2. Controlling Cars on a Bridge

Jean-Raymond Abrial

April 2008

Purpose of this Lecture (1)

- To present an example of system development

- Our approach: a series of more and more accurate models

- This approach is called refinement

- The models formalize the view of an external observer

Purpose of this Lecture (2)

- Each model will be analyzed and proved to be correct

- The aim is to obtain a system that will be correct by construction

- The correctness criterions correspond to a number of proof rules

- Proofs will be performed by using the sequent calculus

- Inference rules used in the sequent calculus will be reviewed

What you will Learn

- The concepts of state and events for defining models

- A few principles of system development: essentially refinement

- A refresher on classical logic and simple arithmetic foundations

- A refresher on formal proofs

Outline

1. Presenting the requirement document (as in previous lecture)

2. Defining the strategy

3. Development of the initial model and the refinements

- Remark: During the development some theoretical background will be provided

A Requirements Document (1)

- The system we are going to build is a piece of software connected to some equipment.

- There are two kinds of requirements:
  - those concerned with the equipment, labeled EQP,
  - those concerned with the function of the system, labeled FUN.

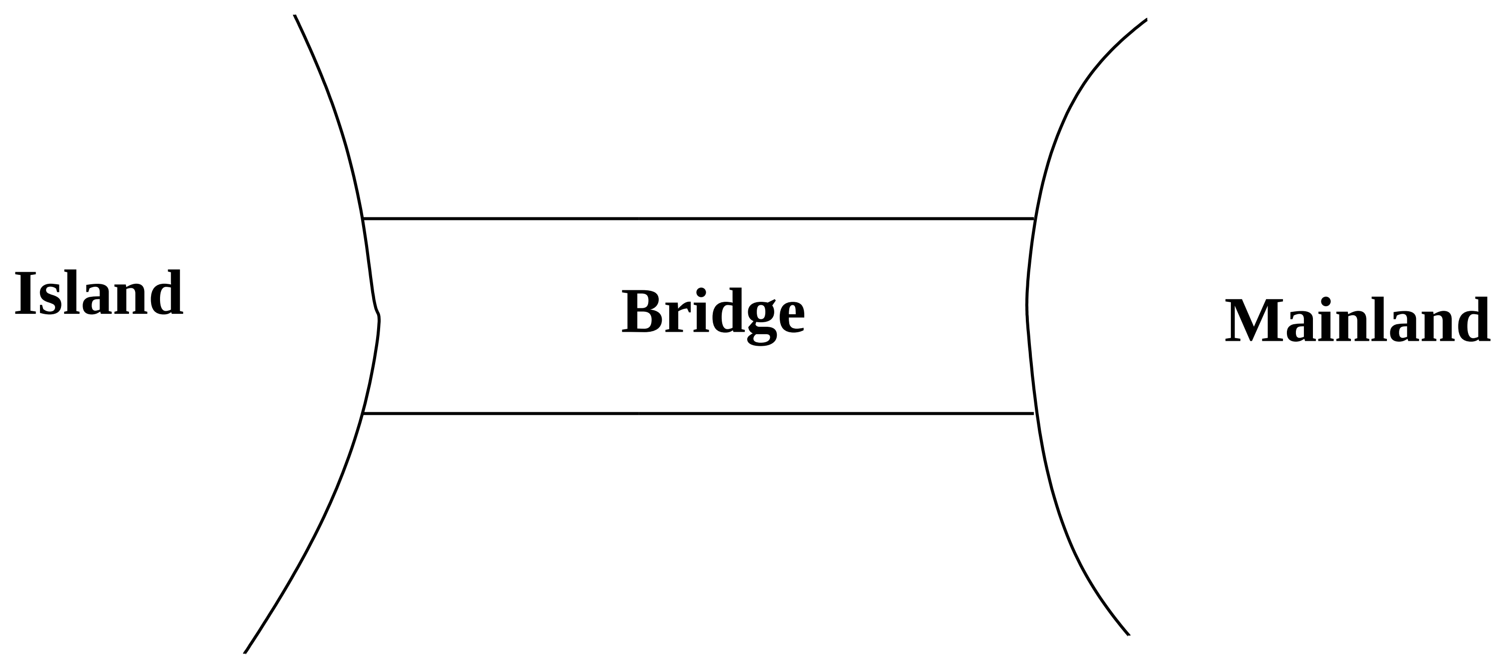
- The function of this system is to control cars on a narrow bridge.

- This bridge is supposed to link the mainland to a small island.

A Requirements Document (2)

The system is controlling cars on a bridge between the mainland and an island	FUN-1

- This can be illustrated as follows



A Requirements Document (3)

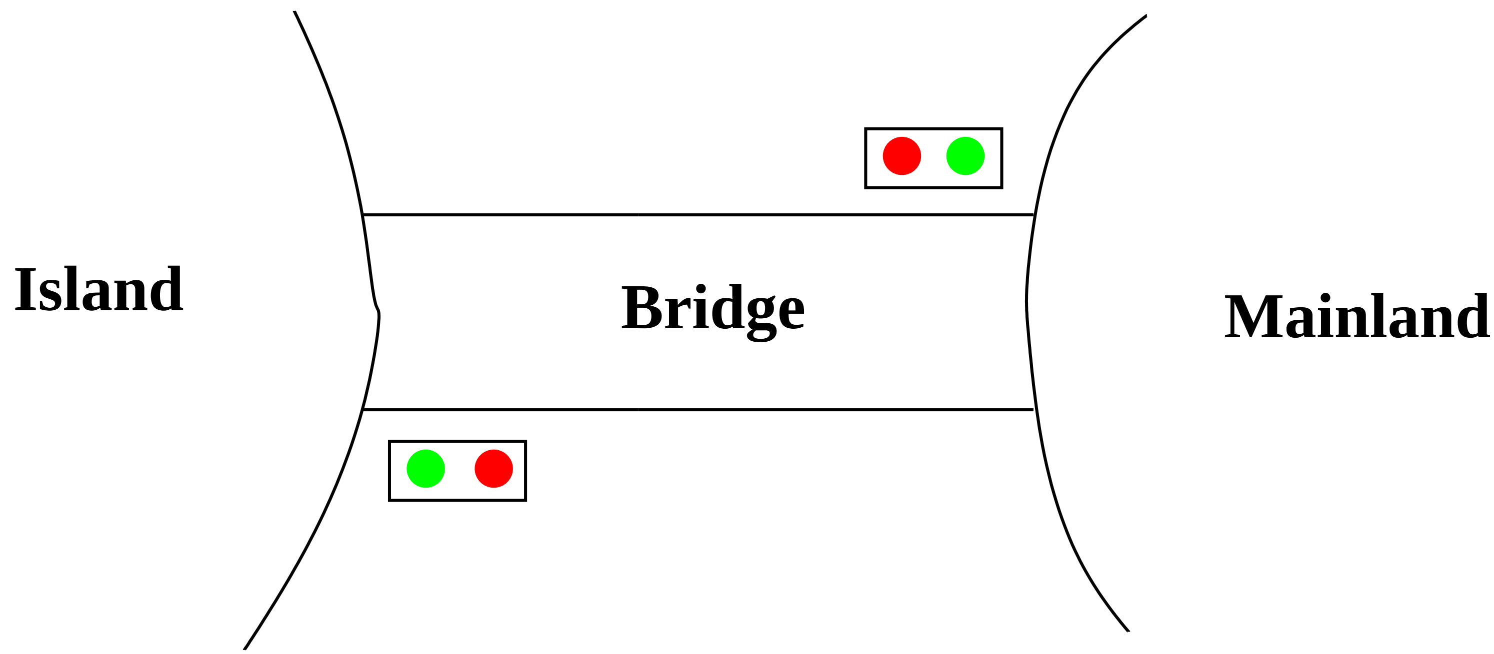
- The controller is equipped with two traffic lights with two colors.

The system has two traffic lights with two colors: green and red	EQP-1

A Requirements Document (4)

- One of the traffic lights is situated on the mainland and the other one on the island. Both are close to the bridge.

- This can be illustrated as follows



The traffic lights control the entrance to the bridge at both ends of it	EQP-2

- Drivers are supposed to obey the traffic light by not passing when a traffic light is red.

Cars are not supposed to pass on a red traffic light, only on a green one	EQP-3

A Requirements Document (6)

- There are also some car sensors situated at both ends of the bridge.

