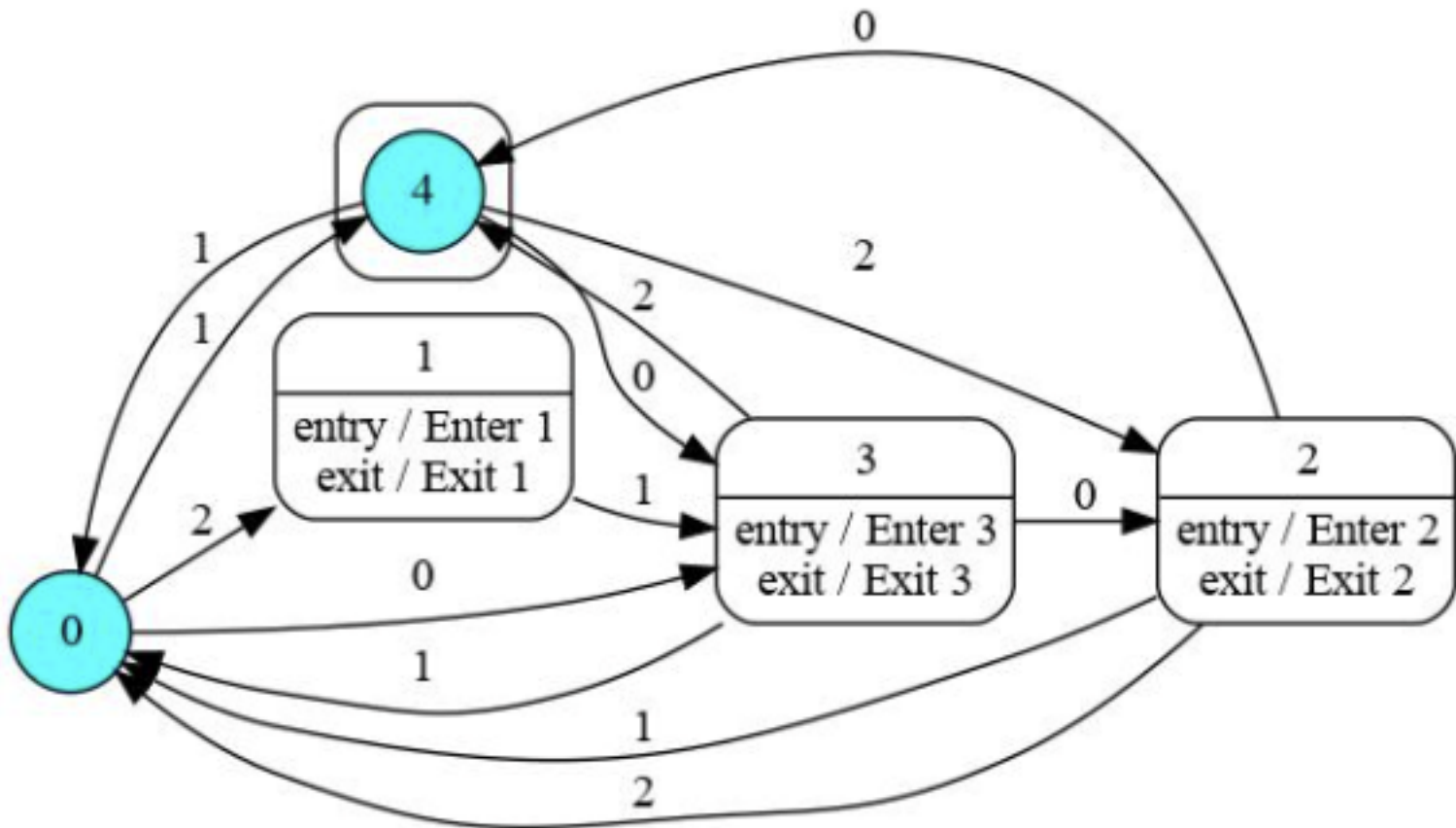
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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

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

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

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



1.State coverage: 87%

2.State coverage: 50%

3.State coverage: 100%

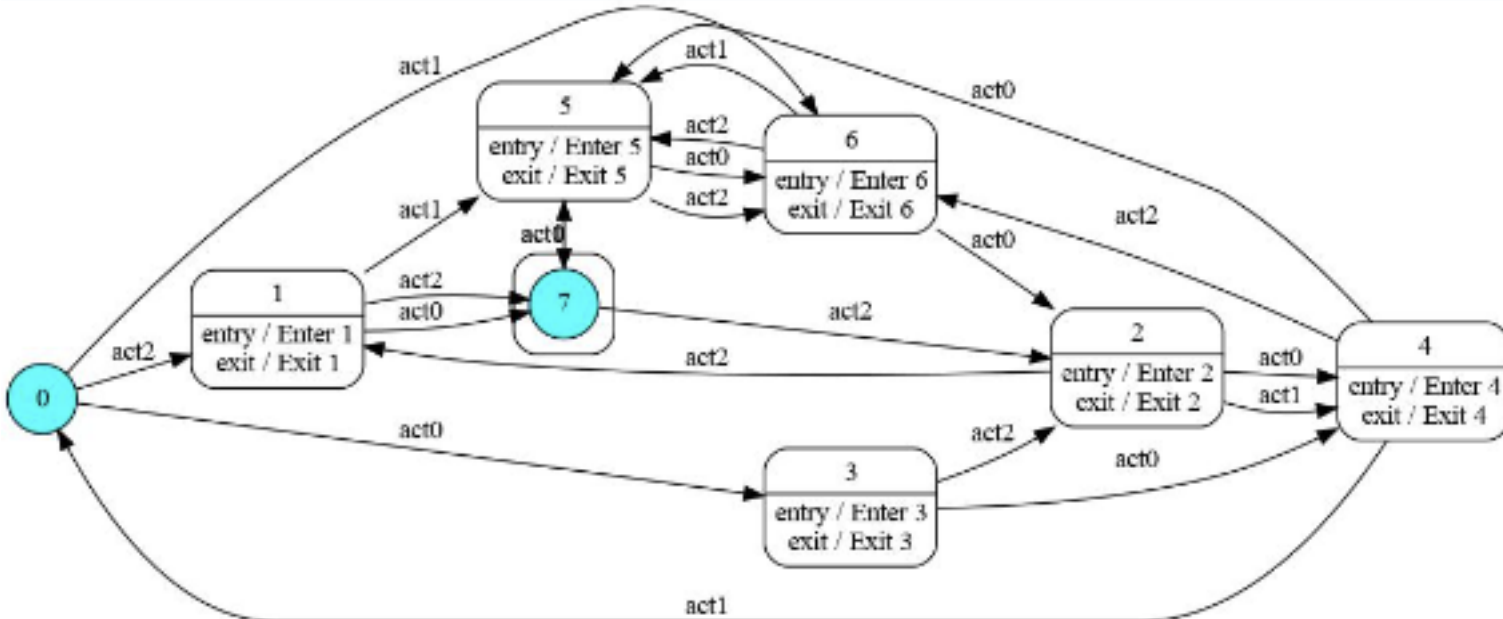
Risposta : 1

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

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

Uno scenario è una sequenza di stati.

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



1.0.03

2.0.27

3.0.07

Risposta : 1

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

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

Uno scenario è una sequenza di stati.

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

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.

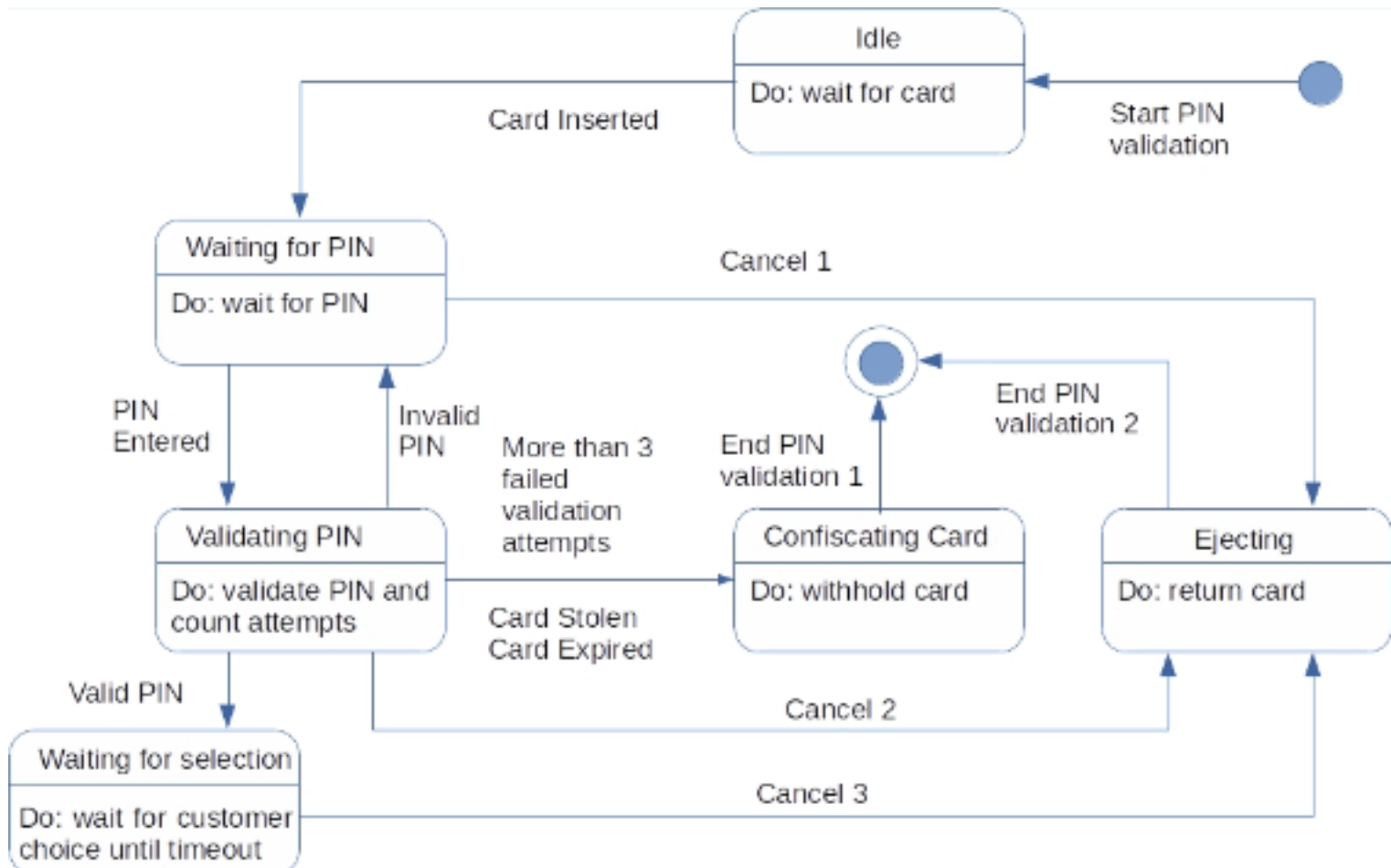
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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?



