```
 1
 2 /* TemaEsame_Schema_Inventato2: Esercizio 5 */
 3 /* Logica Asmeta */
 4
 5 asm dadi
 6
 7 import StandardLibrary
 8 import CTLlibrary
 9 import LTLlibrary
10
11 signature:
12     enum domain Giocatore = {USER,PC}
13     enum domain Risultato = {WINUSER, WINPC, PATTA}
14     // enumerativo che indica lo stato della partita
15     enum domain StatoGioco = {FINE, INCORSO}
16
17     domain NumSoldi subsetof Integer
18     domain FacceDado subsetof Integer
19
20     controlled soldi : Giocatore -> NumSoldi
21
22     monitored lancioUser : FacceDado
23     monitored lancioPC : FacceDado
24
25     // controlled che indica il risultato della giocata
26     // (del turno precedente perché l'aggiornamento avviene
27     // durante la transizione e non subito)
28     controlled risultatoGiocata : Risultato
29
30     // controlled che indica lo stato del gioco
31     controlled statoGioco : StatoGioco
32
33     // controlled che indica chi è il vincitore del gioco
34     controlled vincitore : Giocatore
35
36     // funzione statica che, date due facce del
37     // dado, restituisce chi è il vincitore
38     static risultato : Prod(FacceDado, FacceDado) -> Risultato
39
40     // funzioni derivate per semplificare scrittura di CTL/LTL
41     derived inRange : Giocatore -> Boolean
42
43 definitions:
44     domain NumSoldi = {0:20}
45     domain FacceDado = {1:6}
46
47     function risultato($u in FacceDado, $p in FacceDado) =
48          if ($u > $p) then
49             WINUSER
50          else
51             if($u<$p) then
52                 WINPC
53             else
54                 PATTA
55             endif
56          endif
57
58     // funzione che restituisce se i soldi del giocatore $g
59     // sono compresi tra 0 e 20
60     function inRange($g in Giocatore) =
61             0<=soldi($g) and soldi($g)<=20
62
```

```
63      macro rule r_meno1($g in Giocatore) =
64          soldi($g) := soldi($g)-1
65
66      rule r_piu1($g in Giocatore) =
67          soldi($g) := soldi($g)+1
68
69      // ------------------------------------------------
70      // PROPRIETA':
71      // 1) il saldo dell'utente può assumere un qualsiasi valore nell'intervallo[0, 20]€
72          LTLSPEC g(inRange(USER))
73          CTLSPEC ag(inRange(USER))
74
75      // 2) nel sistema ci sono sempre 20€
76          LTLSPEC g(soldi(USER)+soldi(PC)=20)
77          CTLSPEC ag(soldi(USER)+soldi(PC)=20)
78
79      // 3) esiste un cammino in cui il saldo dell'utente è sempre maggiore di 1€
80          CTLSPEC eg(soldi(USER)>=1)
81
82      // 4) in ogni caso prima o poi la partita finirà sempre
83      // mi aspetto che questa sia falsa perché potrebbe succedere che
84      // la partita continui a ciclare tra aumento/diminuzione soldi
85          CTLSPEC af(statoGioco=FINE)
86      // viene infatti dimostrata falsa e fornito un controesempio
87
88      // MIE PROPRIETA':
89      // 5) il PC non vince mai
90          CTLSPEC ag(vincitore != PC)
91      // viene fornito un controesempio in cui vince il PC
92
93      // 6) il saldo di tutti i giocatori può assumere un qualsiasi valore nell'intervallo[0,
    20]€
94          LTLSPEC g( (forall $g in Giocatore with inRange($g)) )
95      // ------------------------------------------------
96
97      main rule r_Main =
98          // il gioco termina quando uno dei due giocatori ha
99          // terminato i soldi
100         if ( soldi(USER)=0 or soldi(PC)=0 ) then
101             par
102                 statoGioco := FINE
103                 if soldi(USER)=0 then
104                     vincitore := PC
105                 else
106                     vincitore := USER
107                 endif
108             endpar
109         else
110             par
111                 statoGioco := INCORSO
112                 risultatoGiocata := risultato(lancioUser, lancioPC)
113                     if risultato(lancioUser, lancioPC)=WINUSER then
114                         // vince l'utente
115                         par
116                             r_piu1[USER]
117                             r_meno1[PC]
118                         endpar
119                     else
120                         if risultato(lancioUser, lancioPC)=WINPC then
121                             //vince il pc
122                             par
123                                 r_meno1[USER]
```

```
124                                 r_piu1[PC]
125                             endpar
126                         endif
127                         // se è patta, non faccio nulla
128                     endif
129             endpar
130         endif
131
132 default init s0:
133     // all'inizio entrambi i giocatori hanno 10€
134     function soldi($g in Giocatore) = 10
135     function statoGioco = INCORSO
```

```
1
2 /* TemaEsame_Schema_Inventato2: Esercizio 5 */
3 /* Scenario Avalla */
4
5 scenario scenarioDadi
6
7 load dadi.asm
8
9 // faccio uno scenario in cui:
10 // - l'utente perde alla prima giocata
11 // - l'utente pareggia alla seconda giocata
12 // - l'utente vince alla terza giocata
13
14 check soldi(USER)=10;
15 check soldi(PC)=10;
16
17 // l'utente perde
18 set lancioUser := 1;
19 set lancioPC := 6;
20 step
21 check soldi(USER)=9;
22 check soldi(PC)=11;
23 check risultatoGiocata=WINPC;
24
25 // l'utente pareggia
26 set lancioUser := 6;
27 set lancioPC := 6;
28 step
29 check soldi(USER)=9;
30 check soldi(PC)=11;
31 check risultatoGiocata=PATTA;
32
33 // l'utente vince
34 set lancioUser := 6;
35 set lancioPC := 1;
36 step
37 check soldi(USER)=10;
38 check soldi(PC)=10;
39 check risultatoGiocata=WINUSER;
40
41
```