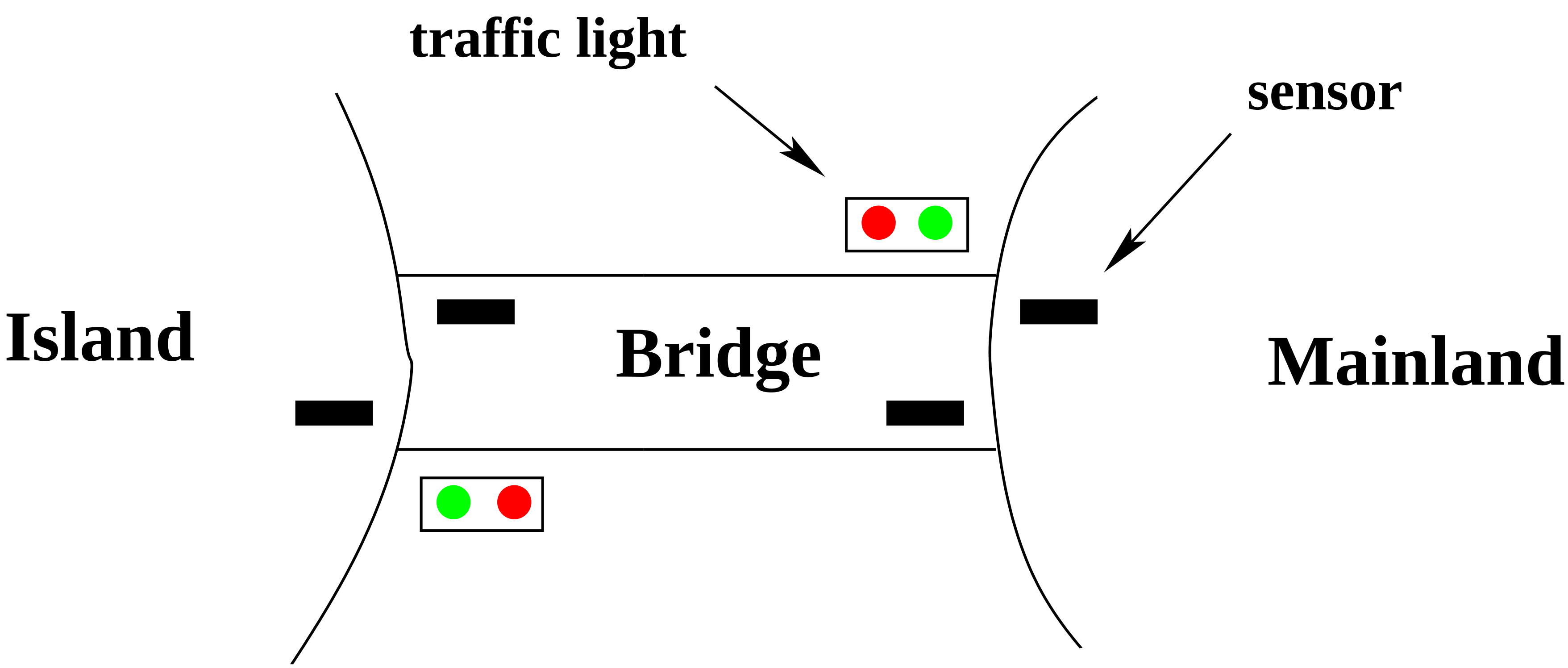
- These sensors are supposed to detect the presence of cars intending to enter or leave the bridge.

- There are four such sensors. Two of them are situated on the bridge and the other two are situated on the mainland and on the island.

The system is equipped with four car sensors each with two states: on or off	EQP-4

The sensors are used to detect the presence of cars entering or leaving the bridge	EQP-5

- The pieces of equipment can be illustrated as follows:



A Requirements Document (8)

- This system has two main constraints: the number of cars on the bridge and the island is limited and the bridge is one way.

The number of cars on the bridge and the island is limited	FUN-2

The bridge is one way or the other, not both at the same time	FUN-3

The Reference Document (1)

The system is controlling cars on a bridge between the mainland and an island	FUN-1

The number of cars on the bridge and the island is limited	FUN-2

The bridge is one way or the other, not both at the same time	FUN-3

The Reference Document (2)

The system has two traffic lights with two colors: green and red	EQP-1

The traffic lights control the entrance to the bridge at both ends of it	EQP-2

Cars are not supposed to pass on a red traffic light, only on a green one	EQP-3

The Reference Document (3)

The system is equipped with four car sensors each with two states: on or off	EQP-4

The sensors are used to detect the presence of cars entering or leaving the bridge	EQP-5

Our Development Approach

- Initial model: Limiting the number of cars (FUN_2)

- First refinement: Introducing the one way bridge (FUN_3)

- Second refinement: Introducing the traffic lights (EQP_1,2,3)

- Third refinement: Introducing the sensors (EQP_4,5)

Initial Model

- It is very simple

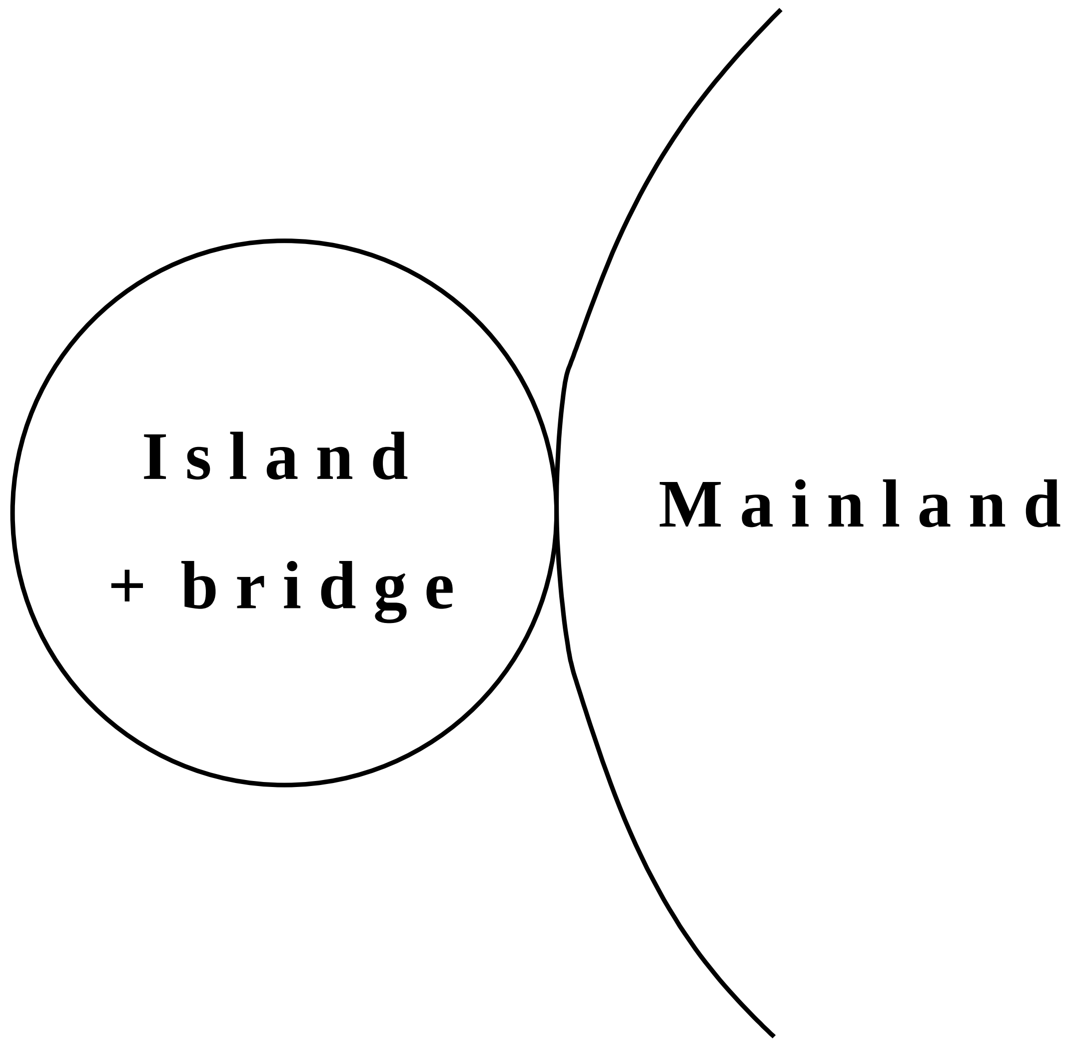
- We do not consider at all the equipment: traffic lights and sensors