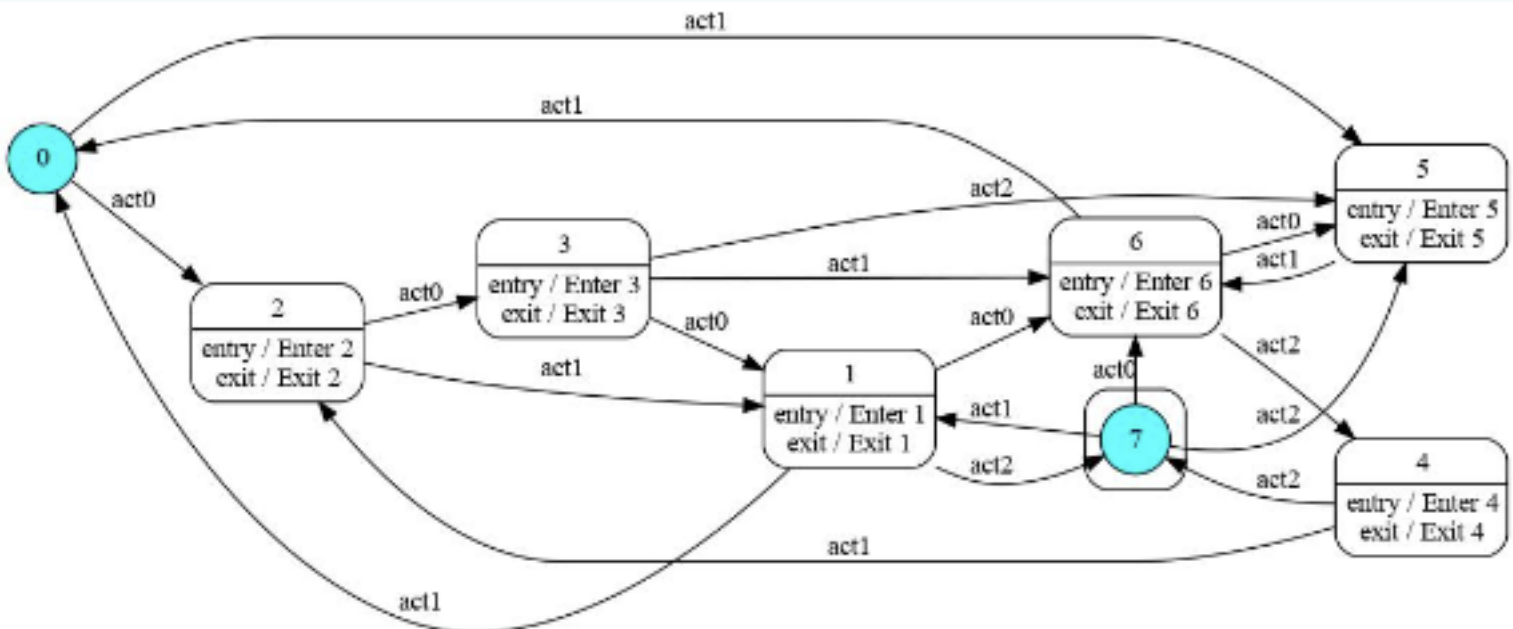
1.Transition coverage: 100%

2.Transition coverage: 70%

3.Transition coverage: 40%

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

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

```

Risposta : 2

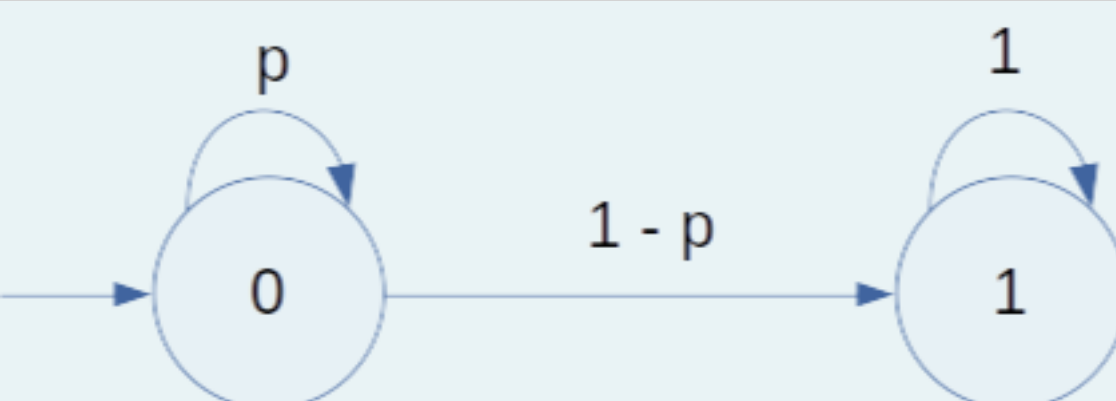
La state coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di stati (inclusi START ed END) raggiunti almeno una volta.

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 act2 act1 act1 act0 act2 act2 act2 act0

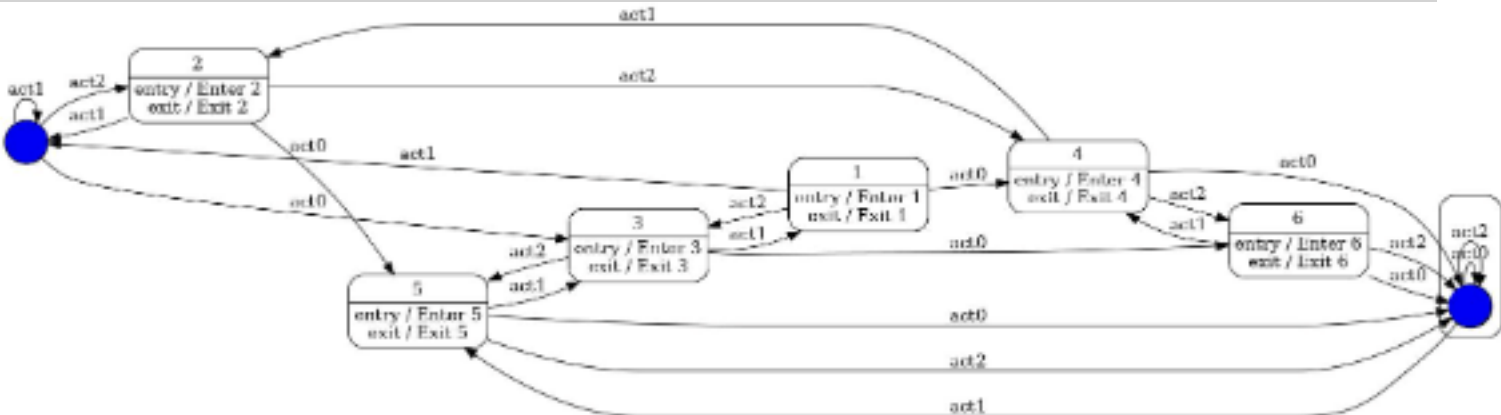
Test case 3: act1 act2 act2 act2 act1 act0 act1 act2 act2 act0 Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



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

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

