

```

1 /* Tema Esame: Schema - Esercizio 5 (logica Asmeta) */
2
3 asm semaforo
4
5 import StandardLibrary
6 import CTLlibrary
7 import LTLLibrary
8
9 signature:
10   enum domain NumSemaforo = {SEM1, SEM2}
11   enum domain ColoreSemaforo = {ROSSO, GIALLO, VERDE}
12
13   // sem prende un numero di un semaforo e restituisce
14   // il suo colore attuale
15   controlled coloreSem : NumSemaforo -> ColoreSemaforo
16
17   // comando che dice quale dei due semafori cambiare
18   // da rosso -> a verde
19   monitored segnale : NumSemaforo
20
21   // funzione statica che mi restituisce il colore
22   // dell'altro semaforo rispetto a quello passato
23   // come parametro
24   static coloreAltroSem : NumSemaforo -> ColoreSemaforo
25
26   // derivate per CTL/LTL
27   derived tuttiVerdi : Boolean
28   derived tuttiRossi : Boolean
29
30 definitions:
31
32   // funzione statica
33   function coloreAltroSem($s in NumSemaforo) =
34     if $s=SEM1 then
35       coloreSem(SEM2)
36     else
37       coloreSem(SEM1)
38     endif
39
40   // funzioni derivata per CTL/LTL
41   function tuttiVerdi =
42     (forall $s in NumSemaforo with coloreSem($s)=VERDE)
43
44   function tuttiRossi =
45     (forall $s in NumSemaforo with coloreSem($s)=ROSSO)
46
47   macro rule r_verde($s in NumSemaforo) =
48     coloreSem($s) := VERDE
49
50   rule r_giallo ($s in NumSemaforo) =
51     coloreSem($s) := GIALLO
52
53   rule r_rosso ($s in NumSemaforo) =
54     coloreSem($s) := ROSSO
55
56   rule r_switchColore($s in NumSemaforo) =
57     switch(coloreSem($s))
58       case VERDE:
59         r_giallo[$s]
60       case GIALLO:
61         r_rosso[$s]
62     endswitch

```

```

63
64 //-----
65 // PROPRIETA' LTL/CTL:
66
67 // 1) non accade mai che i semafori siano entrambi verdi
68 // utilizziamo una "derived" per la definizione della regola
69 CTLSPEC ag(not tuttiVerdi)
70 LTLSPEC g(not tuttiVerdi)
71
72 // 2) il semaforo 2 può diventare sempre verde (non solo allo stato iniziale)
73 CTLSPEC ag(ef(coloreSem(SEM2)=VERDE))
74
75 // 3) se un semaforo è verde allora l'altro è rosso
76 // utilizziamo il forall senza utilizzare una "derived"
77 LTLSPEC g(
78   ( forall $s in NumSemaforo with
79     coloreSem($s)=VERDE implies coloreAltroSem($s)=ROSSO )
80 )
81
82 CTLSPEC ag(
83   ( forall $s in NumSemaforo with
84     coloreSem($s)=VERDE implies coloreAltroSem($s)=ROSSO )
85 )
86 // 4) se entrambi i semafori sono rossi e viene scelto il semaforo 1,
87 // nello stato successivo il semaforo 1 è verde
88 CTLSPEC ag( (tuttiRossi and segnale=SEM1) implies ax(coloreSem(SEM1)=VERDE) )
89
90 // 5) il semaforo 1 non può mai essere verde
91 // mi aspetto che sia FALSA perché sicuramente esiste un percorso
92 // in cui il semaforo 1 può diventare verde
93 CTLSPEC ag(coloreSem(SEM1)!=VERDE)
94 // controesempio fornito:
95 /*
96   -> State: 1.1 <-
97   coloreSem(SEM1) = ROSSO
98   segnale = SEM1
99   coloreSem(SEM2) = ROSSO
100   coloreAltroSem(SEM2) = ROSSO
101   coloreAltroSem(SEM1) = ROSSO
102   -> State: 1.2 <-
103   coloreSem(SEM1) = VERDE
104   coloreAltroSem(SEM2) = VERDE
105 */
106
107 // PROPRIETA' DEFINITE DA ME NON SCRITTE NELL'ESERCIZIO:
108 // 1') i semafori sono sempre tutti rossi
109 // mi aspetto che sia falsa
110 CTLSPEC ag(tuttiRossi)
111 // controesempio:
112 /*
113   -> State: 2.1 <-
114   coloreSem(SEM1) = ROSSO
115   coloreSem(SEM2) = ROSSO
116   segnale = SEM1
117   coloreAltroSem(SEM2) = ROSSO
118   coloreAltroSem(SEM1) = ROSSO
119   -> State: 2.2 <-
120   coloreSem(SEM1) = VERDE
121   coloreAltroSem(SEM2) = VERDE
122 */
123
124 //-----

```

```
125
126  main rule r_Main =
127  par
128      // passo a verde SOLO se entrambi i semafori sono rossi
129      if(coloreSem(segnale)=ROSSO and coloreAltroSem(segnale)=ROSSO) then
130          r_verde[segnale]
131      endif
132      // se il semaforo indicato da "segnale" NON è rosso e viene
133      // richiesto di cambiarlo allora cambio il suo colore
134      if coloreSem(segnale)!=ROSSO then
135          r_switchColore[segnale]
136      endif
137  endpar
138
139 default init s0:
140     function coloreSem($s in NumSemaforo) = ROSSO
```

```
1/* Tema Esame: Schema - Esercizio 5 (scenario Avalla) */
2
3scenario scenario1
4
5load semaforo.asm
6
7// controllo che all'inizio i semafori siano entrambi ROSSI
8check coloreSem(SEM1) = ROSSO;
9check coloreSem(SEM2) = ROSSO;
10
11// chiedo di cambiare il colore del SEM1
12set segnale := SEM1;
13step
14
15// mi assicuro che sia cambiato solo il semaforo 1
16check coloreSem(SEM1) = VERDE;
17check coloreSem(SEM2) = ROSSO;
18
19// chiedo di cambiare il colore del SEM2 e mi assicuro
20// che il colore di SEM2 non cambi
21set segnale := SEM2;
22step
23
24check coloreSem(SEM1) = VERDE;
25check coloreSem(SEM2) = ROSSO;
26
27// concludo il ciclo del SEM1 passando prima a giallo
28// poi a rosso
29
30// SEM 1 da verde -> a giallo
31set segnale := SEM1;
32step
33check coloreSem(SEM1) = GIALLO;
34check coloreSem(SEM2) = ROSSO;
35
36// SEM 1 da giallo -> a rosso
37set segnale := SEM1;
38step
39check coloreSem(SEM1) = ROSSO;
40check coloreSem(SEM2) = ROSSO;
41
42
```