- We even do not consider the bridge

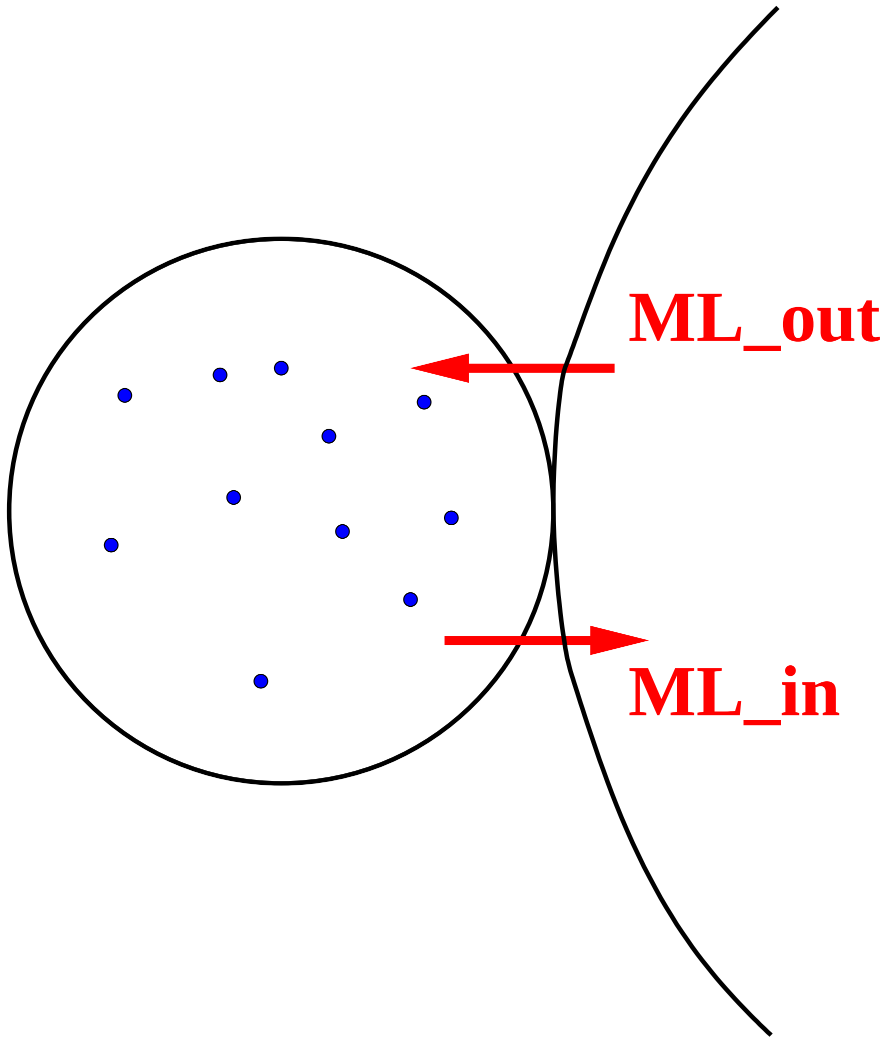
- We are just interested in the pair “island-bridge”

- We are focusing on FUN-2 (limited number of cars on island-bridge)

A Situation as Seen from the Sky



Two Events which can be Observed



- STATIC PART of the state: constant d with axiom **axm0_1**

constant: d	**axm0_1:** $d \in \mathbb{N}$

- d is the maximum number of cars allowed in the Island-Bridge

- **axm0_1** states that d is a natural number

- Constant d is a member of the set $\mathbb{N} = \{0, 1, 2, , \ldots\}$

- DYNAMIC PART: variable v with invariants **inv0_1** and **inv0_2**

variable: n

inv0_1: $n \in \mathbb{N}$

inv0_2: $n \leq d$

- n is the effective number of cars in the Island-Bridge

- n is a natural number (**inv0_1**)

- n is always smaller than or equal to d (**inv0_2**): this is FUN_2

- Labels **axm0_1**, **inv0_1**, etc. have been chosen systematically

- **axm** or **inv** recall the purpose: axioms of constants or invariants of variables

- The **0** as in **inv0_1** stands for the initial model.

- Later we shall have **inv1_1** for an invariant of refinement **1**, etc.

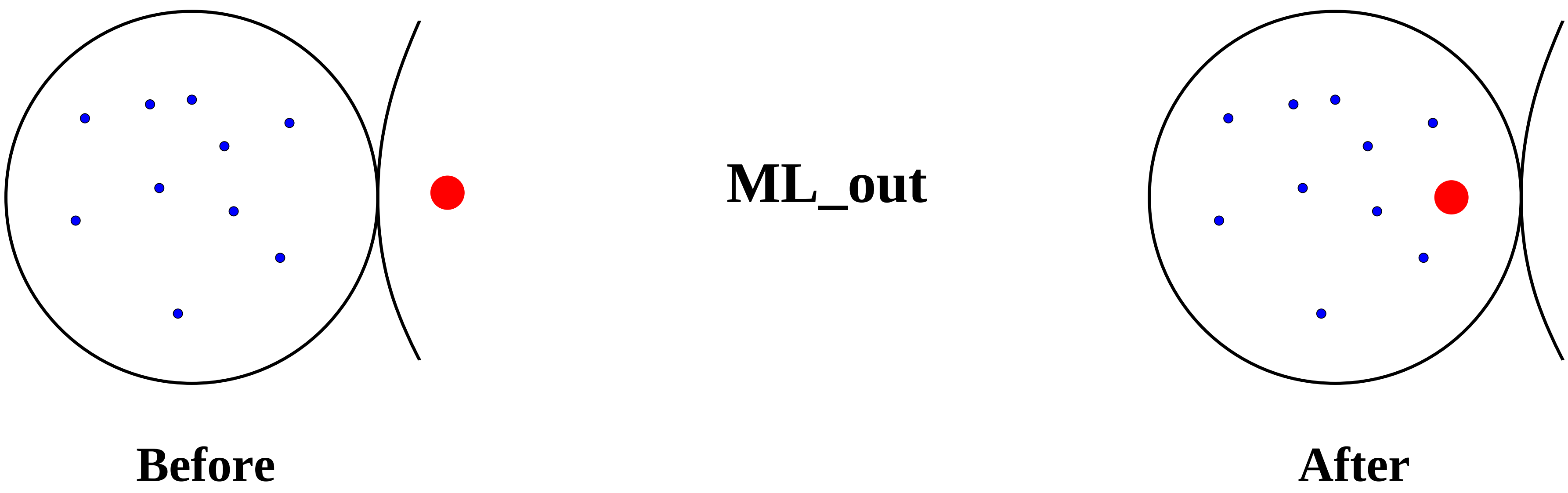
- The **1** as in **inv0_1** is a serial number

- Any convention is valid as long as it is systematic

Event ML_out

- This is the first transition (or event) that can be observed

- A car is leaving the mainland and entering the Island-Bridge

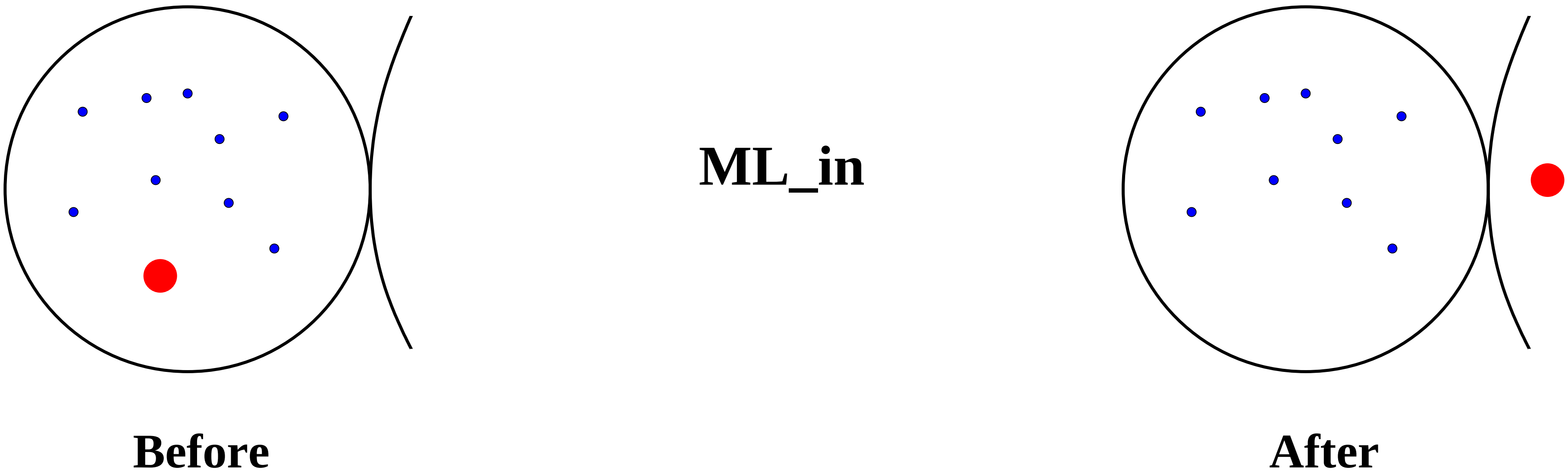


- The number of cars in the Island-Bridge has been incremented

Event ML_in

- We can also observe a second transition (or event)

- A car leaving the Island-Bridge and re-entering the mainland



- The number of cars in the Island-Bridge has been decremented

Formalizing the two Events: an Approximation

- Event ML_out increments the number of cars

ML_out

$$n := n + 1$$

- Event ML_in decrements the number of cars

ML_in

$$n := n - 1$$

- An event is denoted by its name and its action (an assignment)

An Approximation: Why?