```

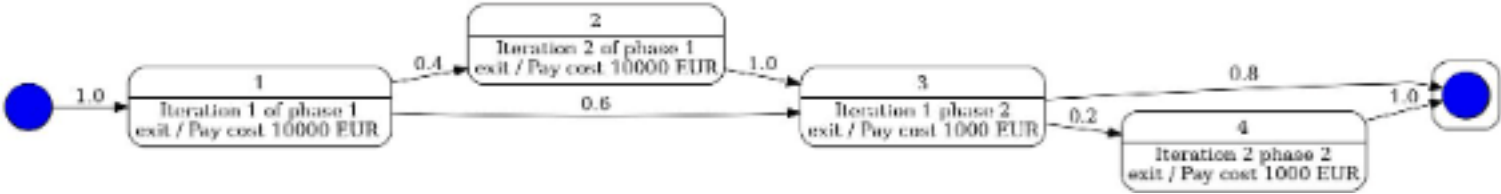
```
defaultend if;end when;end FSA;

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

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



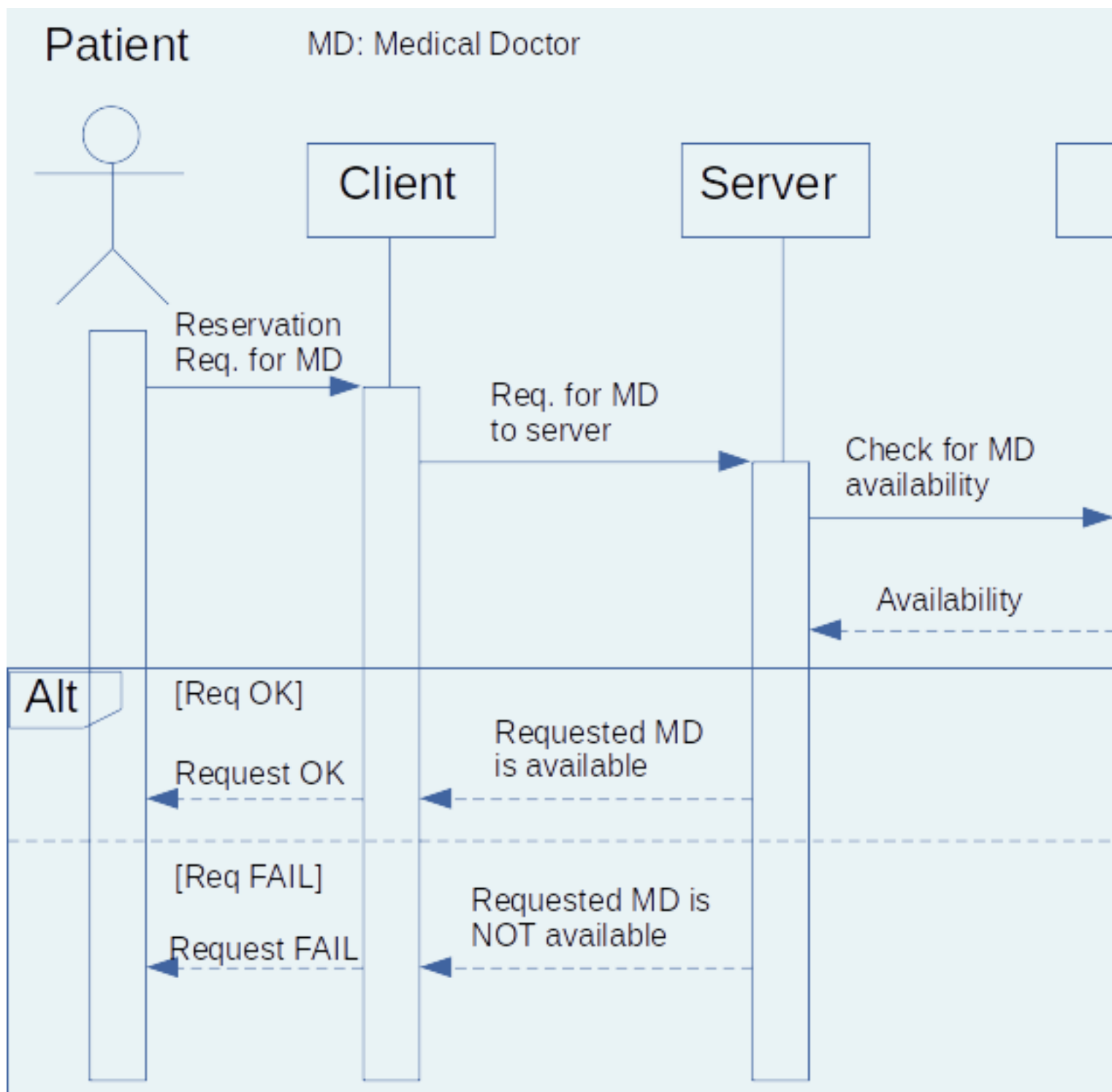
- 1.State coverage: 75%
- 2.State coverage: 50%
- 3.State coverage: 100%

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 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%

Risposta : 2

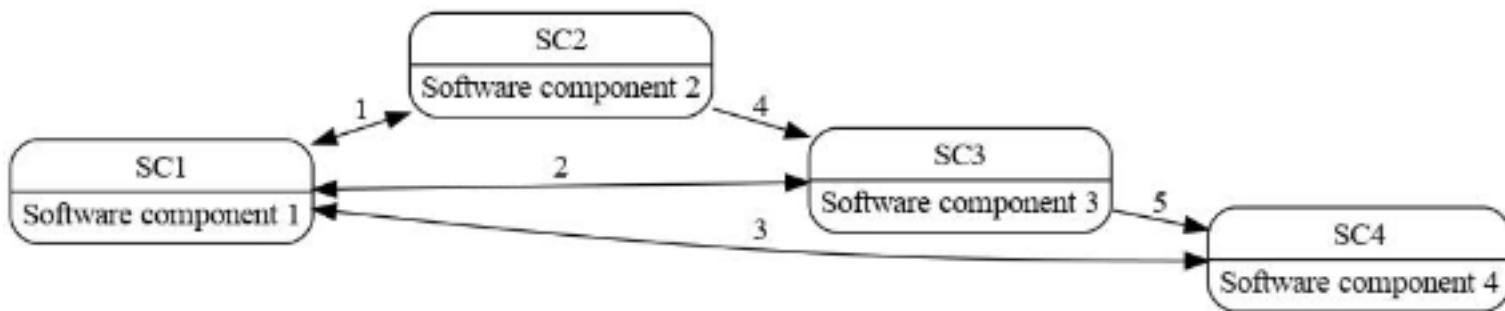
Si consideri il processo software con due fasi (0 ed 1) rappresentato con la Markov chain in figura. Lo stato iniziale 0 e p è in $(0, 1)$. Il tempo necessario per completare la fase x è $\text{time}(x)$. La fase 0 è la fase di design, che ha probabilità p di dover essere ripetuta causa errori. La fase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

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

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

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

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



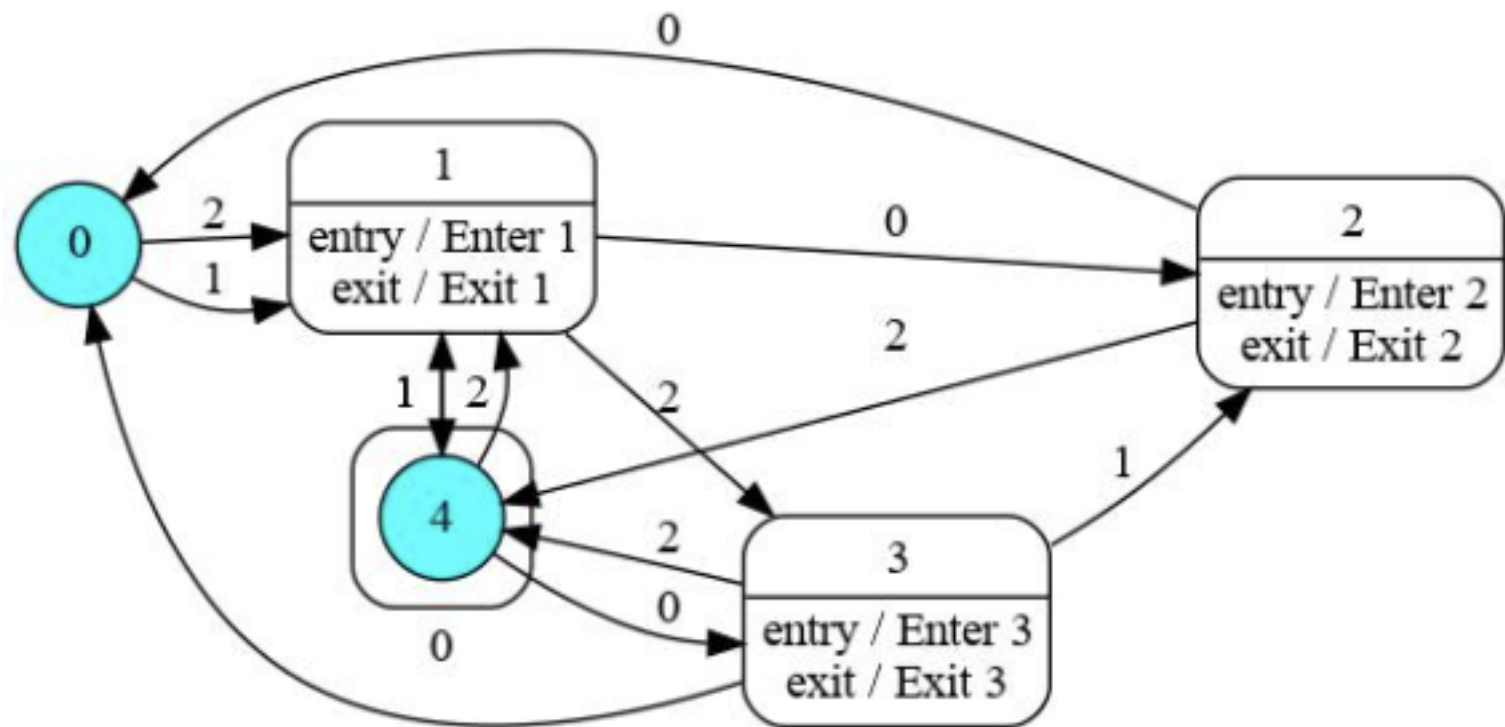
1. $\text{time}(0)/(p \cdot (1 - p))$

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

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

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. $1/(p \cdot (1 - p))$

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.

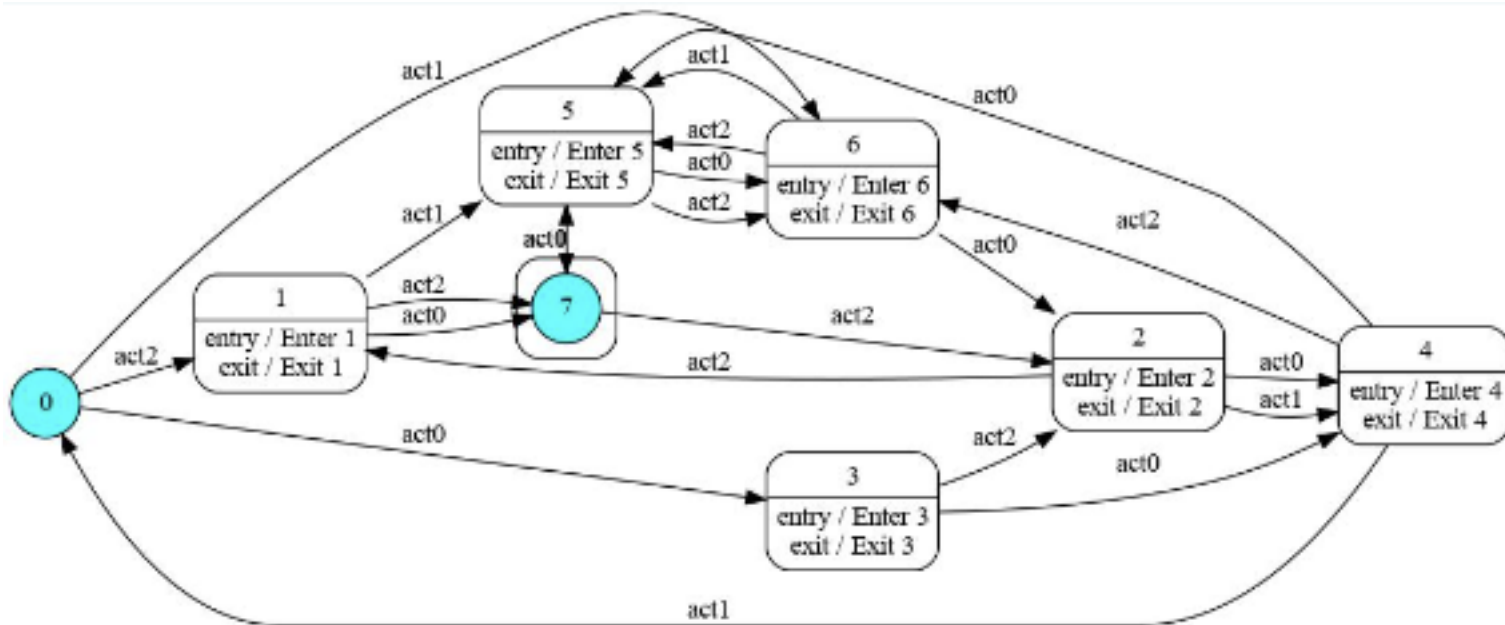
Si consideri lo state diagram in figura ed il seguente insieme di test cases:

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

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

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%

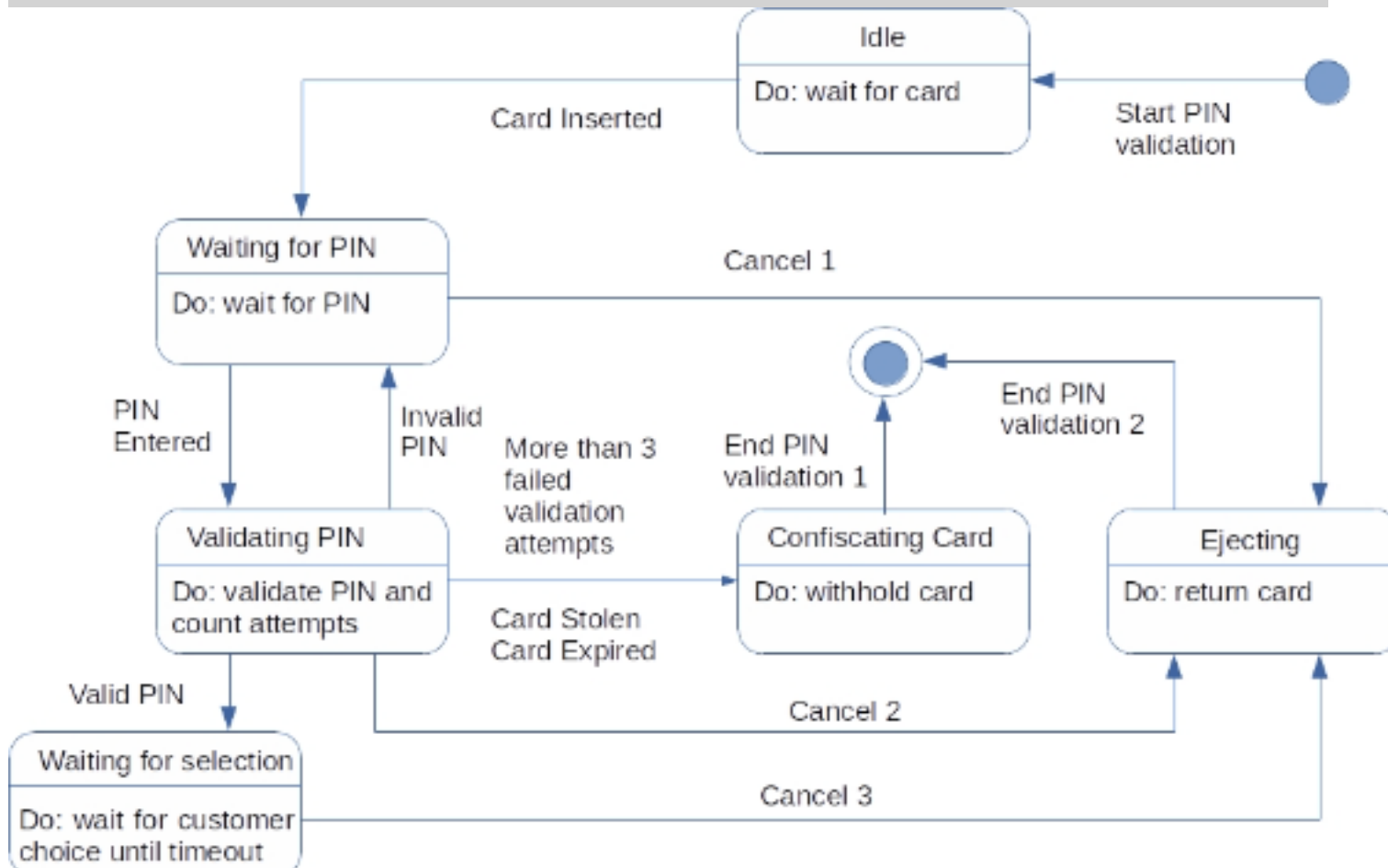
Risposta : 2

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

