

CONTEXT TrafficLights.ctx

SETS

LIGHTS

DIRECTION

CONSTANTS

Red

Green

Amber

CONFLICT

AXIOMS

axm1 : $LIGHTS = \{Red, Green, Amber\}$

axm2 : $Red \neq Green$

axm3 : $Red \neq Amber$

axm4 : $Green \neq Amber$

axm5 : $finite(DIRECTION)$

DIRECTION is a finite set of directions

axm6 : $CONFLICT \in DIRECTION \leftrightarrow DIRECTION$

CONFLICT relates conflicting directions

axm7 : $CONFLICT \cap id(DIRECTION) = \emptyset$

a direction cannot conflict with itself

axm8 : $CONFLICT^{-1} = CONFLICT$

conflicts are symmetric

END

An Event-B Specification of ChangeLight
Generated Date: 6 Oct 2008 @ 09:37:16 AM

MACHINE ChangeLight

SEES TrafficLights.ctx

VARIABLES

lights

INVARIANTS

inv1 : $lights \in DIRECTION \rightarrow \{Red, Green\}$

inv2 : $\forall d \cdot d \in DIRECTION \wedge lights(d) = Green$
 $\Rightarrow lights[CONFLICT[\{d\}]] \subseteq \{Red\}$

Safety

EVENTS

Initialisation

begin

act1 : $lights : |lights' \in DIRECTION \rightarrow \{Red, Green\}$
 $\wedge (\forall d \cdot d \in DIRECTION \wedge lights'(d) = Green$
 $\Rightarrow lights'[CONFLICT[\{d\}]] \subseteq \{Red\})$

end

Event ToGreen $\hat{=}$

any

gdir

where

grd1 : $gdir \in DIRECTION$

then

act1 : $lights := lights \Leftarrow (CONFLICT[\{gdir\}] \times \{Red\}) \Leftarrow \{gdir \mapsto Green\}$

end

Event ToRed $\hat{=}$

any

rdir

where

grd1 : $rdir \in DIRECTION$

then

act1 : $lights(rdir) := Red$

end

END

MACHINE ChangeLightR1

REFINES ChangeLight

SEES TrafficLights.ctx

VARIABLES

xlights Extended lights, Red, Green and Amber lights
delay delay between Amber and Red or Red and Green
adir
togreen
tored

INVARIANTS

inv1 : *adir* ∈ *DIRECTION*
inv2 : *togreen* ∈ *BOOL*
inv3 : *tored* ∈ *BOOL*
inv4 : *togreen* = *TRUE* ⇒ *tored* = *FALSE*
inv5 : *xlights* ∈ *DIRECTION* → *LIGHTS*
inv6 : ∀*d*.*d* ∈ *DIRECTION* ∧ *xlights*[{*d*}] ⊆ {*Green*, *Amber*}
 ⇒ *xlights*[*CONFLICT*[{*d*]}] ⊆ {*Red*}
inv7 : *togreen* = *TRUE* ⇒ *CONFLICT*[{*adir*}] ≪ (*lights* ≪ {*adir* ↦ *Green*}) = *CONFLICT*[{*adir*}] ≪
 (*xlights* ≪ {*adir* ↦ *Green*})
inv8 : *togreen* = *TRUE* ⇒ (*xlights*(*adir*) = *Green* ⇒ *lights* = *xlights*)
inv9 : *delay* ⊆ *DIRECTION*
inv10 : *tored* = *TRUE* ⇒ (*xlights* ≪ {*adir* ↦ *Red*} = *lights* ≪ {*adir* ↦ *Red*})
inv11 : *togreen* = *FALSE* ∧ *tored* = *FALSE* ⇒ *lights* = *xlights*

EVENTS

Initialisation

begin
 with
 lights' : *lights'* = *xlights'*
 act1 : *xlights* : |*xlights'* ∈ *DIRECTION* → {*Red*, *Green*}
 ∧ (∀*d*.*d* ∈ *DIRECTION* ∧ *xlights'*(*d*) = *Green*
 ⇒ *xlights'*[*CONFLICT*[{*d*]}] ⊆ {*Red*})
 act2 : *delay* := ∅
 act3 : *togreen* := *FALSE*
 act4 : *tored* := *FALSE*
 act5 : *adir* := *DIRECTION*

end

Event ToGreen ≐

Refines ToGreen

when
 grd1 : *togreen* = *TRUE*

$grd2 : xlights(adir) = Red$
 $grd3 : xlights[CONFLICT[\{adir\}]] \subseteq \{Red\}$
 $grd4 : adir \notin delay$

with

$gdir : gdir = adir$

then

$act1 : xlights(adir) := Green$

$act2 : togreen := FALSE$

end

Event ToGreenInit $\hat{=}$

Refines ToGreen

any

$gdir$

where

$grd1 : gdir \in DIRECTION$

$grd2 : togreen = FALSE$

$grd3 : tored = FALSE$

$grd4 : xlights(gdir) = Red$

then

$act1 : adir := gdir$

$act2 : togreen := TRUE$

end

Event ToGreenAmber $\hat{=}$

Which is convergent

any

dir

where

$grd1 : togreen = TRUE$

$grd2 : dir \in CONFLICT[\{adir\}]$

$grd3 : xlights(dir) = Green$

then

$act1 : xlights(dir) := Amber$

$act2 : delay := delay \cup \{dir\}$

end

Event ToGreenRed $\hat{=}$

Which is convergent

any

dir

where

$grd1 : togreen = TRUE$

$grd2 : dir \in DIRECTION$

$grd3 : dir \in CONFLICT[\{adir\}]$

$grd4 : xlights(dir) = Amber$

$grd5 : dir \notin delay$
 $grd6 : xlights(adir) \neq Green$
then
 $act1 : xlights(dir) := Red$
 $act2 : delay := delay \cup \{adir\}$
end

Event Delay $\hat{=}$

Which is convergent

any
 dir
where
 $grd3 : dir \in delay$
then
 $act1 : delay := delay \setminus \{dir\}$
end

Event ToRed $\hat{=}$

Refines ToRed

when
 $grd1 : tored = TRUE$
 $grd2 : xlights(adir) = Amber$
 $grd3 : adir \notin delay$
with
 $rdir : rdir = adir$
then
 $act1 : xlights(adir) := Red$
 $act2 : tored := FALSE$
end

Event ToRedInit $\hat{=}$

Refines ToRed

any
 $rdir$
where
 $grd1 : rdir \in DIRECTION$
 $grd2 : xlights(rdir) = Green$
 $grd3 : tored = FALSE$
 $grd4 : togreen = FALSE$
then
 $act1 : adir := rdir$
 $act2 : tored := TRUE$
end

Event TeRedAmber $\hat{=}$

when

```
    grd1 : tored = TRUE
    grd2 : xlights(adir) = Green
  then
    act1 : xlights(adir) := Amber
    act2 : delay := delay ∪ {adir}
  end
VARIANT
END
```