- These events are approximations for two reasons:

1. They might be refined (made more precise) later

2. They might be insufficient at this stage because not consistent with the invariant

- We shall have to perform a proof in order to check this consistency

Towards the Proof: Before-after Predicates

- To each event can be associated a before-after predicate

- It denotes the relationship holding between the values of the variable *just before* and *just after* the event occurrence

- The before-value is denoted by the variable name, say n

- The after-value is denoted by the variable name *primed*, say n'

The Events

ML_out $n := n + 1$	**ML_in** $n := n - 1$

The corresponding before-after predicates

$n' = n + 1$	$n' = n - 1$

These representations are equivalent

About the Shape of the Before-after Predicates

- The before-after predicates we have shown are very simple

$$n' = n + 1 \qquad\qquad n' = n - 1$$

- The after-value n' is defined as a function of the before-value n

- This is because the corresponding events are deterministic

- In further lectures, we shall consider some non-deterministic events:

$$n' \in \{n + 1, n + 2\}$$

Proving Invariant Preservation: the Problem

- When we wrote events ML_out and ML_in we did not take into accounts invariants **inv0_1** and **inv0_2**

- There is no reason a priori for these invariants to be preserved

- This has to be proved rigorously

- We have to clarify what we have to prove

- We are going to define some proof obligations also called verification conditions

Intuition about Invariant Preservation

- Let us consider invariant **inv0_1**

$$n \in \mathbb{N}$$

- And let us consider event ML_out with before-after predicate

$$n' = n + 1$$

- Preserving **inv0_1** means that we have (just after ML_out):

$$n' \in \mathbb{N} \quad \text{that is} \quad n + 1 \in \mathbb{N}$$

Being more Precise

- Under hypothesis $\boldsymbol{n} \in \mathbb{N}$ then $\boldsymbol{n} + \mathbf{1} \in \mathbb{N}$ is provable

- This can be written as follows

$$\boldsymbol{n} \in \mathbb{N} \quad \vdash \quad \boldsymbol{n} + \mathbf{1} \in \mathbb{N}$$

- This statement is called a sequent (more later)

- This can be generalized (next slide)

Invariant Preservation Proof Obligations (1)

- We collectively denote our set of constants by c

- We denote our set of axioms by $A(c)$: $A_1(c), A_2(c), \ldots$

- We collectively denote our set of variables by v

- We denote our set of invariants by $I(c, v)$: $I_1(c, v), I_2(c, v), \ldots$

Invariant Preservation Proof Obligations (2)

- We are given an event with before-after predicate $v' = E(c, v)$

- We have to prove the following in order to preserve invariant $I_i(c, v)$:

$A(c),\ I(c, v) \ \vdash \ I_i(c, E(c, v))$	INV

- It says: prove $I_i(c, E(c, v))$ under hypotheses $A(c)$ and $I(c, v)$

- We have given the name INV to this proof obligation

$A(c),\ I(c,v) \ \vdash \ I_i(c,E(c,v))$	INV

- Just before occurrence of the event represented by $v' = E(c,v)$, $A(c)$ as well as $I(c,v)$ clearly holds. We can then assume them

- Just after the occurrence, invariant $I_i(c,v)$ has become $I_i(c,v')$, that is $I_i(c,E(c,v))$

- This statement $I_i(c,E(c,v))$ must hold since we claimed that $I_i(c,v)$ was an invariant

Vertical Layout of the Invariant Rule

- The proof obligation

$A(c),\; I(c,v) \;\;\vdash\;\; I_i(c, E(c,v))$	INV

can be re-written vertically as follows:

Axioms Invariants $\vdash$ Modified Invariant	$A(c)$ $I(c,v)$ $\vdash$ $I_i(c, E(c,v))$	INV

- A sequent is a formal statement of the following shape

$$\boxed{\mathbf{H} \;\vdash\; \mathbf{G}}$$

- The symbol "$\vdash$" is called the turnstyle (it corresponds to verb "yield")

- **H** denotes a set of predicates: the hypotheses (or assumptions)

- **G** denotes a predicate: the goal (or conclusion)

- It can be read: "Assumptions **H** yield conclusion **G**"

Back to our Example: What is to be Proved

- We have two events

ML_out
$$n := n + 1$$

ML_in
$$n := n - 1$$

- And two invariants

inv0_1: $n \in \mathbb{N}$

inv0_2: $n \leq d$

- Thus, four statements have to be proved

To be Proved for event ML_out and invariant inv0_1

ML_out

$n := n + 1$

$(n' = n + 1)$

Axiom **axm0_1**
Invariant **inv0_1**
Invariant **inv0_2**
$\vdash$
Modified Invariant **inv0_1**

$$
\begin{array}{l}
d \in \mathbb{N} \\
n \in \mathbb{N} \\
n \leq d \\
\vdash \\
n + 1 \in \mathbb{N}
\end{array}
$$

- This proof obligation is named: ML_out / **inv0_1** / INV

ML_out
$n := n + 1$

$(n' = n + 1)$

Axiom **axm0_1**
Invariant **inv0_1**
Invariant **inv0_2**
$\vdash$
Modified Invariant **inv0_2**

$$
\begin{array}{l}
d \in \mathbb{N} \\
n \in \mathbb{N} \\
n \leq d \\
\vdash \\
n + 1 \leq d
\end{array}
$$

- This proof obligation is named: ML_out / **inv0_2** / INV

To be Proved for event ML_in and invariant inv0_1

ML_in
$n := n - 1$

$(n' = n - 1)$

Axiom **axm0_1**
Invariant **inv0_1**
Invariant **inv0_2**
$\vdash$
Modified Invariant **inv0_1**

$$
\begin{array}{l}
d \in \mathbb{N} \\
n \in \mathbb{N} \\
n \leq d \\
\vdash \\
n - 1 \in \mathbb{N}
\end{array}
$$

- This proof obligation is named: ML_in / **inv0_1** / INV

ML_in
$n := n - 1$

$(n' = n - 1)$

Axiom **axm0_1**
Invariant **inv0_1**
Invariant **inv0_2**
$\vdash$
Modified Invariant **inv0_2**

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ \vdash \\ n - 1 \leq d \end{array}$$

- This proof obligation is named: ML_in / **inv0_2** / INV

Summary of What is to be Proved

ML_out / **inv0_1** / INV

$$
\begin{array}{l}
\quad d \in \mathbb{N} \\
\quad n \in \mathbb{N} \\
\quad n \leq d \\
\vdash \\
\quad n + 1 \in \mathbb{N}
\end{array}
$$

ML_out / **inv0_2** / INV

$$
\begin{array}{l}
\quad d \in \mathbb{N} \\
\quad n \in \mathbb{N} \\
\quad n \leq d \\
\vdash \\
\quad n + 1 \leq d
\end{array}
$$

ML_in / **inv0_1** / INV

$$
\begin{array}{l}
\quad d \in \mathbb{N} \\
\quad n \in \mathbb{N} \\
\quad n \leq d \\
\vdash \\
\quad n - 1 \in \mathbb{N}
\end{array}
$$

ML_in / **inv0_2** / INV

$$
\begin{array}{l}
\quad d \in \mathbb{N} \\
\quad n \in \mathbb{N} \\
\quad n \leq d \\
\vdash \\
\quad n - 1 \leq d
\end{array}
$$

$$\boxed{\begin{array}{ll} & d \in \mathbb{N} \\ & n \in \mathbb{N} \\ & n \leq d \\ \vdash & \\ & n+1 \in \mathbb{N} \end{array}} \quad \xrightarrow[\text{hypotheses}]{\text{remove}} \quad \boxed{\begin{array}{ll} & n \in \mathbb{N} \\ \vdash & \\ & n+1 \in \mathbb{N} \end{array}} \quad \text{obvious}$$

- In the first step, we remove some irrelevant hypotheses

- In the second and final step, we accept the sequent as it is

Being more Precise

- In the previous slide, we have applied some implicit rules:
    - For simplifying given sequents
    - For asserting that a sequent is proved without justifications

- In order to be very precise, we make these rules more explicit

- Such rules are called rules of inference

Rule of Inference: Monotonicity of Hypotheses

- The first rule of inference can be stated as follows:

$$\frac{\mathbf{H1} \ \vdash \ \mathbf{G}}{\mathbf{H1}, \mathbf{H2} \ \vdash \ \mathbf{G}} \quad \mathbf{MON}$$

- Above the horizontal line, we have a sequent called the antecedent (this predicate can be missing or on the contrary made multiple)

- Below the horizontal line we have a sequent called the consequent

- To prove the consequent, it is sufficient to prove the antecedent

- The Second Peano Axiom

$$\frac{}{\mathbf{n} \in \mathbb{N} \vdash \mathbf{n}+1 \in \mathbb{N}} \quad \mathbf{P2}$$

$$\frac{}{0 < \mathbf{n} \vdash \mathbf{n}-1 \in \mathbb{N}} \quad \mathbf{P2'}$$

- Such rules are called axioms (because they have no antecedents)

