

```

1 MACHINE
2   Robot_1
3 REFINES
4   Robot_0
5 SEES
6   Robot_0_Ctx
7 VARIABLES  $t$ ,  $pA$ ,  $Target$ ,  $Direction$ 
8 EVENTS
9   INITIALISATION
10  END
11
12  Behave
13  REFINES  $Behave$ 
14  END
15
16  sense_close_enough
17  REFINES  $sense\_close\_enough$ 
18  END
19
20  transition_change_direction
21  REFINES  $transition\_change\_direction$ 
22  END
23
24  transition_change_target
25  REFINES  $transition\_change\_target$ 
26  END
27
28  actuate_movement
29  REFINES  $actuate\_movement$ 
30  ANY  $tp$ 
31  WHERE
32    grd1:  $plannar\_distance(Target, pA(t)) \mapsto CloseEnough \in gt$ 
33    grd7:  $tp \in \mathbb{R}^+$ 
34    grd8:  $t < tp$ 
35    grd9:
36       $CBAPlutionOfFIS(t, tp, pA,$ 
37         $withControl($ 
38           $[t, tp],$ 
39           $FirstOrder2DimensionSystem(DeltaNeighborhoodSet(SpeedLimit, 0 \mapsto 0), t, pA(t)),$ 
40           $PointwiseControl([t, tp], prj1(Direction), prj2(Direction), t)$ 
41         $)$ 
42      ,  $\{px\_ \mapsto py\_ \mid plannar\_distance(Target, px\_ \mapsto py\_ ) \mapsto CloseEnough \in gt$ 
43         $\wedge plannar\_distance(0 \mapsto 0, px\_ \mapsto py\_ ) \mapsto CriticalDistance \in lt\}$ 
44  WITH
45    e:
46    e =  $withControl($ 
47       $[t, tp],$ 
48       $FirstOrder2DimensionSystem(DeltaNeighborhoodSet(SpeedLimit, 0 \mapsto 0), t, pA(t)),$ 
49       $PointwiseControl([t, tp], prj1(Direction), prj2(Direction), t)$ 
50     $)$ 
51  THEN
52    act1:
53       $t, pA :|$ 
54       $pA' \in \mathbb{R} \mapsto S \wedge t' = tp \wedge$ 
55       $[0, t'] \subseteq dom(pA') \wedge$ 
56       $CBAPlutionOf($ 
57         $t, t',$ 
58         $pA, pA',$ 
59         $withControl($ 
60           $[t, t'],$ 
61           $FirstOrder2DimensionSystem(DeltaNeighborhoodSet(SpeedLimit, 0 \mapsto 0), t, pA(t)),$ 
62           $PointwiseControl([t, t'], prj1(Direction), prj2(Direction), t)$ 
63         $)$ 
64      ,  $\{px\_ \mapsto py\_ \mid plannar\_distance(Target, px\_ \mapsto py\_ ) \mapsto CloseEnough \in gt$ 
65         $\wedge plannar\_distance(0 \mapsto 0, px\_ \mapsto py\_ ) \mapsto CriticalDistance \in lt\}$ 
66       $)$ 
67  END
68
69 END

```