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

Uno scenario è una sequenza di stati.

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

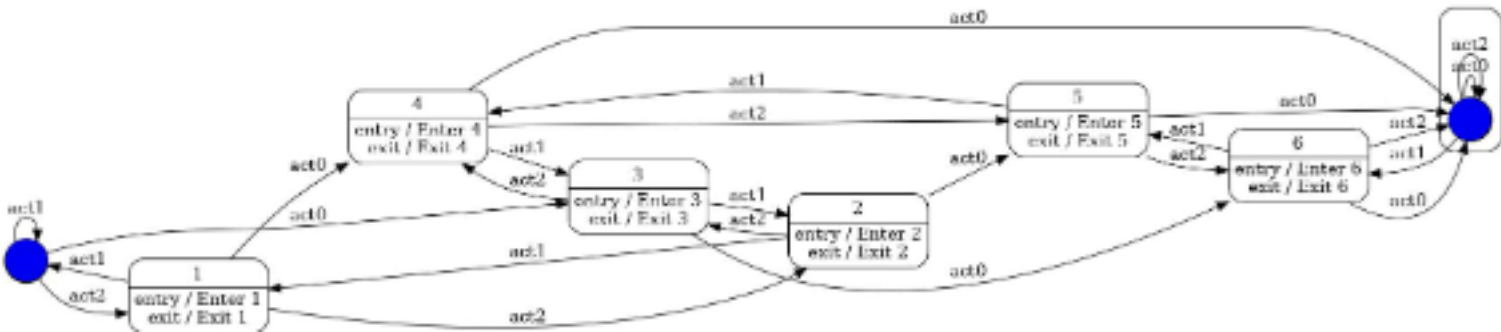


2.0.42

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

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

Risposta : 3

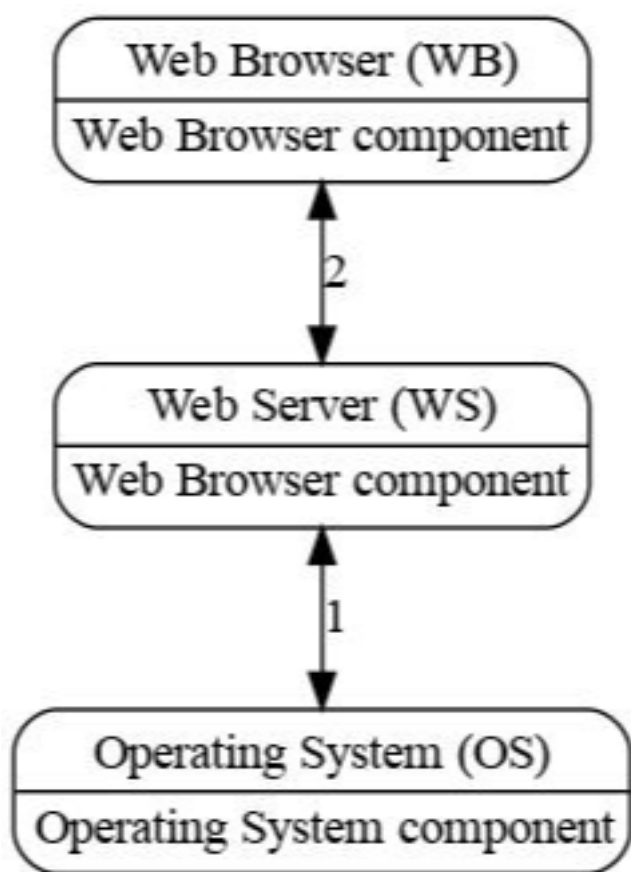
La transition coverage di un insieme di test cases (cioè sequenze di inputs) per uno state diagram è la percentuale di transizioni (archi nel grafo dello state diagram) percorsi almeno una volta.

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

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



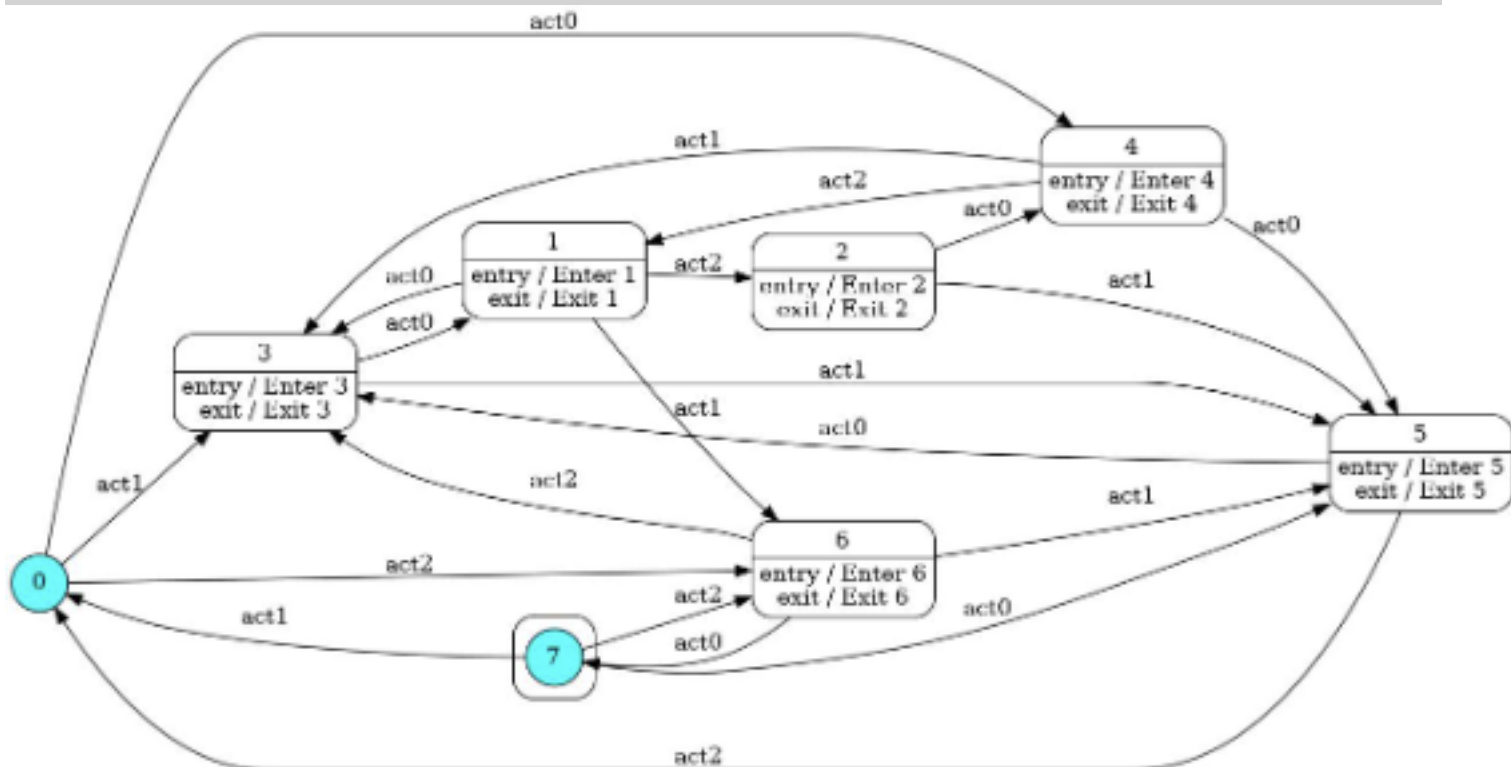
1. Transition coverage: 35%

2. Transition coverage: 90%

3. Transition coverage: 50%

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 ?



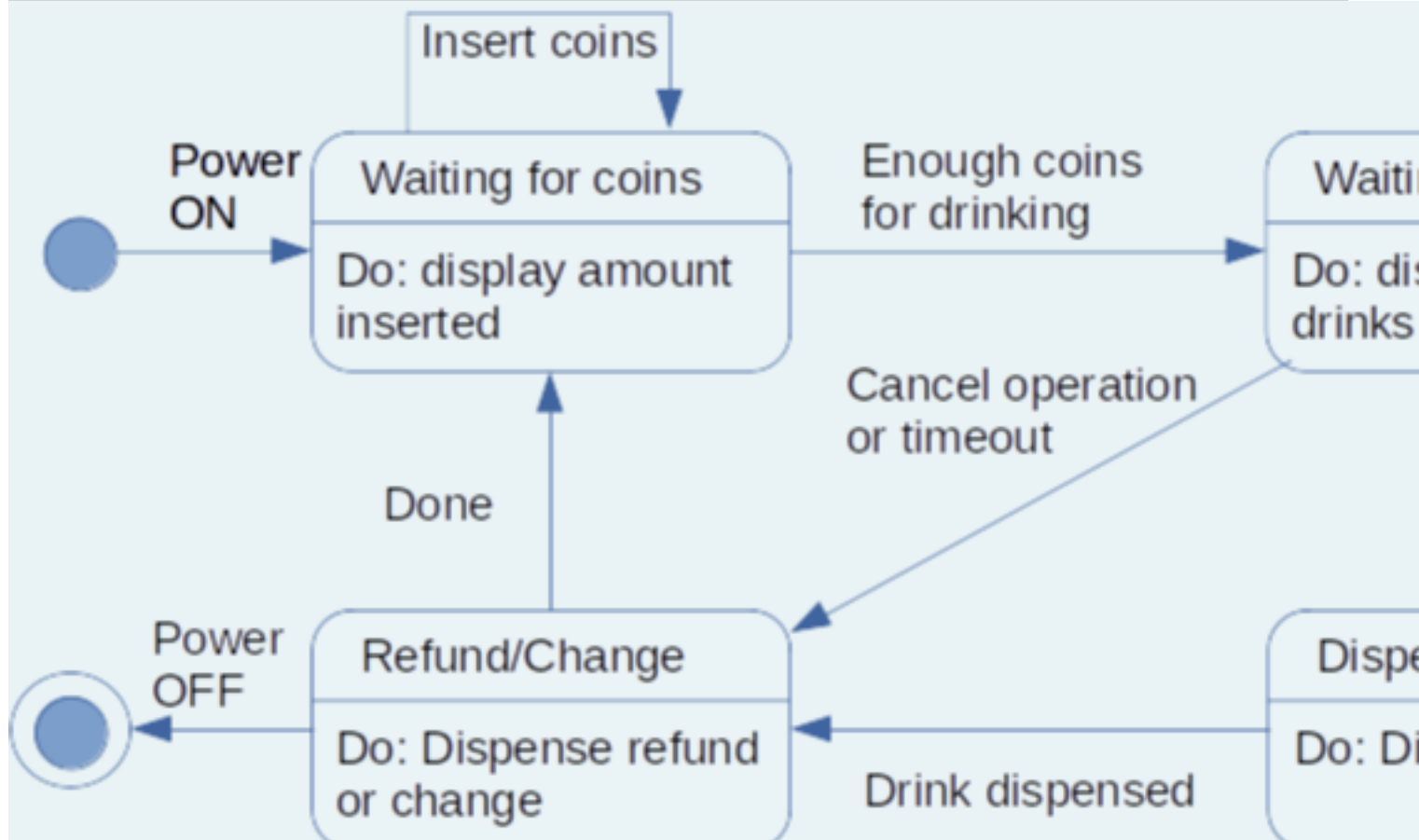
1. Se ci sono abbastanza monete è sempre possibile ottenere la bevanda selezionata.

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.

Risposta : 1

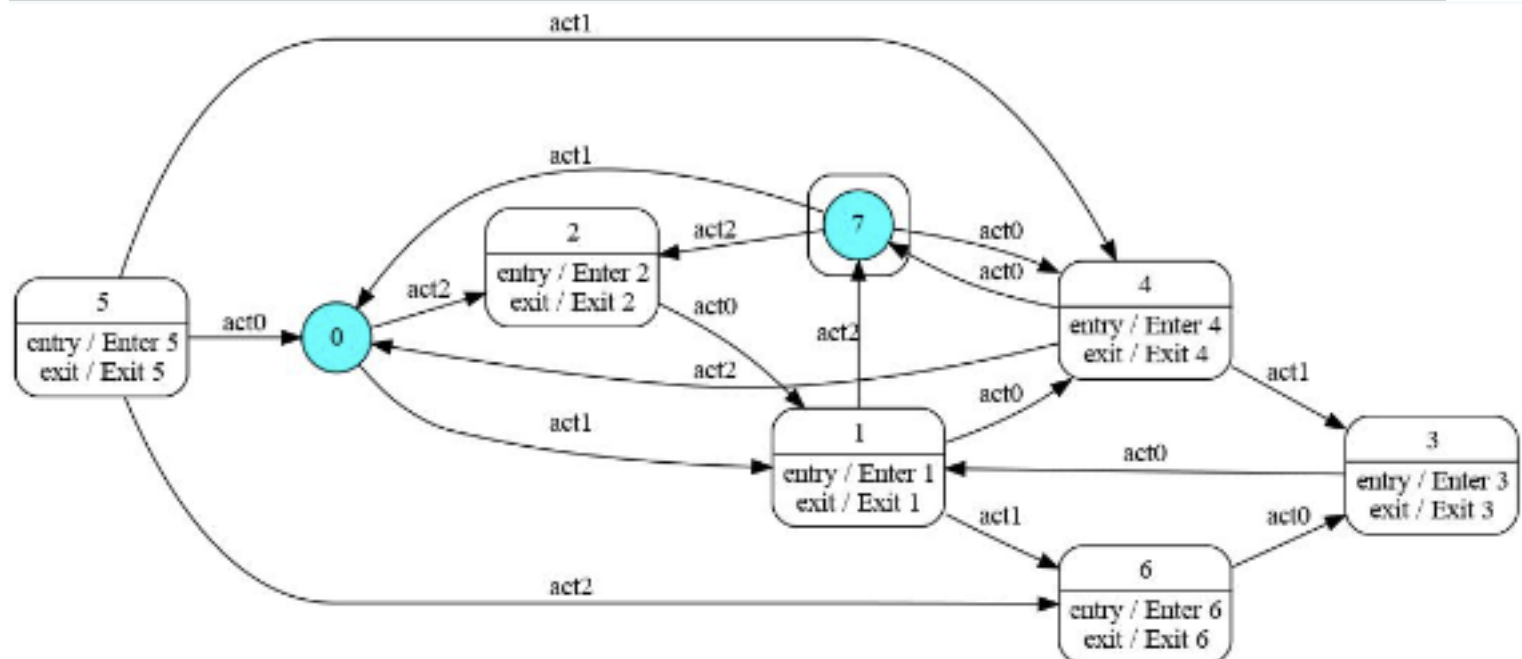
Quale delle seguenti frasi è corretta riguardo all'ctivity diagram in figura ?



1. Una volta selezionato il piatto di mare da preparare, la 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

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 <= 2)

        {return ;}

        else {y = y + 1; goto LOOP;}

} /* f() */

int main() { int i, y; int x[N];

// define test cases

    x[0] = 3;

// testing

    for (i = 0; i < N; i++) {

        y = f(x[i]);    // function under testing

        assert(y == (abs(x[i]) <= 2) ? 0 : (abs(x[i]) - 2)); // oracle

    }

    printf("All %d test cases passed\n", N);

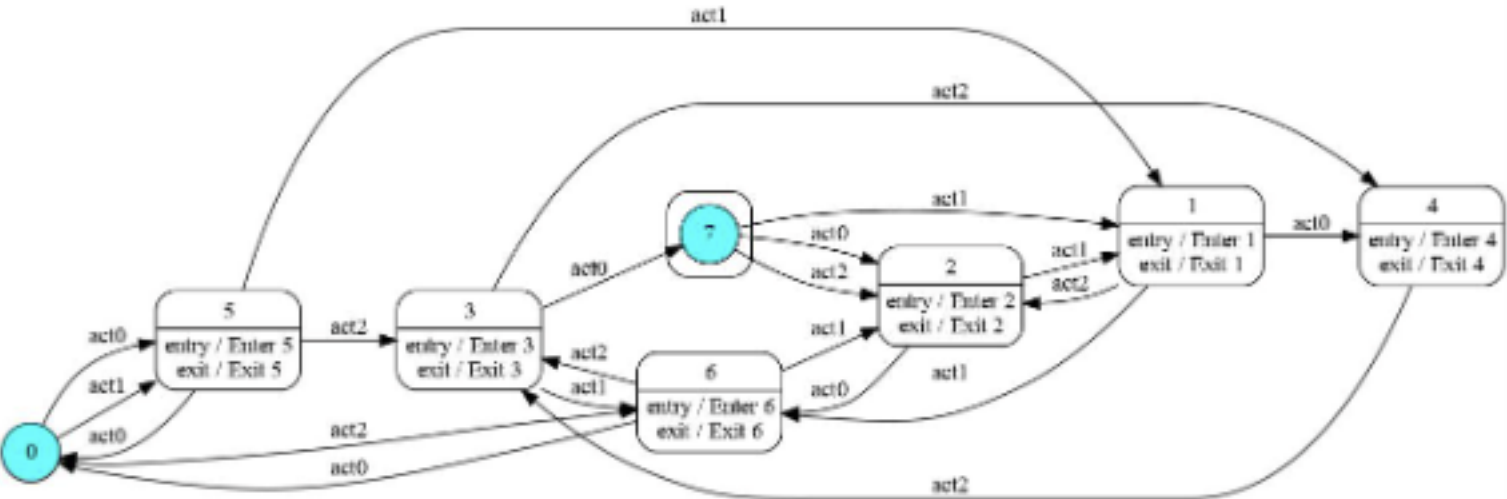
    return (0);

}

-----
```