More Arithmetic Rules of Inference

- Such rules are given here without demonstration

$$\frac{}{\mathbf{n} < \mathbf{m} \;\vdash\; \mathbf{n} + 1 \leq \mathbf{m}} \; \textbf{INC}$$

$$\frac{}{\mathbf{n} \leq \mathbf{m} \;\vdash\; \mathbf{n} - 1 \leq \mathbf{m}} \; \textbf{DEC}$$

- This rule denotes the 2nd Peano axiom in very general terms:

$$\boxed{\dfrac{}{\mathbf{n} \in \mathbb{N} \;\vdash\; \mathbf{n} + 1 \in \mathbb{N}} \quad \textbf{P2}}$$

- It is a rule schema where **n** is called a meta-variable

- It can be applied to the following sequent by pattern matching:

$$\boldsymbol{a} + \boldsymbol{b} \in \mathbb{N} \;\vdash\; \boldsymbol{a} + \boldsymbol{b} + \mathbf{1} \in \mathbb{N}$$

- A proof is (for the moment) just a sequence of sequents

$$\boxed{\begin{array}{l} \quad d \in \mathbb{N} \\ \quad n \in \mathbb{N} \\ \quad n \leq d \\ \vdash \\ \quad n+1 \in \mathbb{N} \end{array}} \quad \text{MON} \quad \boxed{\begin{array}{l} \quad n \in \mathbb{N} \\ \vdash \\ \quad n+1 \in \mathbb{N} \end{array}} \quad \text{P2}$$

- The sequents are separated by the rule (name) connecting them

- A last rule name (with no antecedent) terminates the sequence

$$\begin{array}{l} \quad d \in \mathbb{N} \\ \quad n \in \mathbb{N} \\ \quad n \leq d \\ \vdash \\ \quad n+1 \leq d \end{array} \quad \text{MON} \quad \begin{array}{l} \quad n \leq d \\ \vdash \\ \quad n+1 \leq d \end{array} \quad ?$$

- We put a **?** to indicate that we have no rule to apply

- The proof fails: we cannot conclude with rule **INC** ($n < d$ needed)

$$\frac{}{\mathsf{n} < \mathsf{m} \;\vdash\; \mathsf{n}+1 \leq \mathsf{m}} \quad \textbf{INC}$$

$$\begin{array}{l} \quad d \in \mathbb{N} \\ \quad n \in \mathbb{N} \\ \quad n \leq d \\ \vdash \\ \quad n - 1 \in \mathbb{N} \end{array} \qquad \text{MON} \qquad \begin{array}{l} \quad n \in \mathbb{N} \\ \vdash \\ \quad n - 1 \in \mathbb{N} \end{array} \qquad ?$$

- The proof fails: we cannot conclude with rule **P2′** ($0 < n$ needed)

$$\frac{}{0 < \mathbf{n} \vdash \mathbf{n} - 1 \in \mathbb{N}} \quad \textbf{P2}'$$

$$\boxed{\begin{array}{l} \quad d \in \mathbb{N} \\ \quad n \in \mathbb{N} \\ \quad n \leq d \\ \vdash \\ \quad n - 1 \leq d \end{array}}$$

MON

$$\boxed{\begin{array}{l} \quad n \leq d \\ \vdash \\ \quad n - 1 \leq d \end{array}}$$

DEC

$$\boxed{\dfrac{}{\mathbf{n} \leq \mathbf{m} \vdash \mathbf{n} - 1 \leq \mathbf{m}} \quad \textbf{DEC}}$$

- We needed hypothesis $n < d$ to prove ML_out / **inv0_2** / INV

- We needed hypothesis $0 < n$ to prove ML_in / **inv0_1** / INV

ML_out	**ML_in**
$n := n + 1$	$n := n - 1$

- We are going to add $n < d$ as a guard to event ML_out

- We are going to add $0 < n$ as a guard to event ML_in

Improving the Events: Introducing Guards

ML_out
when
$n < d$
then
$n := n + 1$
end

ML_in
when
$0 < n$
then
$n := n - 1$
end

- We are adding guards to the events

- The guard is the necessary condition for an event to “occur”

Adapting the Invariant Preservation Rule

- Given c with axioms $A(c)$ and v with invariants $I(c,v)$

- Given an event with guard $G(c,v)$ and b-a predicate $v' = E(c,v)$

- We modify the Invariant Preservation Rule as follows:

$$\begin{array}{|l|c|}\hline \begin{array}{l} \quad A(c) \\ \quad I(c,v) \\ \quad G(c,v) \\ \vdash \\ \quad I_i(c, E(c,v)) \end{array} & \text{INV} \\ \hline \end{array}$$

The Modified Rule

Axiomss Invariants Guard of the event $\vdash$ Modified Invariant	$A(c)$ $I(c, v)$ $G(c, v)$ $\vdash$ $I_i(c, E(c, v))$	INV

- Adding a new assumptions to a sequent does not modify a proof

$\boldsymbol{d} \in \mathbb{N}$ $\boldsymbol{n} \in \mathbb{N}$ $\boldsymbol{n} \leq \boldsymbol{d}$ $\boldsymbol{n} < \boldsymbol{d}$ $\vdash$ $\boldsymbol{n} + \mathbf{1} \in \mathbb{N}$	MON	$\boldsymbol{n} \in \mathbb{N}$ $\vdash$ $\boldsymbol{n} + \mathbf{1} \in \mathbb{N}$	P2

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ n < d \\ \vdash \\ n + 1 \leq d \end{array} \quad \text{MON} \quad \begin{array}{l} n < d \\ \vdash \\ n + 1 \leq d \end{array} \quad \textbf{INC}$$

- We can conclude now with rule **INC**

$$\frac{}{\mathbf{n} < \mathbf{m} \;\vdash\; \mathbf{n} + 1 \leq \mathbf{m}} \quad \textbf{INC}$$

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ 0 < n \\ \vdash \\ n - 1 \in \mathbb{N} \end{array}$$

MON

$$\begin{array}{l} 0 < n \\ \vdash \\ n - 1 \in \mathbb{N} \end{array}$$

P2'

- We can now conclude with rule **P2′**

$$\frac{}{0 < \mathbf{n} \vdash \mathbf{n} - 1 \in \mathbb{N}} \quad \mathbf{P2'}$$

$$\boxed{\begin{array}{l} \quad d \in \mathbb{N} \\ \quad n \in \mathbb{N} \\ \quad n \leq d \\ \vdash \\ \quad n - 1 \leq d \end{array}}$$

MON

$$\boxed{\begin{array}{l} \quad n \leq d \\ \vdash \\ \quad n - 1 \leq d \end{array}}$$

DEC

$$\boxed{\dfrac{}{\mathbf{n} \leq \mathbf{m} \vdash \mathbf{n} - 1 \leq \mathbf{m}} \quad \textbf{DEC}}$$

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \le d \\ n < d \\ \vdash \\ n + 1 \in \mathbb{N} \end{array}$$

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \le d \\ n < d \\ \vdash \\ n + 1 \le d \end{array}$$

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \le d \\ 0 < n \\ \vdash \\ n - 1 \in \mathbb{N} \end{array}$$

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \le d \\ 0 < n \\ \vdash \\ n - 1 \le d \end{array}$$

- Our system must be initialized (with no car in the island-bridge)

- The initialization event is never guarded

- It does not mention any variable in the right hand side of $:=$

-It before-after predicate is just an after predicate

init $n := 0$	After predicate	$n' = 0$

- Given c with axioms $A(c)$ and v with invariants $I(c, v)$

- Given an init event with after predicate $v' = K(c)$

- The Invariant Establishment Rule is the following:

Axioms $\vdash$ Modified Invariant	$A(c)$ $\vdash$ $I_i(c, K(c))$	INV

axm0_1 $\vdash$ Modified **inv0_1**	$d \in \mathbb{N}$ $\vdash$ $0 \in \mathbb{N}$	**inv0_1** / INV
axm0_1 $\vdash$ Modified **inv0_2**	$d \in \mathbb{N}$ $\vdash$ $0 \leq d$	**inv0_2** / INV

More Arithmetic Inference Rules

- First Peano Axiom

$$\frac{}{\vdash \ \mathbf{0} \in \mathbb{N}} \quad \textbf{P1}$$

- Third Peano Axiom (slightly modified)

$$\frac{}{\mathbf{n} \in \mathbb{N} \ \vdash \ \mathbf{0} \leq \mathbf{n}} \quad \textbf{P3}$$

$$\boxed{\begin{array}{l} \quad \boldsymbol{d} \;\in\; \mathbb{N} \\ \vdash \\ \quad \mathbf{0} \in \mathbb{N} \end{array}} \quad \text{MON} \quad \boxed{\begin{array}{l} \vdash \\ \quad \mathbf{0} \in \mathbb{N} \end{array}} \quad \text{P1}$$

$$\boxed{\begin{array}{l} \quad \boldsymbol{d} \;\in\; \mathbb{N} \\ \vdash \\ \quad \mathbf{0} \leq \boldsymbol{d} \end{array}} \quad \text{P3}$$

