

Contents

CONTEXT c0	2
CONTEXT c1	3
CONTEXT c2	4
CONTEXT c3	5
MACHINE m0	6
MACHINE m1	7
MACHINE m2	8
MACHINE m3	10

CONTEXT c0**CONSTANTS**

NCOUNTERS

AXIOMSaxm1: $NCOUNTERS \in \mathbb{N}_1$ **END**

CONTEXT c1

EXTENDS c0

SETS

COUNTERS

AXIOMS

axm2: $\text{finite}(\text{COUNTERS})$

axm1: $\text{card}(\text{COUNTERS}) = \text{NCOUNTERS}$

END

CONTEXT c2

EXTENDS c1

SETS

SCREEN

CONSTANTS

WAIT

NOWAIT

AXIOMS

axm2: $SCREEN = \{WAIT, NOWAIT\}$

axm1: $WAIT \neq NOWAIT$

END

CONTEXT c3

EXTENDS c2

AXIOMS

axm1: $\forall S, y. S \subseteq COUNTERS \wedge finite(S) \wedge card(S) = 1 \wedge y \in S \Rightarrow S = \{y\}$

END

MACHINE m0**SEES** c0**VARIABLES***nclients***INVARIANTS***inv1*: $nclients \in 0 .. NCOUNTERS$ *DLF*: $\langle \text{theorem} \rangle nclients < NCOUNTERS \vee nclients > 0$ **EVENTS****Initialisation****begin***act1*: $nclients := 0$ **end****Event** arrive $\langle \text{ordinary} \rangle \hat{=}$ **when***grd1*: $nclients < NCOUNTERS$ **then***act1*: $nclients := nclients + 1$ **end****Event** leave $\langle \text{ordinary} \rangle \hat{=}$ **when***grd1*: $nclients > 0$ **then***act1*: $nclients := nclients - 1$ **end****END**

```

MACHINE m1
REFINES m0
SEES c1
VARIABLES
    busy
INVARIANTS
    inv1:  $busy \subseteq COUNTERS$ 
    inv2:  $nclients = card(busy)$ 
EVENTS
Initialisation
    begin
        act1:  $busy := \emptyset$ 
    end
Event arrive  $\langle ordinary \rangle \hat{=}$ 
refines arrive
    any
        c
    where
        grd2:  $c \in COUNTERS$ 
        grd1:  $c \notin busy$ 
    then
        act1:  $busy := busy \cup \{c\}$ 
    end
Event leave  $\langle ordinary \rangle \hat{=}$ 
refines leave
    any
        c
    where
        grd2:  $c \in busy$ 
    then
        act1:  $busy := busy \setminus \{c\}$ 
    end
END

```

MACHINE m2**REFINES** m1**SEES** c1**VARIABLES**

busy

next_counter

wait If wait = TRUE, WAIT is displayed in screen; otherwise 'next_counter' is displayed in screen

in_corridor

INVARIANTSinv1: next_counter \in COUNTERSinv7: wait \in BOOL

If FALSE, a counter # will be shown in screen

inv3: in_corridor \in {0, 1}inv4: in_corridor = 1 \Rightarrow wait = TRUE

** If anyone is walking to counter, wait

inv2: COUNTERS = busy \Rightarrow wait = TRUE

If there are no free counters, clients should wait

inv6: wait = FALSE \Rightarrow next_counter \notin busy

If showing a counter #, it must be of a free counter (needed for pass_screen/inv8/INV?)

inv8: in_corridor > 0 \Rightarrow next_counter \notin busy

If anyone is going to the counter, it is going to a free countre

inv9: (theorem) in_corridor > 0 \Rightarrow busy \neq COUNTERS

If anyone is going to the counter, there has to be a free counter

EVENTS**Initialisation** (extended)

begin

act1: busy := \emptyset

act2: wait := FALSE

It could be TRUE as well, and it will be updated by screen_num

act4: next_counter \in COUNTERS

act3: in_corridor := 0

end

Event enter (ordinary) $\hat{=}$

when

grd2: wait = FALSE

Note we do not need to specify in_corridor = FALSE (inv5, MT)

then

act1: in_corridor := in_corridor + 1

act2: wait := TRUE

wait := TRUE

end

Event arrive (ordinary) $\hat{=}$ **refines** arrive

when

grd1: in_corridor > 0

screen_num \neq busy inv.

with

c: c = next_counter

then

act1: in_corridor := in_corridor - 1

act2: busy := busy \cup {next_counter}

end

Event screen_num (ordinary) $\hat{=}$

when

grd2: COUNTERS \neq busy


```
    grd3: in_corridor = 0
           Check how this impacts convergence, invariants
    grd4: wait = TRUE
  then
    act1: next_counter :∈ COUNTERS \ busy
    act2: wait := FALSE
  end
Event leave ⟨ordinary⟩ ≐
extends leave
  any
    c
  where
    grd2: c ∈ busy
  then
    act1: busy := busy \ {c}
  end
END
```

MACHINE m3**REFINES** m2**SEES** c3**VARIABLES**

busy

next_counter

wait If wait = TRUE, WAIT is displayed in screen; otherwise 'next_counter' is displayed in screen

in_corridor

SCREEN_CNT Whether the screen *displays* WAIT or a counter number (NOWAIT)

S_E Sensor of entering corridor. Triggered by person / untriggered by control

CROSSING_E Model behavior: people do not rush to enter / system fast enough to change screen quickly

S_A

Set of arrival sensor where someone has arrived. Only one should be active at a time.

