

System Modelling and Design

Traffic Lights

Revision: 1.2, May 19, 2004

Ken Robinson

16th March 2005

©Ken Robinson 2005

[mailto::k.robinson@unsw.edu.au](mailto:k.robinson@unsw.edu.au)

Contents

| | | |
|----------|--|----------|
| 1 | Objectives of this lecture | 2 |
| 2 | A simple 2-way intersection | 2 |
| 3 | The Context Machine | 2 |
| 4 | The SimpleTwoWay Machine | 2 |
| 5 | The Invariant | 3 |
| 5.1 | Animation | 3 |
| 5.2 | A Safe state | 3 |
| 5.3 | Further Animation | 4 |
| 5.4 | Strengthening the invariant | 4 |
| 5.5 | Other Invariants | 5 |
| 5.6 | Strengthening the Precondition | 5 |
| 5.7 | The SimpleTwoWay machine | 6 |
| 6 | Sequencing | 6 |
| 6.1 | The precondition of ToRed | 6 |
| 6.2 | The precondition of ToGreen | 6 |
| 6.3 | Equivalent predicates | 7 |
| 6.4 | The precondition of ToAmber | 7 |
| 7 | The TwoWay Machine | 8 |

1 Objectives of this lecture

- To explore the specification of some simple traffic light controllers.
- To explore the use of a state invariant to ensure safety.
- To explore the use of preconditions that ensure an operation will not violate the state invariant, when the precondition is satisfied.
- To use the animator to illustrate these explorations.

2 A simple 2-way intersection

Consider traffic lights at the intersection of two roads, one running *North-South* and the other *East-West*.

There are four sets of lights, each capable of showing **Red**, **Green** and **Amber**, placed at *North*, *East*, *South* and *West* positions.

The *North* and *South* lights are always identical, as are the *East* and *West* lights.

There are no *right-turn* lights.

Lights should change in the sequence:

Red \longrightarrow **Green** \longrightarrow **Amber** \longrightarrow **Red** \longrightarrow ...

We wish to specify a traffic light controller that ensures safety and the correct sequencing.

3 The Context Machine

We will introduce a context machine containing the enumerated sets *DIRECTION* and *LIGHT*.

We will also specify a constant (function) *OTHER_DIR* that maps each direction into the other direction.

Context machines are used commonly and frequently in *B Method (B)* to define *sets* and *constants*.

MACHINE *TwoWay_Ctx*

SETS

DIRECTION = { *NorthSouth* , *EastWest* } ;

LIGHT = { *Red* , *Green* , *Amber* }

CONSTANTS *OTHER_DIR*

PROPERTIES

OTHER_DIR \in *DIRECTION* \rightarrow *DIRECTION* \wedge

OTHER_DIR = { *NorthSouth* \mapsto *EastWest* , *EastWest* \mapsto *NorthSouth* }

END

4 The SimpleTwoWay Machine

1. We will write a basic machine with no non-trivial state invariant and no non-trivial preconditions.

2. The machine will see the context machine and have a state of one variable, *lights*, which is a total function from *DIRECTION* to *LIGHT*.
3. There is one operation: *ChangeLight(dir, colour)* that changes the light to *colour* in the direction *dir*

MACHINE *SimpleTwoWay0*

SEES *TwoWay_Ctx*

VARIABLES *lights*

INVARIANT $lights \in DIRECTION \rightarrow LIGHT$

INITIALISATION $lights := \{ NorthSouth \mapsto Red, EastWest \mapsto Red \}$

OPERATIONS

ChangeLight (*dir*, *colour*) $\hat{=}$

PRE $dir \in DIRECTION \wedge colour \in LIGHT$

THEN $lights (dir) := colour$

END

END

5 The Invariant

5.1 Animation

1. Use the animator to explore the behaviour. When animating, choose to display the invariant, normally turned off.
2. It is, of course, trivial to establish that the controller is unsafe.
3. Try to formulate an invariant that will ensure safety.
4. Whenever the state is unsafe, the invariant must be *false*.

$$\neg(\text{safe}) \Rightarrow \neg(\text{invariant}) \quad (1)$$

5. Conversely, whenever the invariant is *true*, the state should be safe.

$$\text{invariant} \Rightarrow \text{safe} \quad (2)$$

6. Of course, 1 and 2 are equivalent; one is the *contrapositive* of the other.
7. If we have the *weakest* invariant, then whenever the state is safe, the invariant will be *true*.
8. Find adequately strong preconditions.