A Missing Requirement

- It is possible for the system to be blocked if both guards are false

- We do not want this to happen

- We figure out that one important requirement was missing

Once started, the system should work for ever	FUN-4

- Given c with axioms $A(c)$ and v with invariants $I(c, v)$

- Given the guards $G_1(c, v), \ldots, G_m(c, v)$ of the events

- We have to prove the following:

$A(c)$ $I(c, v)$ $\vdash$ $G_1(c, v) \;\vee\; \ldots \;\vee\; G_m(c, v)$	DLF

axm0_1 **inv0_1** **inv0_2** $\vdash$ Disjunction of guards	$d \in \mathbb{N}$ $n \in \mathbb{N}$ $n \leq d$ $\vdash$ $n < d \;\vee\; 0 < n$

- This cannot be proved with the inference rules we have so far

- $n \leq d$ can be replaced by $n = d \;\vee\; n < d$

- We have then to perform a proof by cases :

  - when $n = d$

  - when $n < d$

- Proof by cases. Note that we have two antecedents

$$\frac{\mathbf{H, P} \ \vdash \ \mathbf{R} \qquad\qquad \mathbf{H, Q} \ \vdash \ \mathbf{R}}{\mathbf{H},\ \mathbf{P} \lor \mathbf{Q} \ \vdash \ \mathbf{R}} \quad \mathbf{OR\_L}$$

- Choice for proving a disjunctive goal

$$\frac{\mathbf{H} \ \vdash \ \mathbf{P}}{\mathbf{H} \ \vdash \ \mathbf{P} \lor \mathbf{Q}} \quad \mathbf{OR\_R1}$$

$$\frac{\mathbf{H} \ \vdash \ \mathbf{Q}}{\mathbf{H} \ \vdash \ \mathbf{P} \lor \mathbf{Q}} \quad \mathbf{OR\_R2}$$

$$\boxed{\begin{array}{l} \quad \boldsymbol{d} \in \mathbb{N} \\ \quad \boldsymbol{n} \in \mathbb{N} \\ \quad \boldsymbol{n} \leq \boldsymbol{d} \\ \vdash \\ \quad \boldsymbol{n} < \boldsymbol{d} \;\vee\; \mathbf{0} < \boldsymbol{n} \end{array}} \;\textbf{MON}\; \boxed{\begin{array}{l} \quad \boldsymbol{n} \leq \boldsymbol{d} \\ \vdash \\ \quad \boldsymbol{n} < \boldsymbol{d} \;\vee\; \mathbf{0} < \boldsymbol{n} \end{array}} \;\ldots$$

$$\boxed{\begin{array}{l} \quad n \le d \\ \vdash \\ \quad n < d \;\; \vee \;\; 0 < n \end{array}} \quad \textbf{OR\_L} \quad \left\{ \begin{array}{l} \boxed{\begin{array}{l} \quad n < d \\ \vdash \\ \quad n < d \;\; \vee \;\; 0 < n \end{array}} \; \cdots \\ \\ \boxed{\begin{array}{l} \quad n = d \\ \vdash \\ \quad n < d \;\; \vee \;\; 0 < n \end{array}} \; \cdots \end{array} \right.$$

$$\left\{\begin{array}{l} \boxed{\begin{array}{l} \quad n < d \\ \vdash \\ \quad n < d \ \lor \ 0 < n \end{array}} \ \textbf{OR\_R1} \ \boxed{n < d \ \vdash \ n < d} \ \textbf{?} \\ \\ \boxed{\begin{array}{l} \quad n = d \\ \vdash \\ \quad n < d \ \lor \ 0 < n \end{array}} \ \textbf{?} \end{array}\right.$$

- The first **?** seems to be obvious

- The second **?** can be (partially) solved by applying the equality

- The basic property of a sequent

$$\frac{}{\mathbf{P} \;\vdash\; \mathbf{P}} \quad \textbf{HYP}$$

- Applying an equality (from left to right and vice-versa)

$$\frac{\mathbf{H}_{\mathbf{F}},\ \mathbf{E} = \mathbf{F} \;\vdash\; \mathbf{P}_{\mathbf{F}}}{\mathbf{H}_{\mathbf{E}},\ \mathbf{E} = \mathbf{F} \;\vdash\; \mathbf{P}_{\mathbf{E}}} \quad \textbf{EQ\_LR}$$

$$\frac{\mathbf{H}_{\mathbf{E}},\ \mathbf{E} = \mathbf{F} \;\vdash\; \mathbf{P}_{\mathbf{E}}}{\mathbf{H}_{\mathbf{F}},\ \mathbf{E} = \mathbf{F} \;\vdash\; \mathbf{P}_{\mathbf{F}}} \quad \textbf{EQ\_RL}$$

$$\left\{\begin{array}{l}
\boxed{\begin{array}{l} \quad n < d \\ \vdash \\ \quad n < d \;\;\vee\;\; 0 < n \end{array}} \;\; \textbf{OR_R1} \;\; \boxed{n < d \;\vdash\; n < d} \;\; \textbf{HYP} \\[2em]
\boxed{\begin{array}{l} \quad n = d \\ \vdash \\ \quad n < d \;\;\vee\;\; 0 < n \end{array}} \;\; \textbf{EQ_LR} \;\; \boxed{\vdash \;\; d < d \;\;\vee\;\; 0 < d} \;\; \textbf{OR_R2}
\end{array}\right.$$

$$\textbf{OR\_R2} \;\; \boxed{\vdash \;\; 0 < d} \;\; \textbf{?}$$

- We still have a problem: d must be positive!

- If d is equal to 0, then no car can ever enter the Island-Bridge

$$\boxed{\textbf{axm0\_2:} \quad 0 < d}$$

Initial Model: Conclusion

- Thanks to the proofs, we discovered 3 errors

- They were corrected by:

    - adding guards to both events

    - adding an axiom

- We have seen three Proof Rules:
  - The Invariant Establishment Rule: INV
  - The Invariant Preservation Rule: INV
  - The Deadlock Freeness Rule (not mandatory): DLF

Proof Rules (cont'd)

Axioms $\vdash$ Modified Invariant	INV

Axioms Invariants Guard of the event $\vdash$ Modified Invariant	INV

Axioms Invariants $\vdash$ Disjunction of the guards	DLF

$$\frac{\mathbf{H1} \vdash \mathbf{G}}{\mathbf{H1}, \mathbf{H2} \vdash \mathbf{G}} \quad \text{MON}$$

$$\frac{}{\mathbf{P} \vdash \mathbf{P}} \quad \text{HYP}$$

$$\frac{\mathbf{H}, \mathbf{P} \vdash \mathbf{R} \qquad \mathbf{H}, \mathbf{Q} \vdash \mathbf{R}}{\mathbf{H}, \mathbf{P} \lor \mathbf{Q} \vdash \mathbf{R}} \quad \text{OR\_L}$$

$$\frac{\mathbf{H} \vdash \mathbf{P}}{\mathbf{H} \vdash \mathbf{P} \lor \mathbf{Q}} \quad \text{OR\_R1}$$

$$\frac{\mathbf{H} \vdash \mathbf{Q}}{\mathbf{H} \vdash \mathbf{P} \lor \mathbf{Q}} \quad \text{OR\_R2}$$

$$\frac{\mathbf{H_F},\ \mathbf{E} = \mathbf{F} \ \vdash \ \mathbf{P_F}}{\mathbf{H_E},\ \mathbf{E} = \mathbf{F} \ \vdash \ \mathbf{P_E}} \qquad \text{EQ\_LR}$$

$$\frac{\mathbf{H_E},\ \mathbf{E} = \mathbf{F} \ \vdash \ \mathbf{P_E}}{\mathbf{H_F},\ \mathbf{E} = \mathbf{F} \ \vdash \ \mathbf{P_F}} \qquad \text{EQ\_RL}$$

$$\frac{}{\vdash \ 0 \in \mathbb{N}} \ \mathsf{P1}$$

$$\frac{}{\mathbf{n} \in \mathbb{N} \ \vdash \ \mathbf{n} + 1 \in \mathbb{N}} \ \mathsf{P2}$$

$$\frac{}{0 < \mathbf{n} \ \vdash \ \mathbf{n} - 1 \in \mathbb{N}} \ \mathsf{P2'}$$

$$\frac{}{\mathbf{n} \in \mathbb{N} \ \vdash \ 0 \leq \mathbf{n}} \ \mathsf{P3}$$

$$\frac{}{\mathbf{n} \leq \mathbf{m} \ \vdash \ \mathbf{n} - 1 \leq m} \ \mathsf{DEC}$$

$$\frac{}{\mathbf{n} < \mathbf{m} \ \vdash \ \mathbf{n} + 1 \leq \mathrm{m}} \ \mathsf{INC}$$

