

```

1  ┌────────────────────────── MODULE Euclid ───────────────────────────┐
2  EXTENDS Integers, GCD, TLC
3
4  CONSTANTS N
5
6  ASSUME  $\wedge N \in \text{Nat} \setminus \{0\}$ 
7
8  *****
9  --fair algorithm Euclid{
10   variables  $x \in 1 \dots N, y \in 1 \dots N, x0 = x, y0 = y$ ;
11   { abc: while (  $x \neq y$  )
12     { d: if (  $x < y$  ) {  $y := y - x$ ; }
13       else {  $x := x - y$ ; }
14     } ;
15
16   at: assert (  $x = y$  )  $\wedge$  (  $x = \text{GCD}(x0, y0)$  );
17 }
18 }
19 *****
20 BEGIN TRANSLATION
21 VARIABLES  $x, y, x0, y0, pc$ 
22
23 vars  $\triangleq \langle x, y, x0, y0, pc \rangle$ 
24
25 Init  $\triangleq$  Global variables
26    $\wedge x \in 1 \dots N$ 
27    $\wedge y \in 1 \dots N$ 
28    $\wedge x0 = x$ 
29    $\wedge y0 = y$ 
30    $\wedge pc = \text{"abc"}$ 
31
32 abc  $\triangleq$   $\wedge pc = \text{"abc"}$ 
33    $\wedge$  IF  $x \neq y$ 
34     THEN  $\wedge pc' = \text{"d"}$ 
35     ELSE  $\wedge pc' = \text{"at"}$ 
36    $\wedge$  UNCHANGED  $\langle x, y, x0, y0 \rangle$ 
37
38 d  $\triangleq$   $\wedge pc = \text{"d"}$ 
39    $\wedge$  IF  $x < y$ 
40     THEN  $\wedge y' = y - x$ 
41        $\wedge x' = x$ 
42     ELSE  $\wedge x' = x - y$ 
43        $\wedge y' = y$ 
44    $\wedge pc' = \text{"abc"}$ 
45    $\wedge$  UNCHANGED  $\langle x0, y0 \rangle$ 
46
47 at  $\triangleq$   $\wedge pc = \text{"at"}$ 

```

```

48       $\wedge \text{Assert}((x = y) \wedge (x = \text{GCD}(x0, y0)),$ 
49          "Failure of assertion at line 16, column 9.")
50       $\wedge pc' = \text{"Done"}$ 
51       $\wedge \text{UNCHANGED } \langle x, y, x0, y0 \rangle$ 

53   $\text{Next} \triangleq abc \vee d \vee at$ 
54       $\vee$  Disjunct to prevent deadlock on termination
55       $(pc = \text{"Done"} \wedge \text{UNCHANGED } vars)$ 

57   $\text{Spec} \triangleq \wedge \text{Init} \wedge \Box[\text{Next}]_{vars}$ 
58       $\wedge \text{WF}_{vars}(\text{Next})$ 

60   $\text{Termination} \triangleq \Diamond(pc = \text{"Done"})$ 

62  END TRANSLATION

64  |_____|

```