Making it a set and ensuring that it has cardinality ≤ 1 makes it possible to ensure this.

IN_CORRIDOR Human behavior: someone is in the corridor

INVARIANTSinv1: $SCREEN_CNT \in SCREEN$ inv9: $IN_CORRIDOR \in \{0, 1\}$ inv2: $CROSSING_E \in BOOL$ inv3: $S_E \in BOOL$ inv7: $S_A \subseteq COUNTERS$ inv_sens_arr: $card(S_A) \leq 1$ inv20: $IN_CORRIDOR = 1 \Rightarrow S_A = \emptyset$ inv_ent_grd: $S_E = TRUE \Rightarrow wait = FALSE$ inv_aux_ent_grd: $SCREEN_CNT = NOWAIT \Rightarrow wait = FALSE$ inv_grd_arr: $S_A \neq \emptyset \Rightarrow in_corridor > 0$ inv_aux_grd_arr: $(IN_CORRIDOR = 1 \wedge CROSSING_E = FALSE) \Rightarrow in_corridor = 1$ **EVENTS****Initialisation** ⟨extended⟩**begin**act1: $busy := \emptyset$ act2: $wait := FALSE$

It could be TRUE as well, and it will be updated by screen_num

act4: $next_counter \in COUNTERS$ act3: $in_corridor := 0$ act5: $SCREEN_CNT := NOWAIT$ act6: $S_E := FALSE$ act7: $CROSSING_E := FALSE$ act8: $S_A := \emptyset$ act9: $IN_CORRIDOR := 0$ **end****Event** enter_s ⟨ordinary⟩ $\hat{=}$ **when**grd1: $SCREEN_CNT = NOWAIT$ grd2: $CROSSING_E = FALSE$

A person does not attempt to enter while another is entering

thenact1: $CROSSING_E := TRUE$

A person is entering

act2: $S_E := TRUE$ act3: $IN_CORRIDOR := IN_CORRIDOR + 1$

Can we already say it's in the corridor?

end**Event** enter ⟨ordinary⟩ $\hat{=}$

```

refines enter
  when
    grd2:  $S\_E = TRUE$ 
    We only check the control signal!
  then
    act1:  $in\_corridor := in\_corridor + 1$ 
    act2:  $wait := TRUE$ 
    act3:  $S\_E := FALSE$ 
    act4:  $CROSSING\_E := FALSE$ 
    Assume: system fast enough that when this is changed,
    an unnoticeable amount of time has lapsed, and that a normal
    person behavior will not try to enter the corridor before
    this happens. Otherwise, barriers have to be installed.
    act5:  $SCREEN\_CNT := WAIT$ 
  end
Event arrive_s  $\langle ordinary \rangle \hat{=}$ 
  when
    grd1:  $IN\_CORRIDOR > 0$ 
    A person in the corridor. There must be a free counter!
    grd2:  $CROSSING\_E = FALSE$ 
    We cannot see a person in the counter if she has not crossed the entrance
  then
    act2:  $IN\_CORRIDOR := IN\_CORRIDOR - 1$ 
    act3:  $S\_A := S\_A \cup \{next\_counter\}$ 
    Sensor activated
  end
Event arrive  $\langle ordinary \rangle \hat{=}$ 
refines arrive
  when
    grd1:  $next\_counter \in S\_A$ 
    screen_num  $\wedge$  busy inv.
  then
    act1:  $in\_corridor := in\_corridor - 1$ 
    act2:  $busy := busy \cup \{next\_counter\}$ 
    act3:  $S\_A := S\_A \setminus \{next\_counter\}$ 
  end
Event screen_num  $\langle ordinary \rangle \hat{=}$ 
refines screen_num
  when
    grd2:  $COUNTERS \neq busy$ 
    grd3:  $in\_corridor = 0$ 
    Check how this impacts convergence, invariants
    grd4:  $wait = TRUE$ 
  then
    act1:  $next\_counter := COUNTERS \setminus busy$ 
    Wat to display
    act2:  $wait := FALSE$ 
    act3:  $SCREEN\_CNT := NOWAIT$ 
    Display number
  end
Event leave  $\langle ordinary \rangle \hat{=}$ 
  This is quite straightforward,
  so we are not refining it to add the sensor
extends leave
  any
     $c$ 
  where
    grd2:  $c \in busy$ 

```

```
    then
      act1: busy := busy \ {c}
    end
  END
```