Summary of Initial Model

constant: d

variable: n

axm0_1: $d \in \mathbb{N}$

axm0_2: $d > 0$

inv0_1: $n \in \mathbb{N}$

inv0_2: $n \leq d$

init
$n := 0$

ML_out
when
$n < d$
then
$n := n + 1$
end

ML_in
when
$0 < n$
then
$n := n - 1$
end

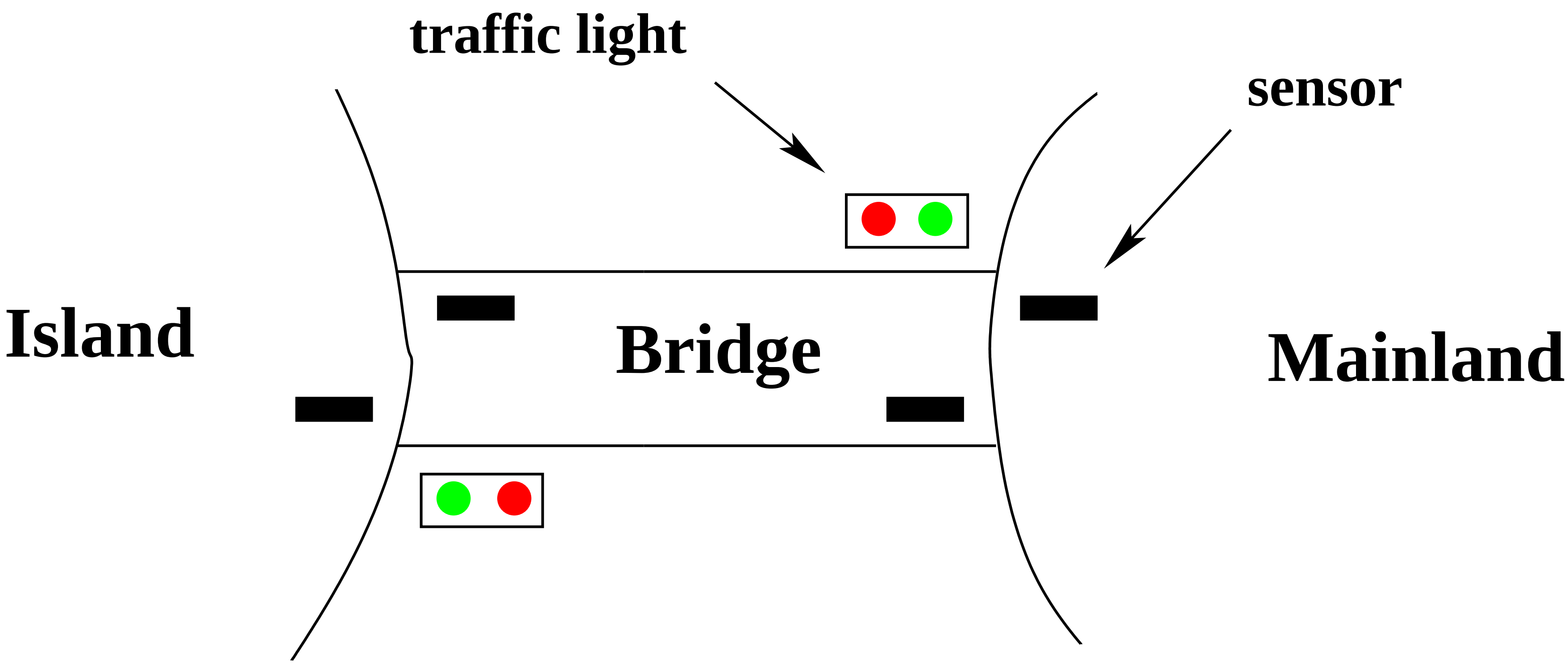
Our Development Approach

- Initial model: Limiting the number of cars (FUN_2)

- First refinement: Introducing the one way bridge (FUN_3)

- Second refinement: Introducing the traffic lights (EQP_1,2,3)

- Third refinement: Introducing the sensors (EQP_4,5)


traffic light
sensor
Island
Bridge
Mainland

First Refinement: Introducing a One-Way Bridge

- We go down with our parachute

- Our view of the system gets more accurate

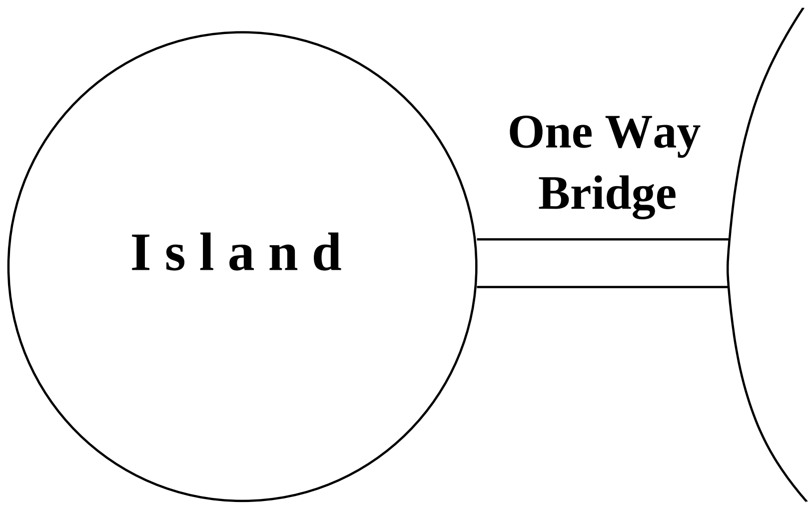
- We introduce the bridge and separate it from the island

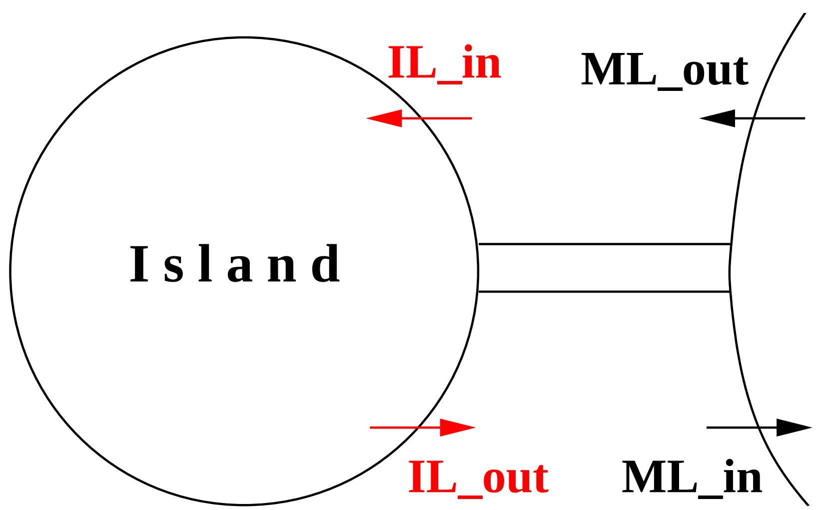
- We shall refine the state and the events

- But we add also two new events: IL_in and IL_out

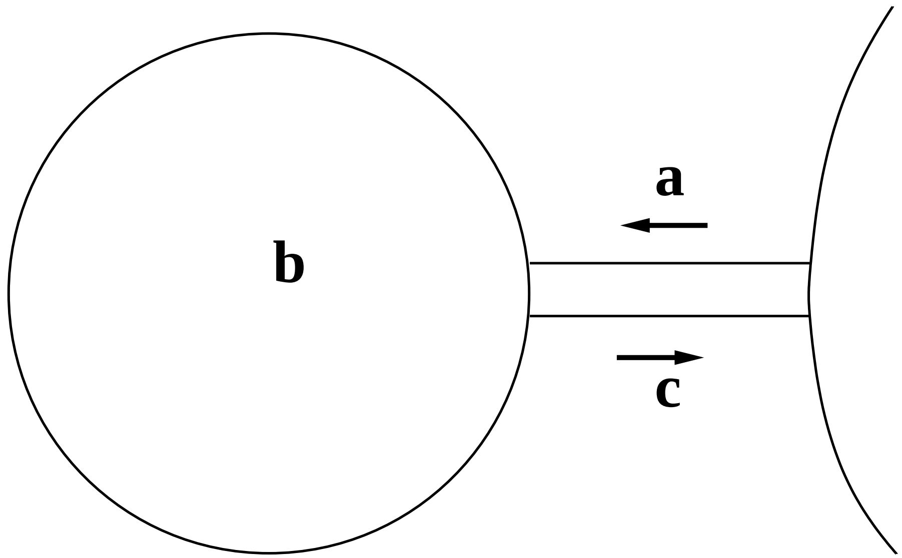
- We are focusing on FUN-3 (one way bridge)

First Refinement: Introducing a one Way Bridge





Introducing Three New Variables: a, b, and c



- a denotes the number of cars on bridge going to island
- b denotes the number of cars on island
- c denotes the number of cars on bridge going to mainland
- a, b, and c are the concrete variables
- They replace the abstract variable n