5.2 A Safe state

The state invariant should be:

1. *false* for all unsafe states *true* for all safe states
2. An initial attempt at an invariant might be

$$\neg (\text{lights}(\text{NorthSouth}) = \text{Green} \wedge \text{lights}(\text{EastWest}) = \text{Green})$$

which, using the predicate identities

$$\neg (P \wedge Q) \equiv \neg P \vee \neg Q \equiv P \Rightarrow \neg Q ,$$

may be written

$$\text{lights}(\text{NorthSouth}) = \text{Green} \Rightarrow \neg (\text{lights}(\text{EastWest}) = \text{Green})$$

To avoid an error during analysis caused by the left associativity of operators in B, the above implication must be parenthesised.

5.3 Further Animation

Further Animation

Try animating with the invariant on the preceding slide, using the animation script on the following slide.

```

ANIMATE
  SimpleTwoWay0.test1.anm
PARAMETER_VALUES
  ?
SETS_VALUES
  ?
CONSTANTS_VALUES
  OTHER_DIR = {NorthSouth |-> EastWest , EastWest |-> NorthSouth}
ENUM_SETS_VALUES
  DIRECTION = {NorthSouth , EastWest};
  LIGHT = {Red , Green , Amber}
OPERATIONS
  INI_SimpleTwoWay0;
  ChangeLight (NorthSouth, Green);
  ChangeLight (EastWest, Green);
  undo;
  ChangeLight (EastWest, Amber);
  undo;
  ChangeLight (NorthSouth, Amber);
  ChangeLight (EastWest, Amber)
END

```

5.4 Strengthening the invariant

Clearly, the light in both directions cannot be either **Green** or **Amber**.

| | Red | Green | Amber |
|-------|------|--------|--------|
| Red | safe | safe | safe |
| Green | safe | unsafe | unsafe |
| Amber | safe | unsafe | unsafe |

This leads to the invariant:

$$\begin{aligned}
& \neg (\text{lights}(\text{NorthSouth}) \in \{\text{Green}, \text{Amber}\} \wedge \\
& \text{lights}(\text{EastWest}) \in \{\text{Green}, \text{Amber}\}) \\
& \equiv \\
& (\text{lights}(\text{NorthSouth}) \in \{\text{Green}, \text{Amber}\} \Rightarrow \\
& \neg (\text{lights}(\text{EastWest}) \in \{\text{Green}, \text{Amber}\})) \\
& \equiv \\
& (\text{lights}(\text{NorthSouth}) \in \{\text{Green}, \text{Amber}\} \Rightarrow \\
& \text{lights}(\text{EastWest}) = \text{Red})
\end{aligned}$$

5.5 Other Invariants

There are other invariants that adequately express safety for a two-way intersection:

$$\text{lights}(\text{NorthSouth}) = \text{Red} \vee \text{lights}(\text{EastWest}) = \text{Red}$$

$$\text{Red} \in \text{ran}(\text{lights})$$

But these conditions do not generalise to intersections with more than two ways. Indeed the expression of the invariant that best generalises is

$$\forall \text{dir}. (\text{dir} \in \text{DIRECTION} \wedge \text{lights}(\text{dir}) \in \{\text{Green}, \text{Amber}\} \Rightarrow \\
\text{lights}(\text{OTHER_DIR}(\text{dir})) = \text{Red})$$

5.6 Strengthening the Precondition

We now have states and arguments of the operation *ChangeLight* that lead to a state that violates the state invariant.

This is not satisfactory!

The preconditions need to be strengthened so that the post-state violates the invariant *only if* the precondition is *false*.

This illustrates how preconditions and invariants collaborate in achieving safety.

Notice that preconditions ensure safety by imposing a proof obligation to be discharged in any context in which the operation is used.