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

Quale delle seguenti è la branch coverage conseguita?



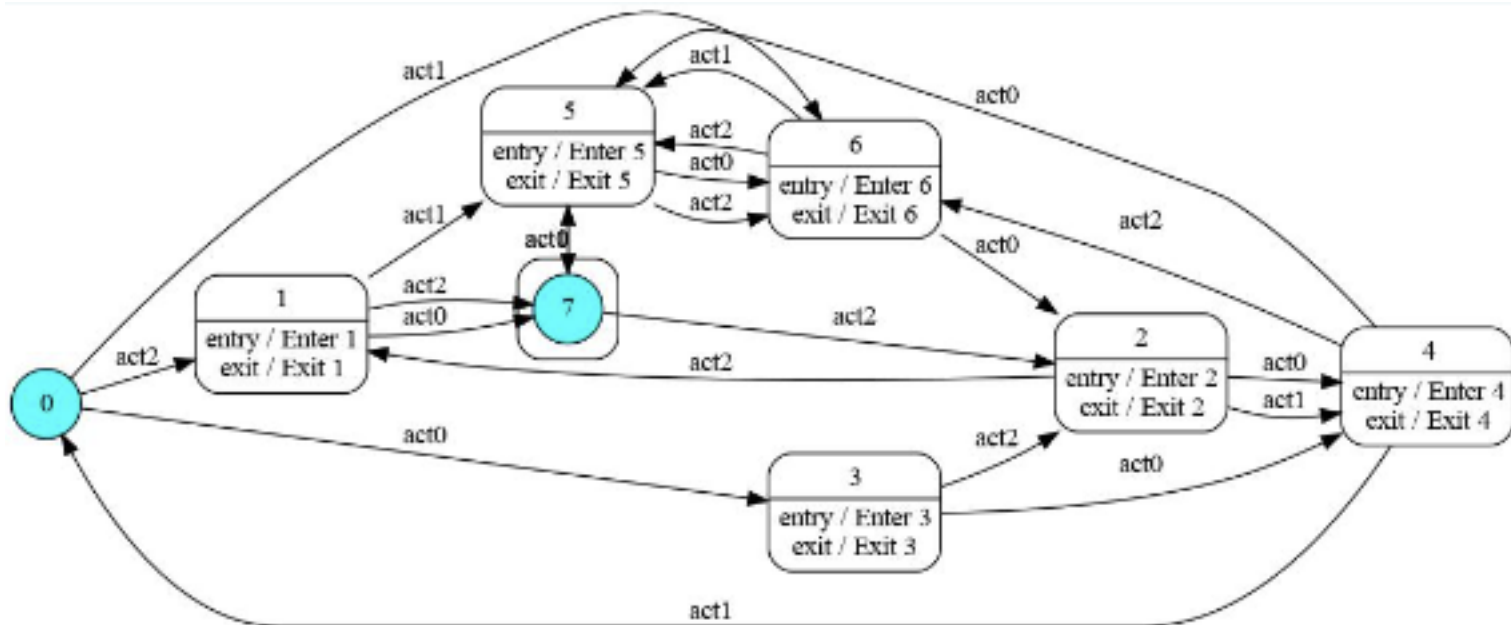
- 1.50%
- 2.100%
- 3.80%

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:

- 1) Start PIN validation, card inserted, PIN Entered, Valid PIN, Cancel 3, End PIN Validation 2
 - 2) Start PIN validation, card inserted, PIN Entered, Cancel 2, End PIN Validation 2
- Quale delle seguenti è la migliore stima della transition coverage per i test cases di cui sopra



1.60%

2.40%

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) rggunti almeno una volta.

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

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

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

Web Browser

Forms and Query Manager

Database

1.90%

2.80%

3.60%

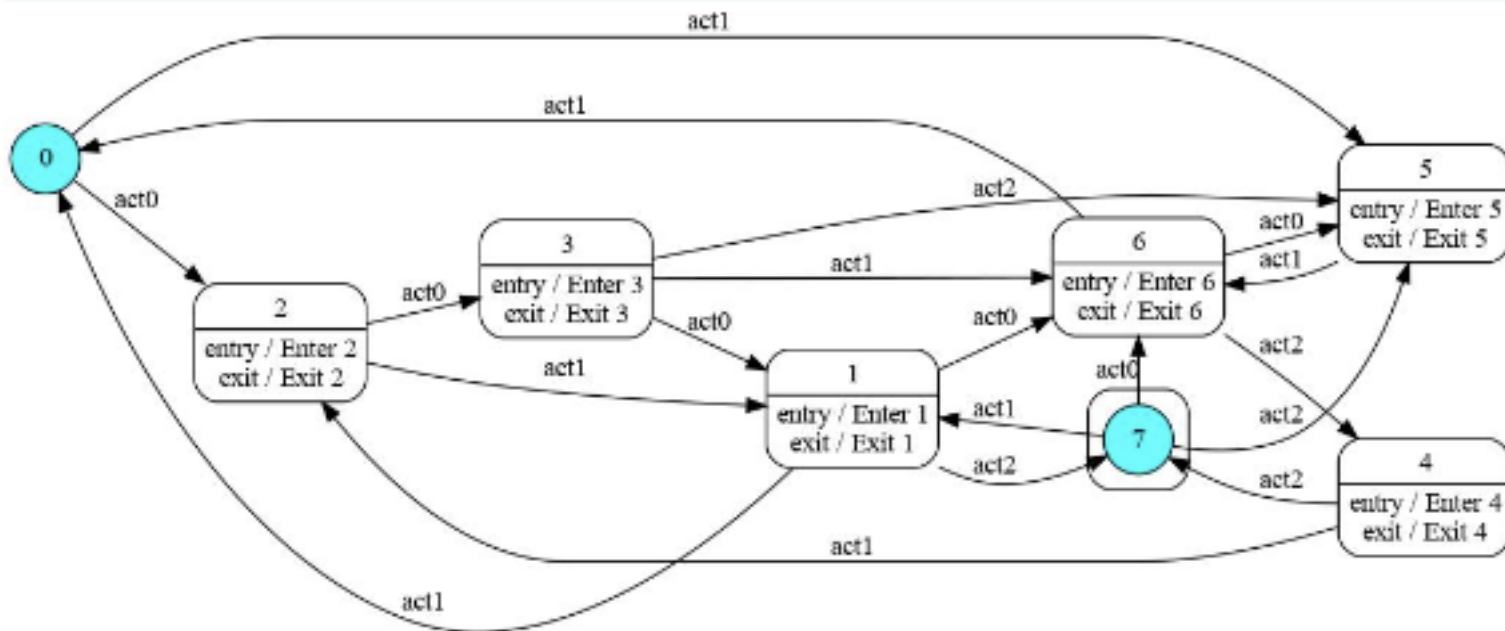
Risposta : 1

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) rggunti 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.

