

```

1 MODULE main
2 VAR
3   state : {s1, s2, s3, s4, s5,s6,s7};
4 ASSIGN
5
6   next(state) :=
7
8       case
9
10          state = s1 : {s2, s4};
11          state = s2 : s3;
12          state = s3 : {s2, s4};
13          state = s4 : {s1, s6};
14          state = s5 : {s1, s4, s5};
15          state = s6 : s7;
16          state = s7 : s5;
17
18       esac;
19
20 DEFINE
21
22 start  := (state = s2) | (state = s3) | (state = s6) | (state = s7);
23 closed := (state = s3) | (state = s4) | (state = s5) | (state = s6) |
   (state = s7);
24 heat   := (state = s5) | (state = s7);
25 error  := (state = s2) | (state = s3);
26
27
28
29 -----
30 -- 1 EXPRESSAO CONVERTIDA:
31 ---- ORIGINAL ->    AG( start -> AF heat )
32
33 ---- TRADUÇÃO : SEMPRE QUE O MICROONDAS ESTIVER NO ESTADO 'START', EM ALGUM
   MOMENTO DO FUTURO ELE PODERÁ AQUECER (HEAT).
34
35 ---- !EF !( !start | ! EG !heat)
36
37 -- ou
38
39 ---- !E[TRUE U !(start | !EG !heat)]
40 -----
41
42
43
44 -----
45 -- 2 EXPRESSAO CONVERTIDA:
46 -- ORIGINAL -> A[!heat U closed]
47
48
49 -- TRADUÇÃO : O MICROONDAS SEMPRE ESTARÁ OU FECHADOU OU NÃO ESTARÁ QUENTE
50 -- !(E [(!closed) U (!(heat | closed))] | (EG (!closed)))
51 -----

```

```
52
53
54 -----
55 -- 3 EXPRESSAO CONVERDIDA:
56 -- ORIGINAL -> AG( error -> AF !error )
57
58 -- TRADUÇÃO: SEMPRE QUE HOUVER UM ERRO (ERROR), EM ALGUM MOMENTO NO FUTURO
   O ERRO DEIXARÁ DE EXISTIR (!ERROR)
59 -- !EF !(!error | !EG(error))
60 -----
61
62 -- 4 EXPRESSAO (NOVA):
63 -- EF(AG(heat & closed))
64 -- EF( !(!EF !(heat & closed)) | !E[TRUE U !(heat & closed)])
```