A possible precondition is

$$\begin{aligned}
& (\text{colour} \in \{\text{Green}, \text{Amber}\} \Rightarrow \\
& \text{lights}(\text{OTHER_DIR}(\text{dir})) = \text{Red})
\end{aligned}$$

5.7 The SimpleTwoWay machine

MACHINE *SimpleTwoWay*

SEES *TwoWay_Ctx*

VARIABLES *lights*

INVARIANT

$lights \in DIRECTION \rightarrow LIGHT \wedge$

$(lights (NorthSouth) \in \{ Green, Amber \} \Rightarrow lights (EastWest) = Red)$

INITIALISATION

$lights := \{ NorthSouth \mapsto Red, EastWest \mapsto Red \}$

OPERATIONS

ChangeLight (*dir*, *colour*) $\hat{=}$

PRE $dir \in DIRECTION \wedge colour \in LIGHT \wedge$

$(colour \in \{ Green, Amber \} \Rightarrow lights (OTHER_DIR (dir)) = Red)$

THEN $lights (dir) := colour$

END

END

6 Sequencing

The *SimpleTwoWay* machine ensures safety, but does not enforce any sequencing.

To achieve the desired sequencing we will introduce a new machine: *TwoWay*, that *INCLUDES SimpleTwoWay* and has three operations *ToRed*(*dir*), *ToGreen*(*dir*), and *ToAmber*(*dir*) for changing the lights.

The notion is that the body of each operation will use the operation *ChangeLight*(*dir*, *colour*) to change the colour.

The precondition of each operation will constrain the sequencing, and also must ensure that the precondition of *ChangeLight* is satisfied.

6.1 The precondition of ToRed

Since the precondition of *ChangeLight* is always satisfied for the colour **Red**, we need to only be concerned with sequencing.

Hence the precondition is

$$lights(dir) = Amber$$

6.2 The precondition of ToGreen

When setting a light to **Green** the precondition of *ChangeLight* requires

$$lights(OTHER_DIR(dir)) = Red$$

Sequencing requires

$$\mathit{lights}(\mathit{dir}) = \mathit{Red}$$

Thus the precondition is the conjunction

$$\mathit{lights}(\mathit{dir}) = \mathit{Red} \wedge \mathit{lights}(\mathit{OTHER_DIR}(\mathit{dir})) = \mathit{Red}$$

6.3 Equivalent predicates

This may be expressed by a number of equivalent predicates, when combined with the sequencing predicate $\mathit{lights}(\mathit{dir}) = \mathit{Red}$:

1. $\mathit{dir} = \mathit{NorthSouth} \Rightarrow \mathit{lights}(\mathit{EastWest}) = \mathit{Red} \wedge$
 $\mathit{dir} = \mathit{EastWest} \Rightarrow \mathit{lights}(\mathit{NorthSouth}) = \mathit{Red}$
2. $\forall \mathit{dir} . (\mathit{dir} \in \mathit{DIRECTION} \Rightarrow \mathit{lights}(\mathit{dir}) = \mathit{Red})$
3. $\mathit{lights}[\mathit{DIRECTION}] = \{\mathit{Red}\}$ ¹
4. $\mathit{ran}(\mathit{lights}) = \{\mathit{Red}\}$

It is worth noting that alternatives 2, 3 and 4 if generalized to more directions, are too strong.

6.4 The precondition of ToAmber

In order to satisfy the precondition of *ChangeLight*, the light in the other direction must be showing **Red**.

But, since the sequencing condition is

$$\mathit{lights}(\mathit{dir}) = \mathit{Green},$$

the state invariant implies that

$$\mathit{lights}(\mathit{OTHER_DIR}(\mathit{dir})) = \mathit{Red} .$$

So, only the sequencing condition is required

$$\mathit{lights}(\mathit{dir}) = \mathit{Green}$$

¹This uses *relational image* $r[s] = \mathit{ran}(s \triangleleft r)$

7 The TwoWay Machine

MACHINE *TwoWay*
SEES *TwoWay_Ctx*
INCLUDES *SimpleTwoWay*

OPERATIONS

ToRed (*dir*) $\hat{=}$
 PRE $dir \in DIRECTION \wedge lights (dir) = Amber$
 THEN *ChangeLight (dir , Red)*
 END ;

ToGreen (*dir*) $\hat{=}$
 PRE $dir \in DIRECTION \wedge lights (dir) = Red \wedge$
 $lights (OTHER_DIR (dir)) = Red$
 THEN *ChangeLight (dir , Green)*
 END ;

ToAmber (*dir*) $\hat{=}$
 PRE $dir \in DIRECTION \wedge lights (dir) = Green$
 THEN *ChangeLight (dir , Amber)*
 END

END