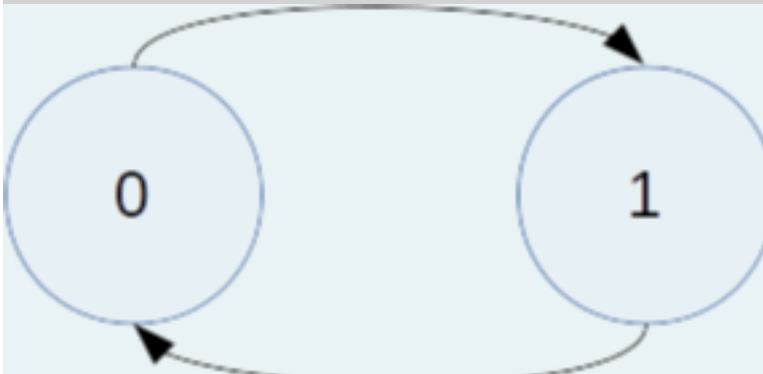
Quale delle seguenti è la migliore stima della state coverage per i test cases di cui sopra



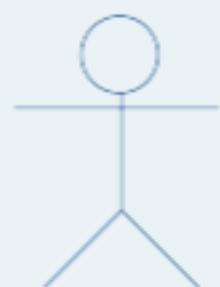
- 1.80%
- 2.60%
- 3.90%

Risposta : 3

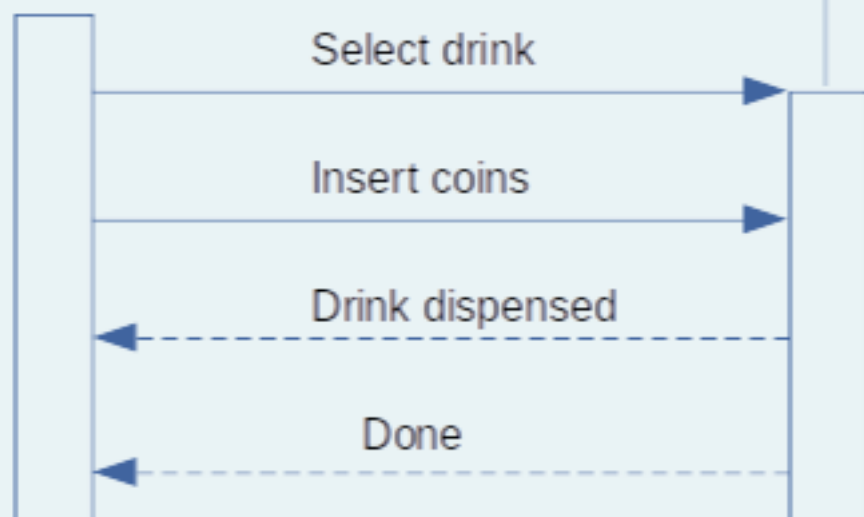
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?



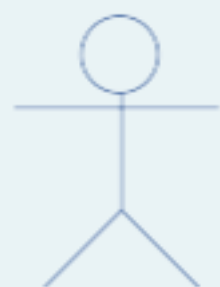
Customer



Vending Machine



Customer



Vending
Machine

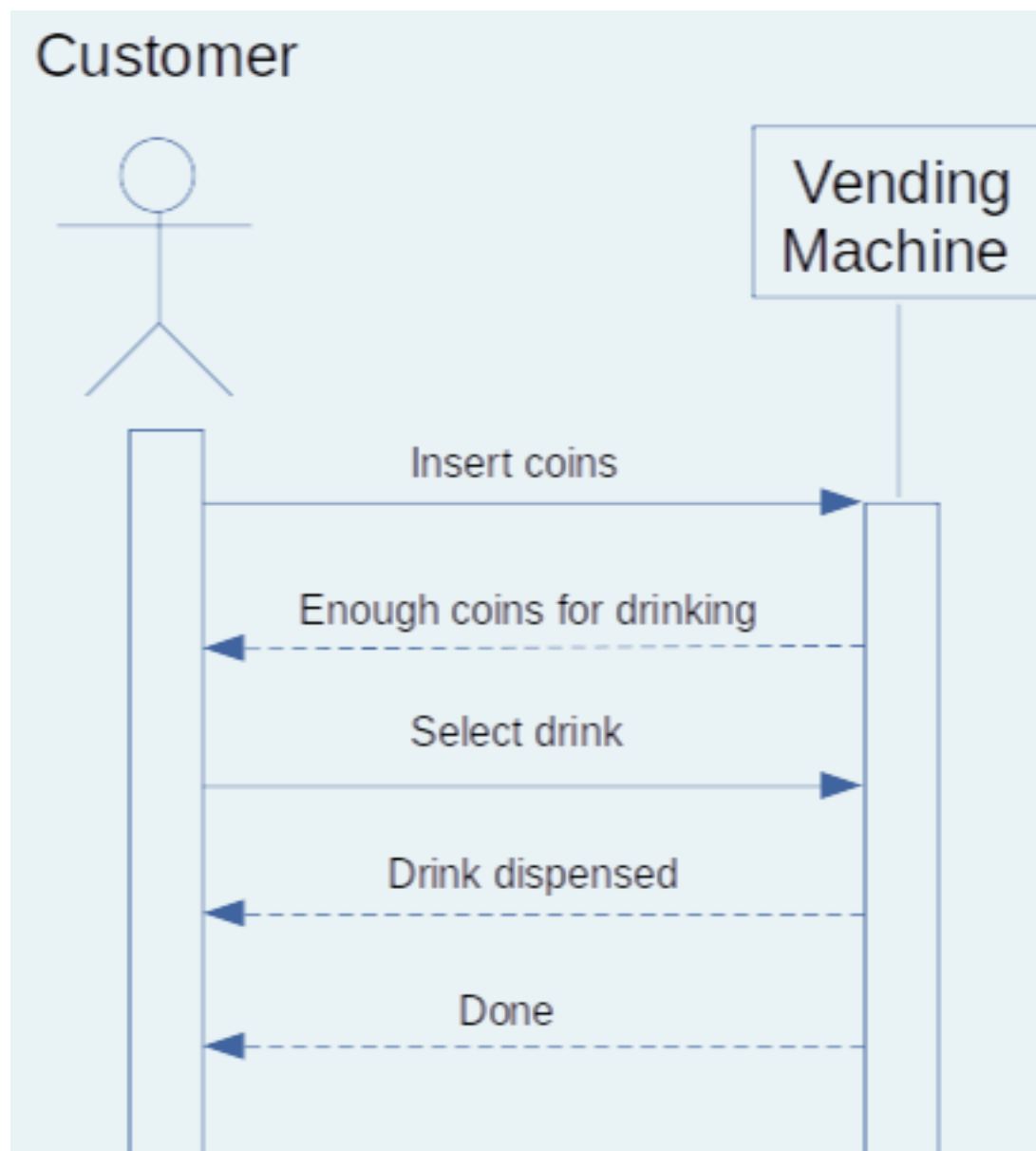
Insert coins

Enough coins for drinking

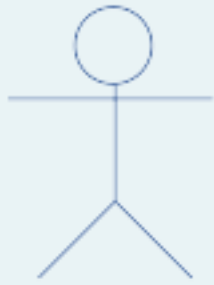
Select drink

Drink dispensed

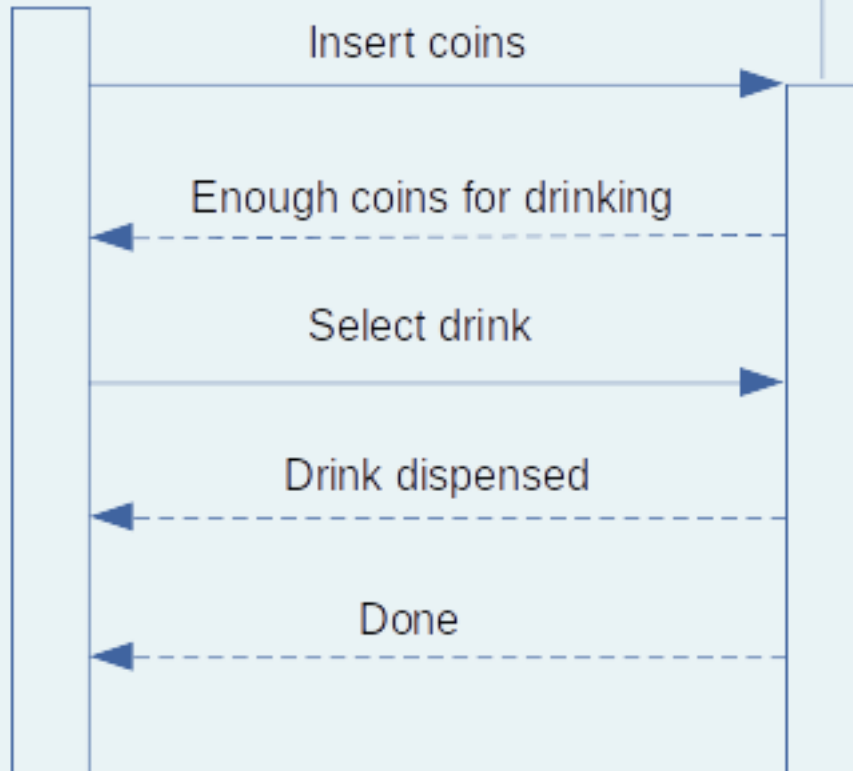
Done



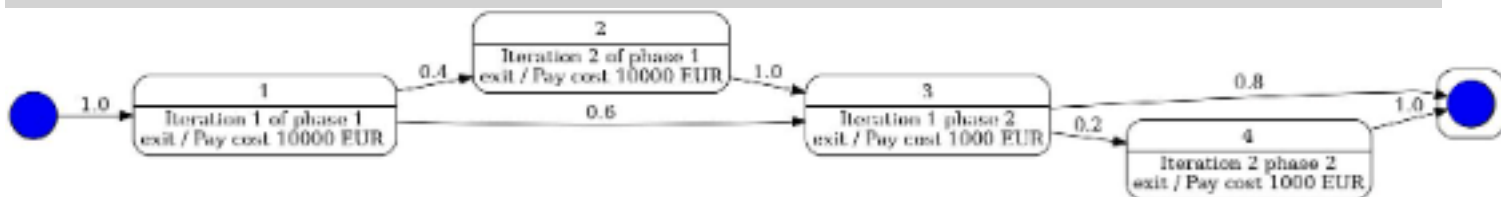
Customer



Vending Machine



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.

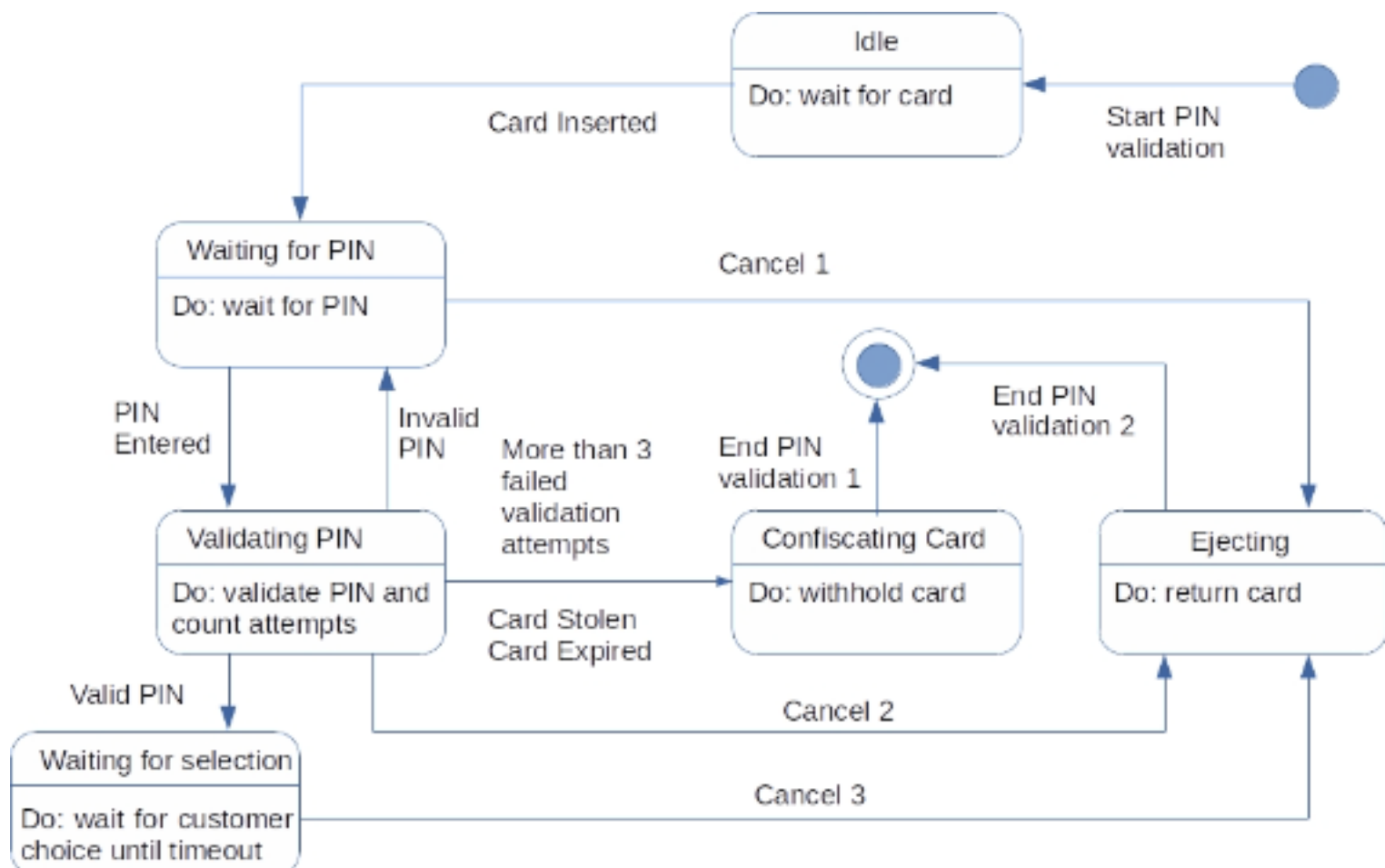
Risposta : 3

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 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%

2.100%

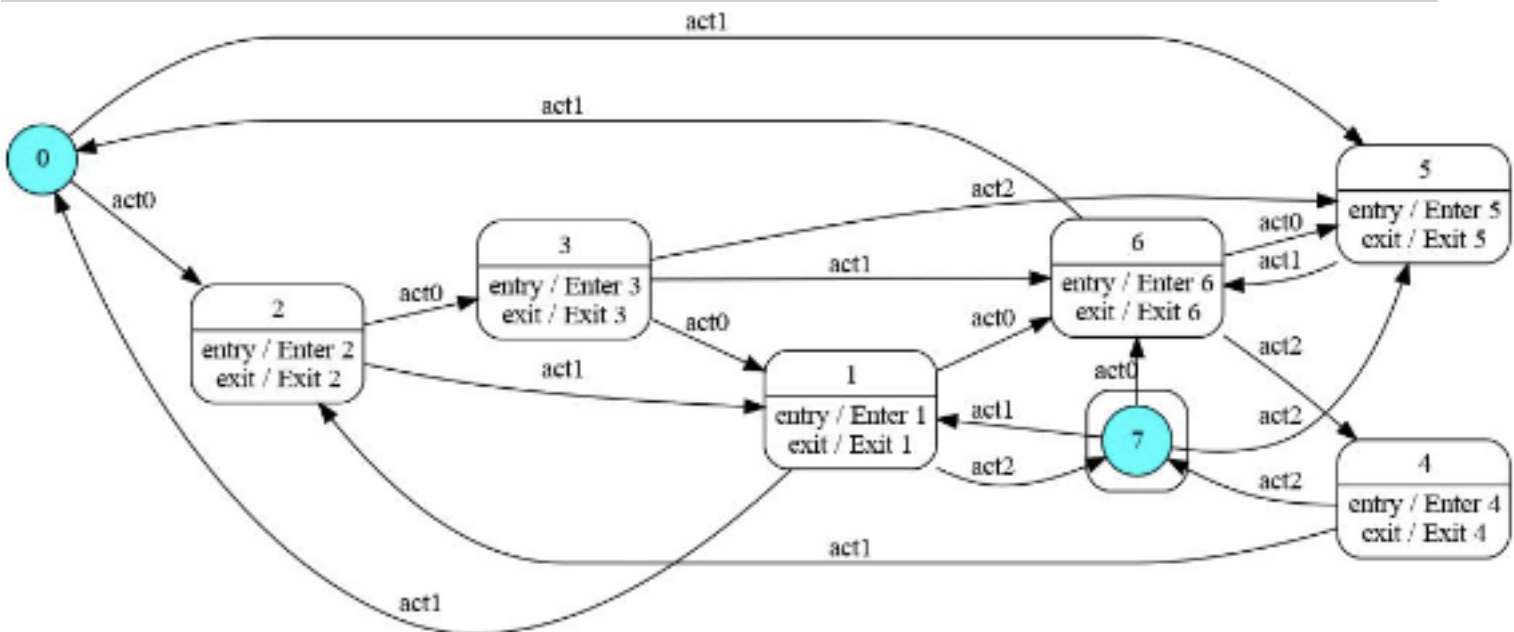
3.90%

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

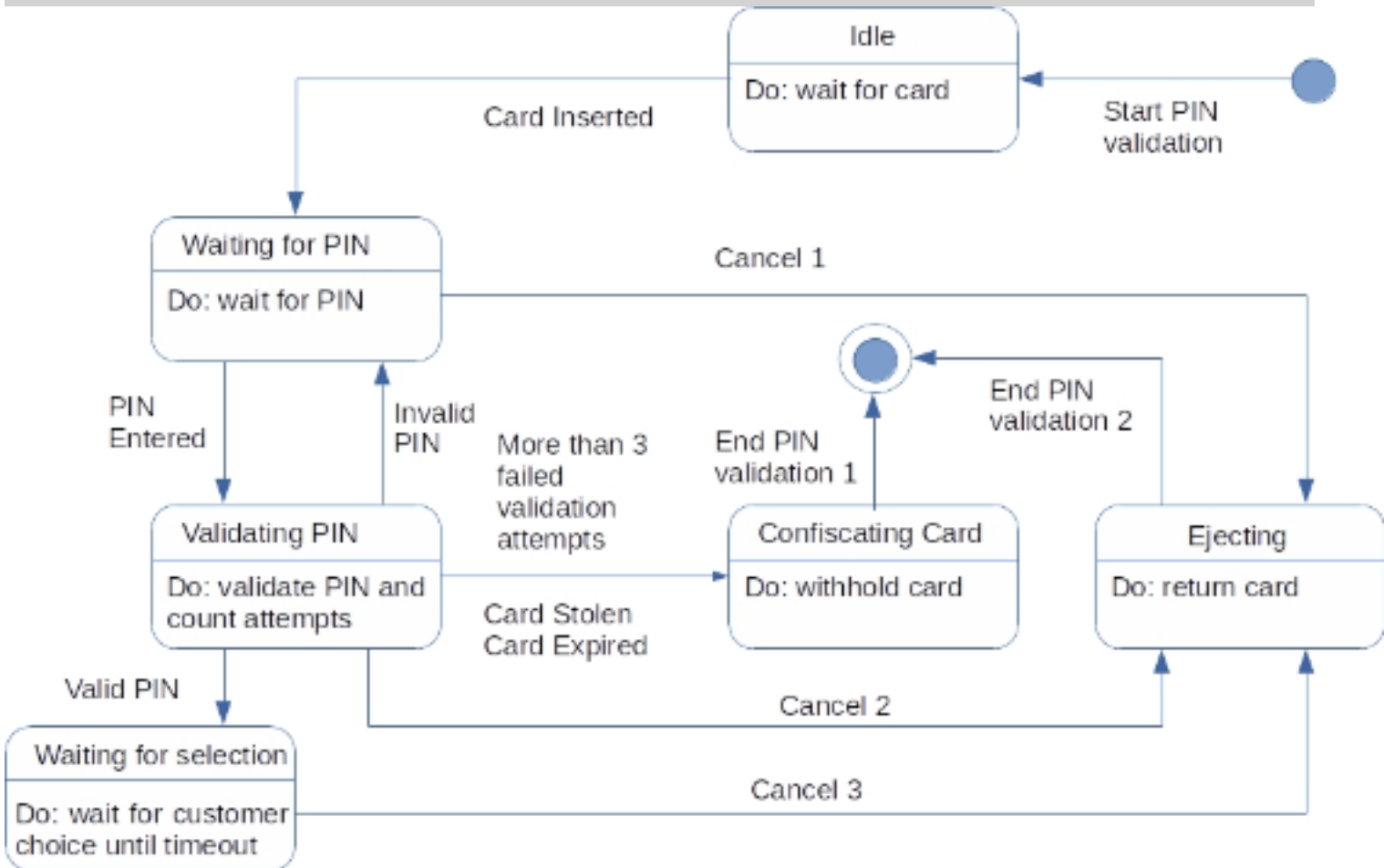


1.100%

2.80%
3.90%

Risposta : 3

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

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

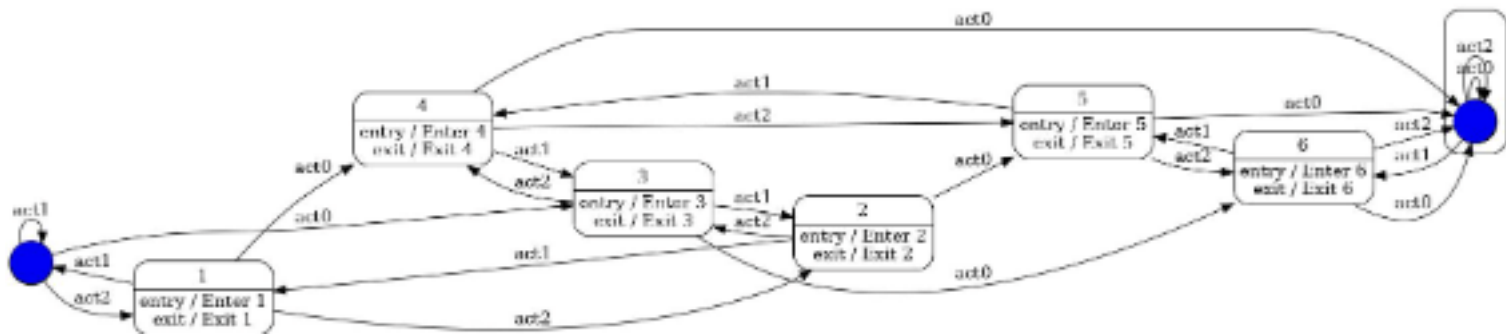
3.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer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

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

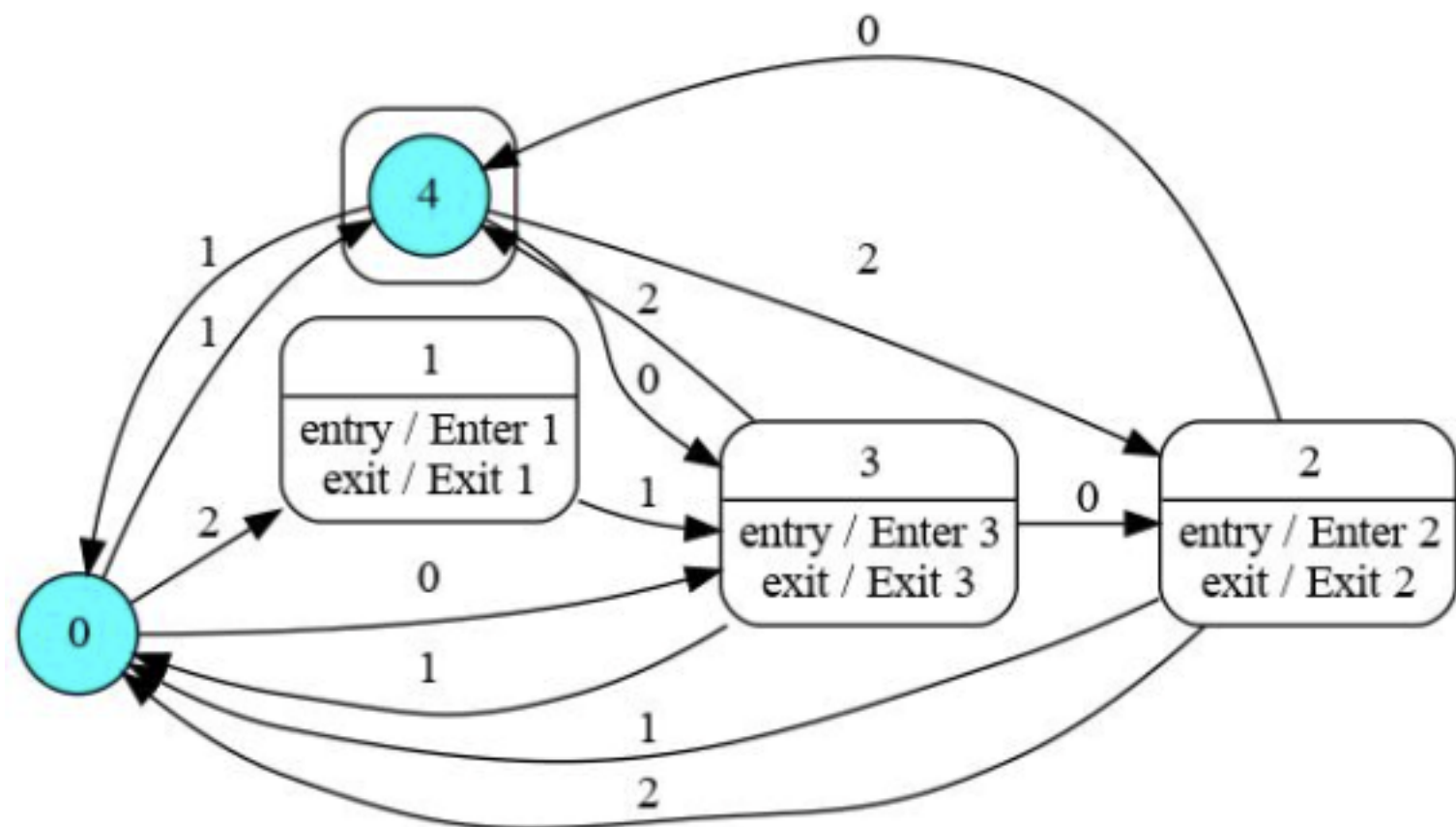


- 1.40%
- 2.80%
- 3.60%

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; 0, 0, 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] + 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; 0, p, 1-p, 0; 0, 0, 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] + 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; 0, p, 1-p, 0; 0, 0, 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] + A[x, F3]) then x := F3; else x := End;end if;end when;end System;

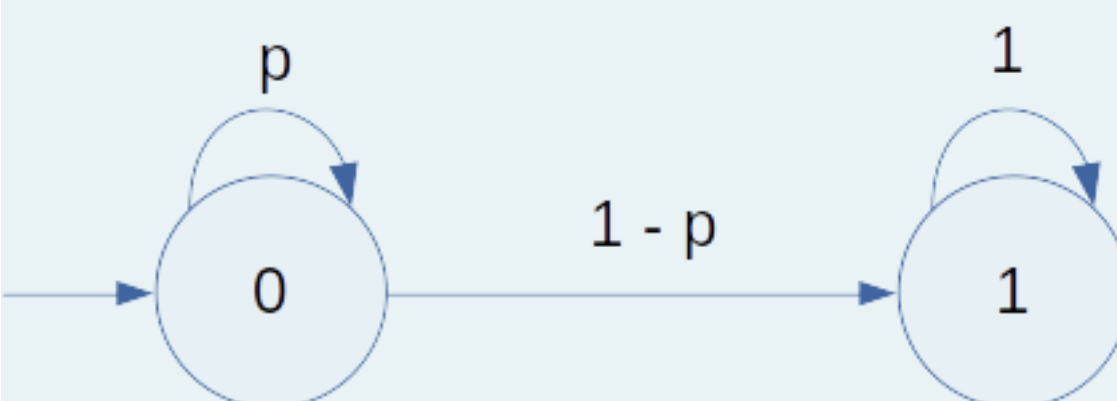
```

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?

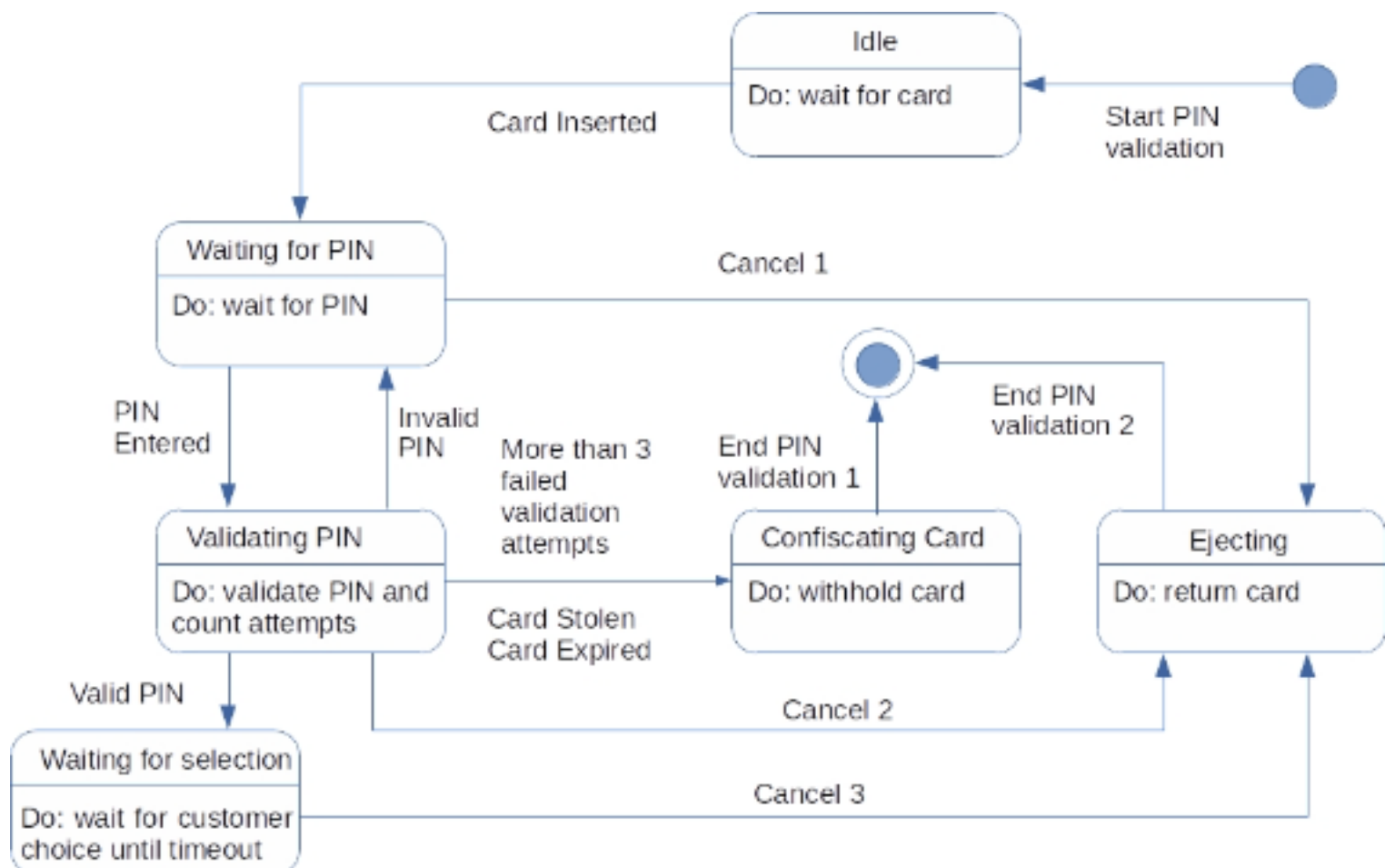


1.model Systemparameter Integer F1 = 1;parameter Integer F2 = 2;parameter Integer F3 = 3;parameter Integer End = 4;parameter Real p = 0.3;parameter Real A[4, 4] =[0, 1, 0, 0;p, 1-p, 0, 0;0, 0, p, 1-p;0, 0, 0, 1];Integer x; Real r1024;Integer state1024[Modelica.Math.Random.Generators.Xorshift1024star.nState];algorithmwhen 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] + 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] =[0, 1, 0, 0;p, 1-p, 0;0, p, 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;

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

Quale pattern architetturale meglio descrive l'architettura in figura ?



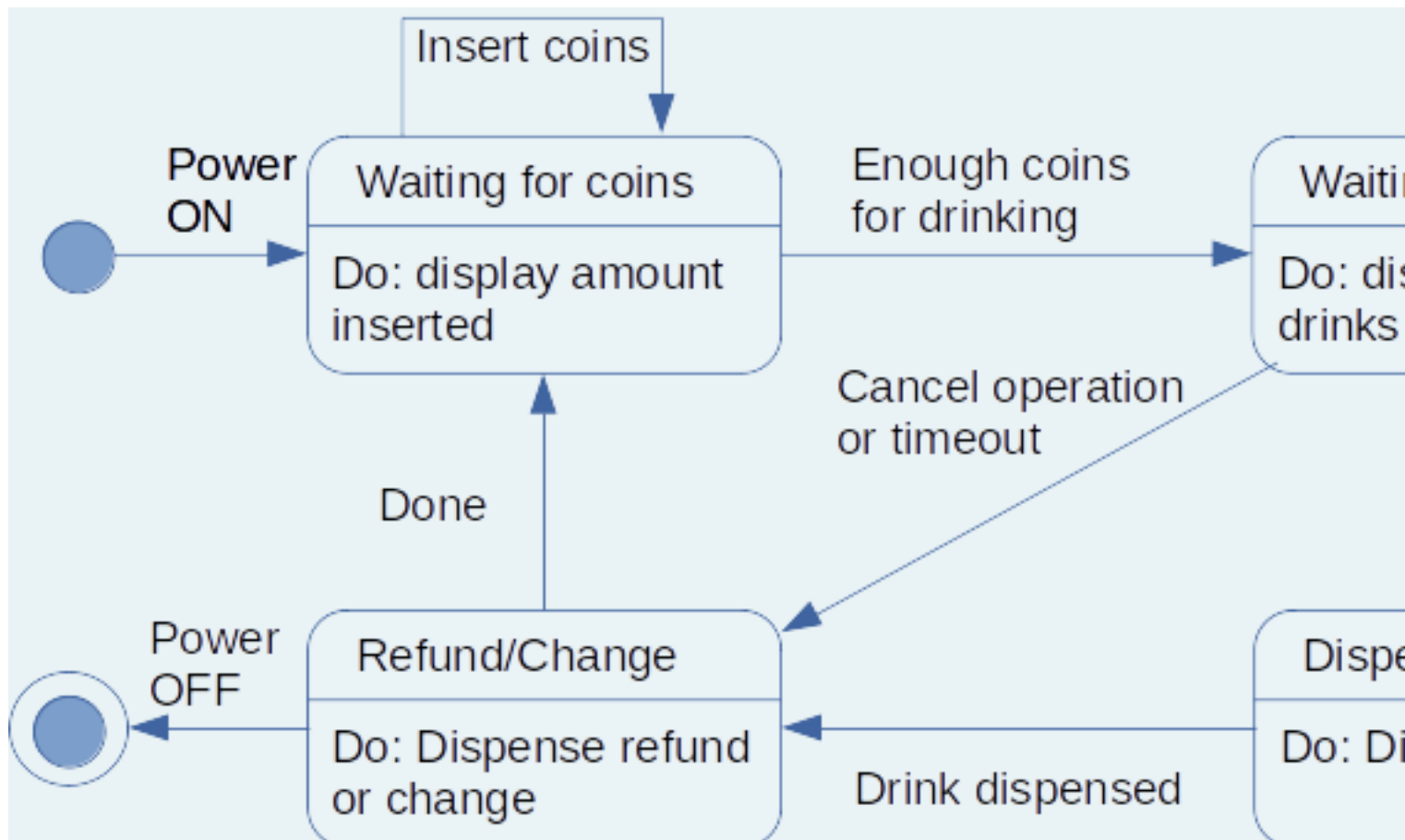
1.Pipe and filter architecture.

2.Model View Controller.

3.Layred architecture.

Risposta : 3

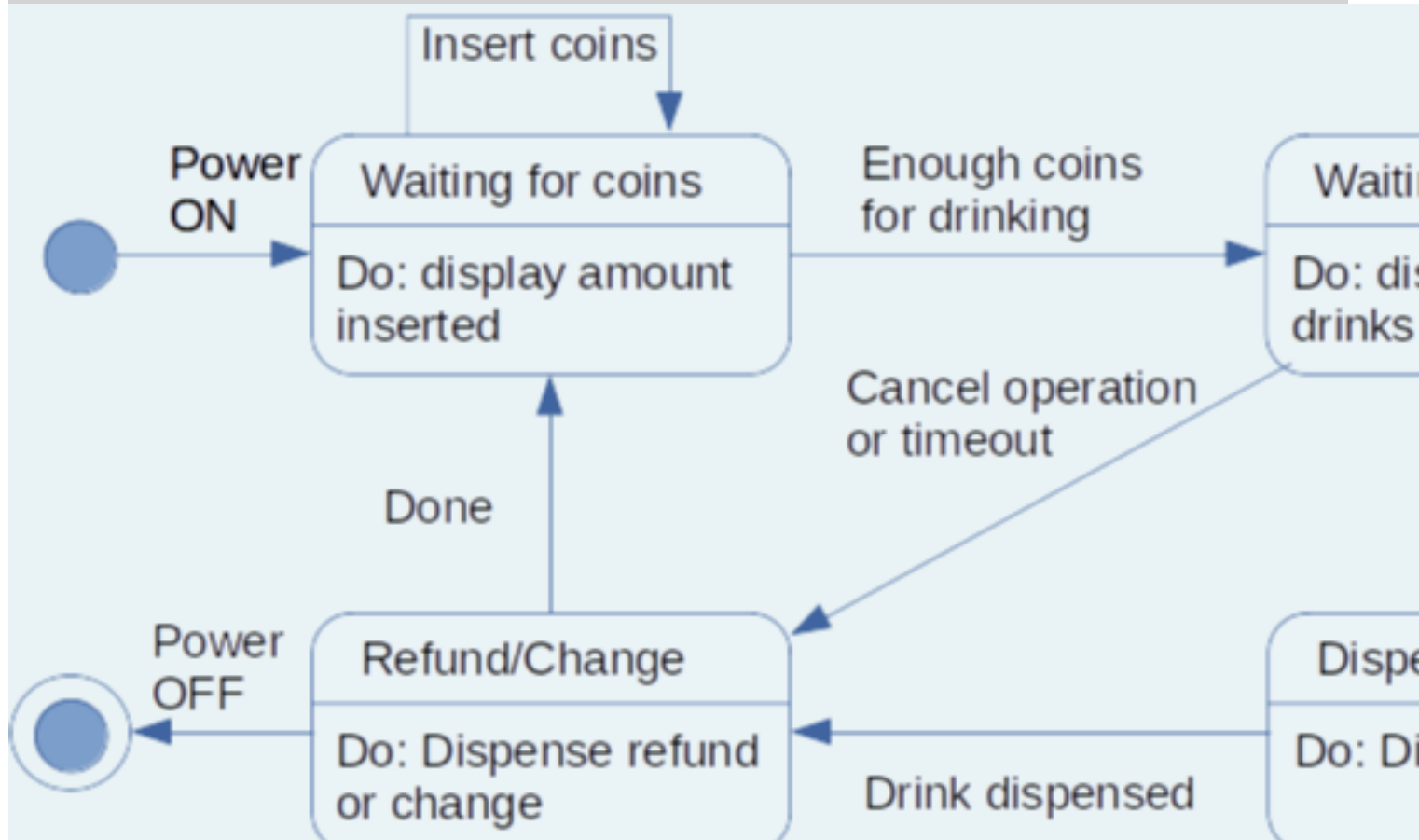
Quale delle seguenti frasi è corretta riguardo all'ctivity diagram in figura ?



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.

Risposta : 2

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?



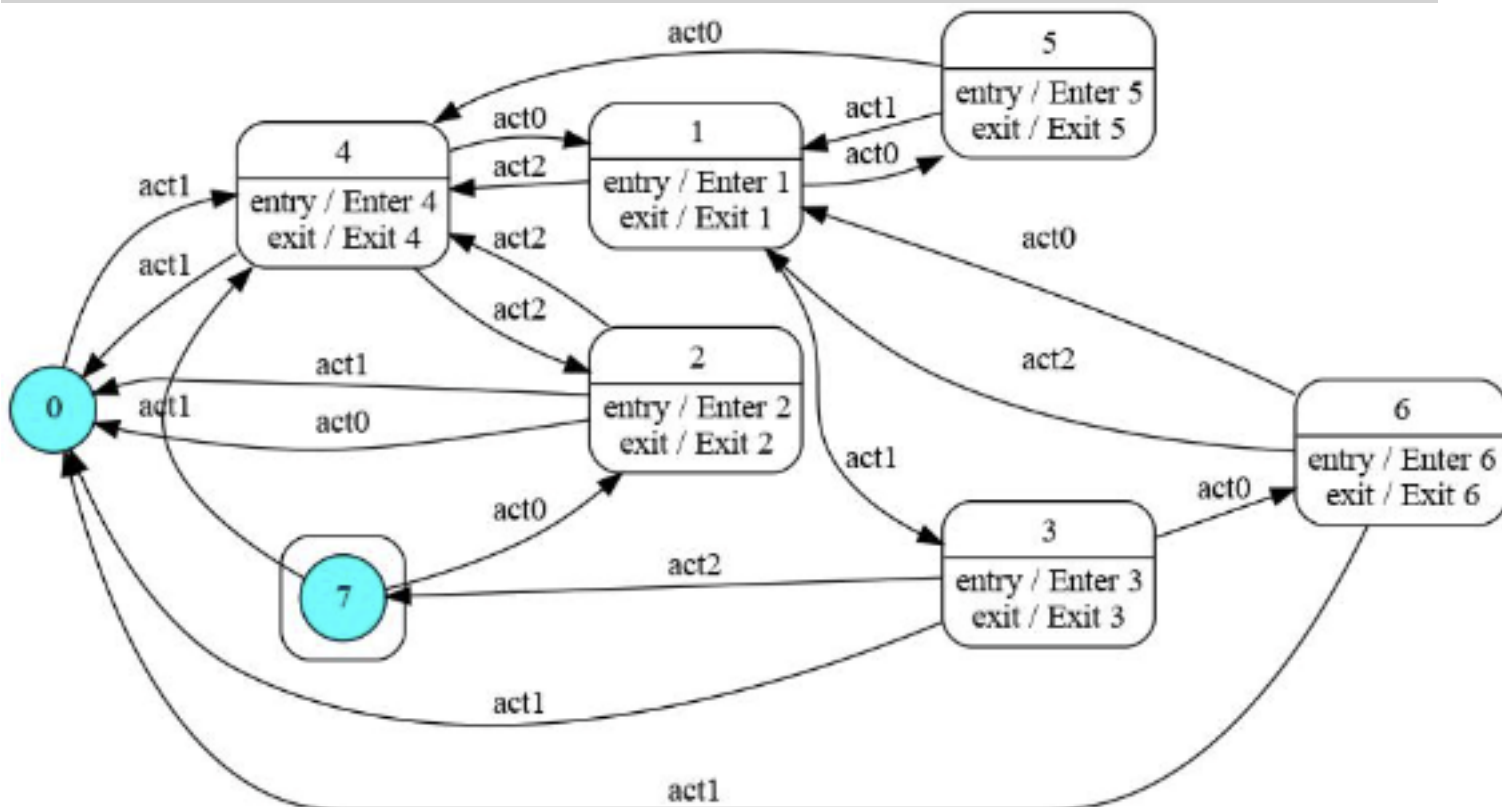
1.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer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

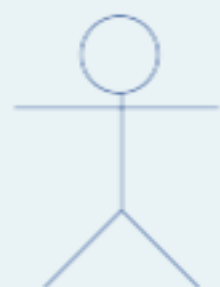
3.block CoffeeMachineparameter Real T = 1; // clockInputInteger Customer2Machine;OutputInteger Machine2Customer;/*0: nop1: enough coins inserted2: drink dispensed3: done*/Integer state;/*0: waiting for coins1: waiting for selection2: dispensing 3: refund/change*/algorithmwhen initial() thenstate := 0;Machine2Customer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

Risposta : 1

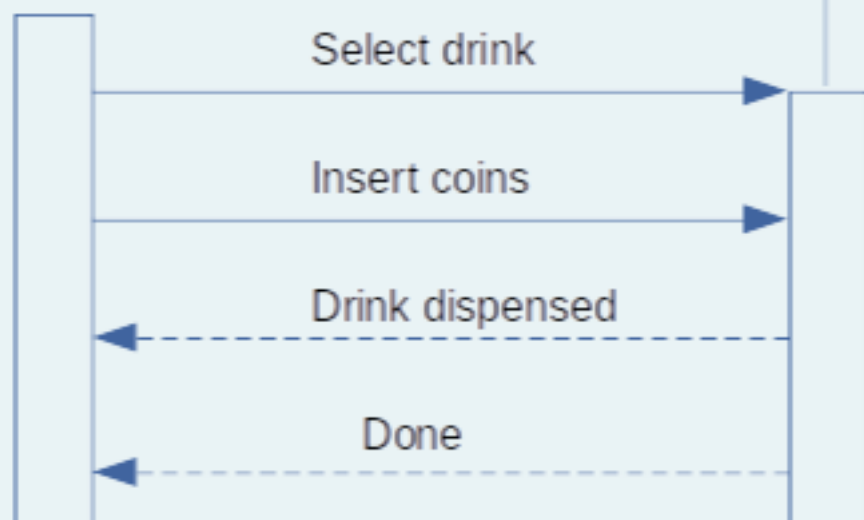
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale dei seguenti Sequence Diagram è consistente con lo State Diagram in figura ?



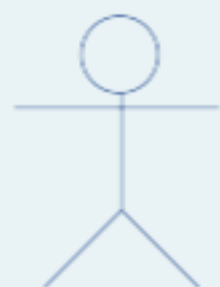
Customer



Vending Machine



Customer



Vending
Machine

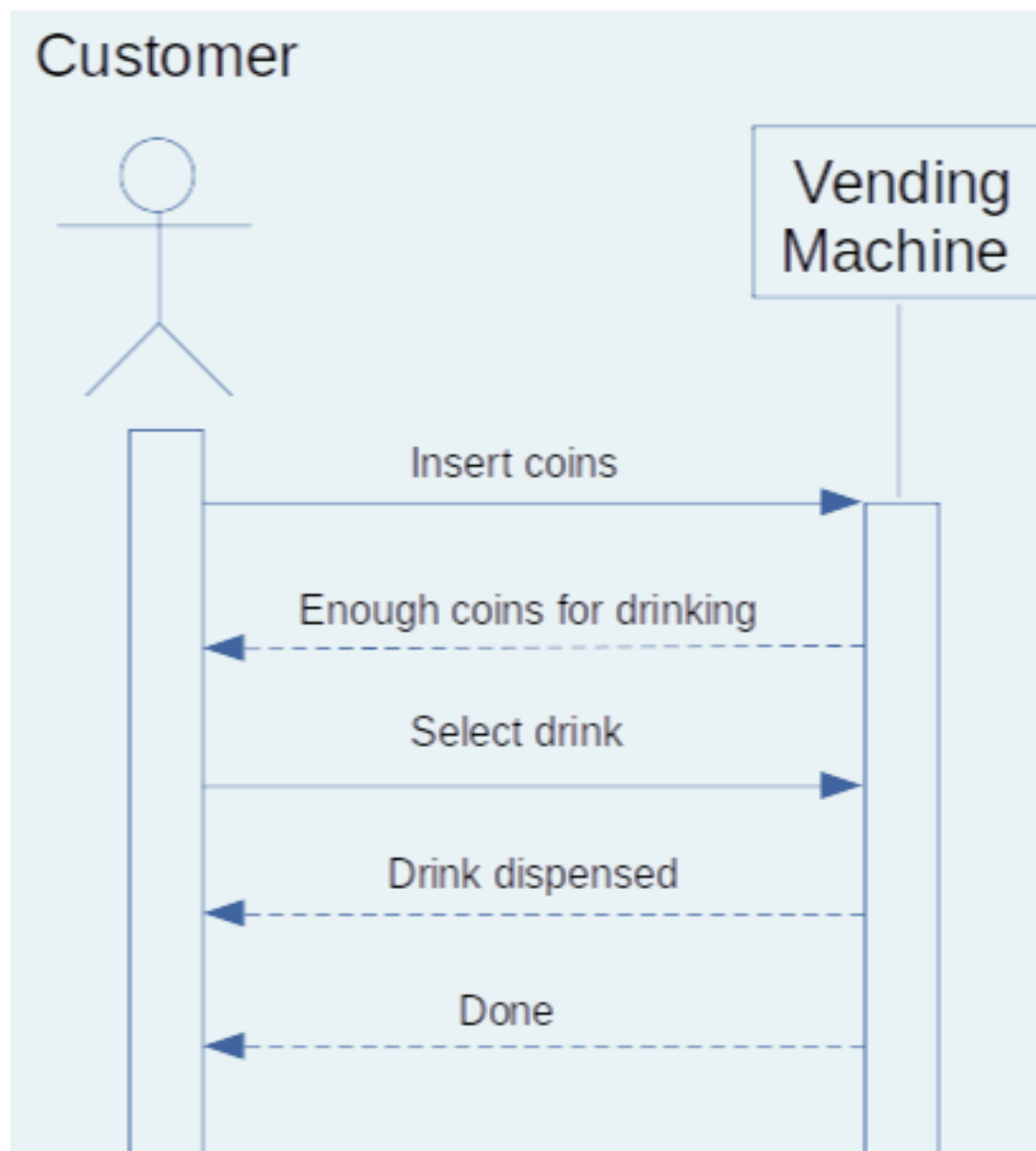
Insert coins

Enough coins for drinking

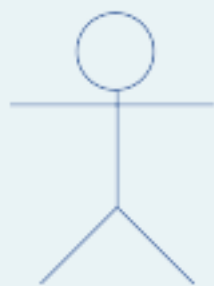
Select drink

Drink dispensed

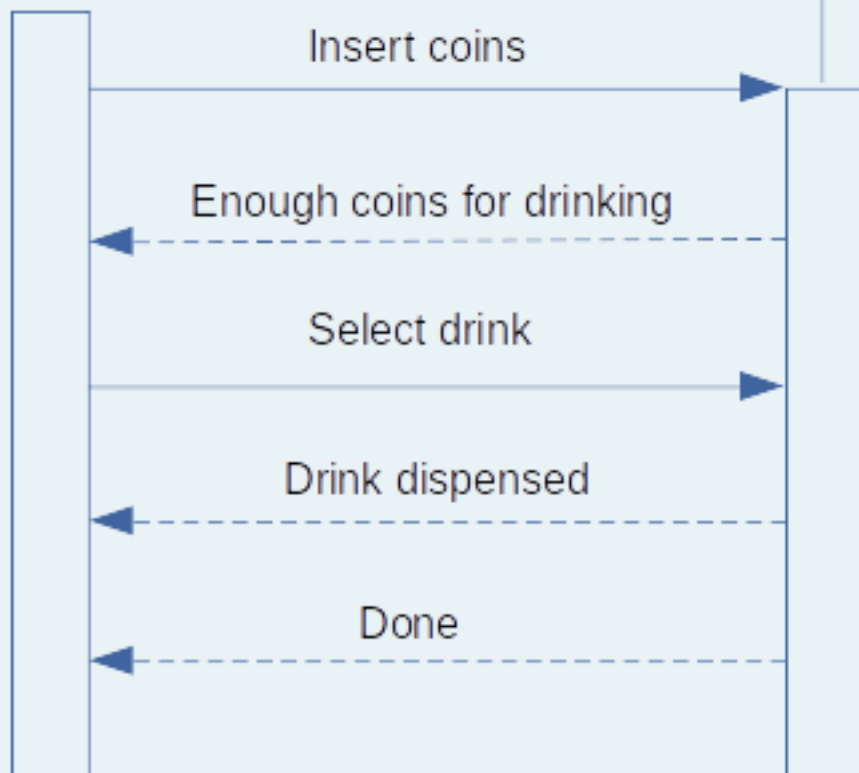
Done



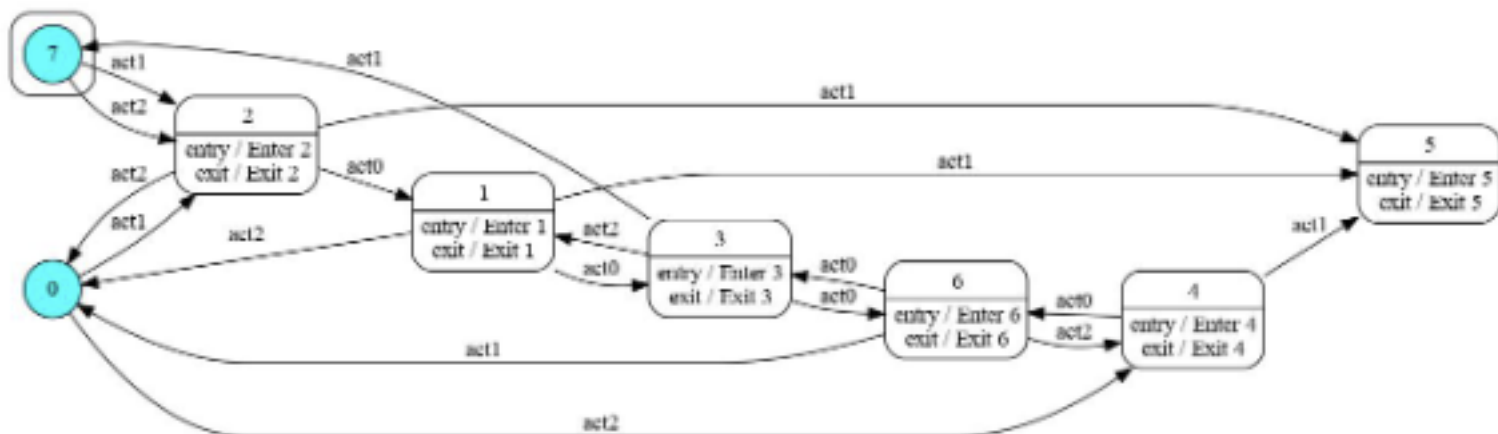
Customer



Vending Machine



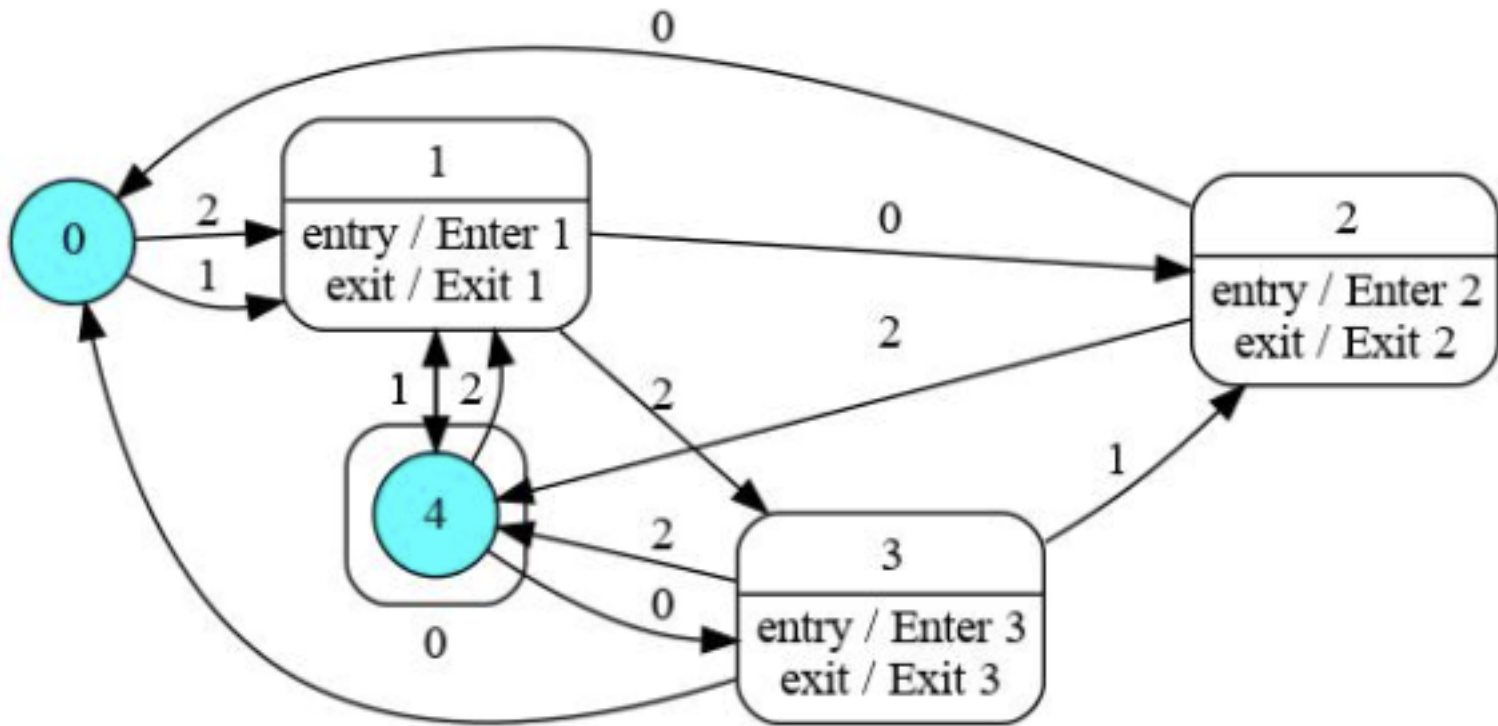
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?



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

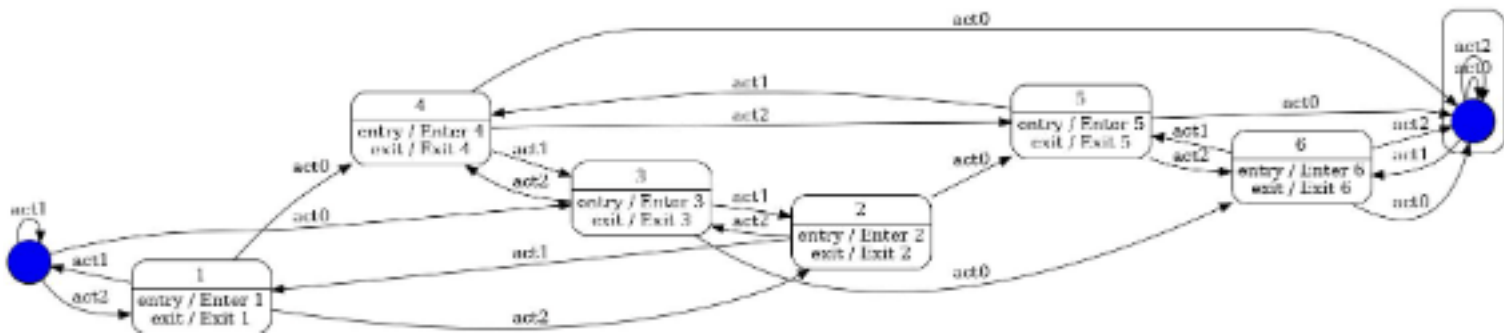
Lo State Diagram in figura descrive (in modo semplificato) una macchina distributrice di bevande. Quale delle seguenti frasi è corretta riguardo allo State Diagram in figura ?



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.

Risposta : 3

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 : 2