- Variables $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ denote natural numbers

$$\boxed{\begin{array}{l} \boldsymbol{a} \in \mathbb{N} \\ \boldsymbol{b} \in \mathbb{N} \\ \boldsymbol{c} \in \mathbb{N} \end{array}}$$

- Relating the concrete state (a, b, c) to the abstract state (n)

$$\boxed{a + b + c = n}$$

- Formalizing the new invariant: one way bridge (this is FUN-3)

$$\boxed{a = 0 \quad \vee \quad c = 0}$$

Refining the State: Summary

constants: d

variables: a, b, c

inv1_1: $a \in \mathbb{N}$

inv1_2: $b \in \mathbb{N}$

inv1_3: $c \in \mathbb{N}$

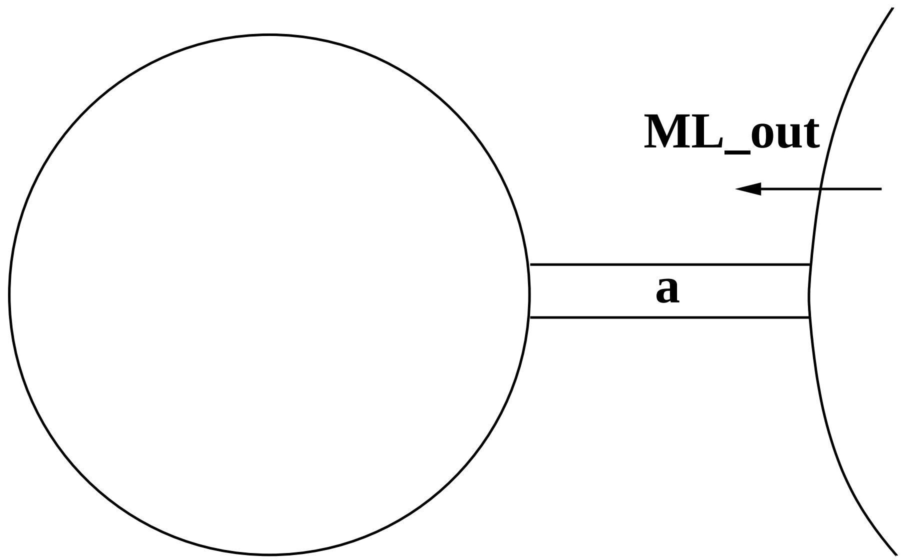
inv1_4: $a + b + c = n$

inv1_5: $a = 0 \;\lor\; c = 0$

- Invariants **inv1_1** to **inv1_5** are called the concrete invariants

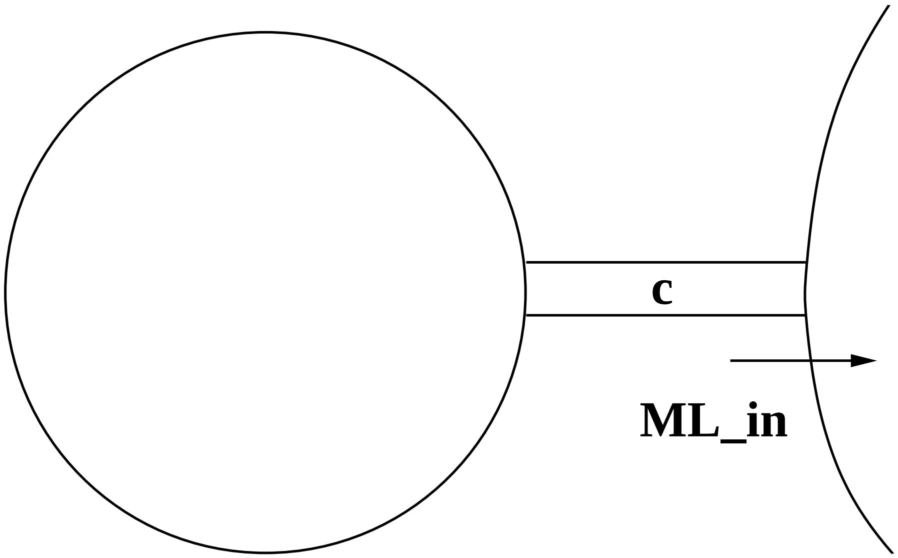
- **inv1_4** glues the abstract state, n, to the concrete state, a, b, c

Proposal for Refining Event ML_out



ML_out
when
$a + b < d$
$c = 0$
then
$a := a + 1$
end

Proposal for Refining Event ML_in



ML_in
when
$0 < c$
then
$c := c - 1$
end

B-A Predicates: Missing Variables are not Modified

ML_out
when
$a + b < d$
$c = 0$
then
$a := a + 1$
end

ML_in
when
$0 < c$
then
$c := c - 1$
end

Before-after predicates showing the unmodifed variables:

$$a' = a + 1 \quad \land \quad b' = b \quad \land \quad c' = c$$

$$a' = a \quad \land \quad b' = b \quad \land \quad c' = c - 1$$

Intuition about refinement (1)

(abstract_)ML_out
  when
    $n < d$
  then
    $n := n + 1$
  end

(concrete_)ML_out
  when
    $a + b < d$
    $c = 0$
  then
    $a := a + 1$
  end

- The concrete version is not contradictory with the abstract one

- When the concrete version is enabled then so is the abstract one

- Executions seems to be compatible

Intuition about refinement (2)

(abstract_)ML_in **when** $0 < n$ **then** $n := n - 1$ **end**	(concrete_)ML_in **when** $0 < c$ **then** $c := c - 1$ **end**

- Same remarks as in the previous slide

- But this has to be confirmed by well defined proof obligations

What is to be proved

- The concrete guard is stronger than the abstract one

- The actions are compatible

Constants c with axioms $A(c)$

Abstract variables v with abstract invariant $I(c, v)$

Concrete variables w with concrete invariant $J(c, v, w)$

Abstract event with guards $G(c, v)$: $G_1(c, v), G_2(c, v), \ldots$

Abstract event with before-after predicate $v' = E(c, v)$

Concrete event with guards $H(c, w)$ and b-a predicate $w' = F(c, w)$

Axioms Abstract Invariant Concrete Invariant Concrete Guard $\vdash$ Abstract Guard	$A(c)$ $I(c, v)$ $J(c, v, w)$ $H(c, w)$ $\vdash$ $G_i(c, v)$	GRD

What we Have to Prove

- ML_out / GRD

- ML_in / GRD

axm0_1
axm0_2
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
inv1_4
inv1_5
Concrete guards of ML_out

$\vdash$
Abstract guard of ML_out

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \quad \lor \quad c = 0 \\
a + b < d \\
c = 0 \\
\vdash \\
n < d
\end{array}
$$

ML_out / GRD

(abstract-)ML_out
when
$n < d$
then
$n := n + 1$
end

(concrete-)ML_out
when
$a + b < d$
$c = 0$
then
$a := a + 1$
end

$$
\begin{array}{l}
\quad d \in \mathbb{N} \\
\quad 0 < d \\
\quad n \in \mathbb{N} \\
\quad n \leq d \\
\quad a \in \mathbb{N} \\
\quad b \in \mathbb{N} \\
\quad c \in \mathbb{N} \\
\quad a + b + c = n \\
\quad a = 0 \quad \lor \quad c = 0 \\
\quad a + b < d \\
\quad c = 0 \\
\vdash \\
\quad n < d
\end{array}
\quad \text{MON} \quad
\begin{array}{l}
\quad a + b + c = n \\
\quad a + b < d \\
\quad c = 0 \\
\vdash \\
\quad n < d
\end{array}
\quad \text{EQ\_LR} \quad
\begin{array}{l}
\quad \underline{a + b + 0 = n} \\
\quad a + b < d \\
\vdash \\
\quad n < d
\end{array}
\quad \text{ARITH} \quad \ldots
$$

$$
\ldots \quad
\begin{array}{l}
\quad a + b = n \\
\quad a + b < d \\
\vdash \\
\quad n < d
\end{array}
\quad \text{EQ\_LR} \quad
\begin{array}{l}
\quad n < d \\
\vdash \\
\quad n < d
\end{array}
\quad \text{HYP}
$$

- We use the "rule" ARITH to stand for simple arithmetic simplifications (where underlined)

axm0_1
axm0_2
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
inv1_4
inv1_5
Concrete guard of ML_in
$\vdash$
Abstract guard of ML_in

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \quad \lor \quad c = 0 \\
0 < c \\
\vdash \\
0 < n
\end{array}
$$

ML_in / GRD

(abstrasct-)ML_in
when
$0 < n$
then
$n := n - 1$
end

(concrete-)ML_in
when
$0 < c$
then
$c := c - 1$
end

$$\begin{array}{l} d \in \mathbb{N} \\ 0 < d \\ n \in \mathbb{N} \\ n \leq d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ c \in \mathbb{N} \\ a+b+c=n \\ a=0 \;\vee\; c=0 \\ 0 < c \\ \vdash \\ 0 < n \end{array} \quad \text{MON} \quad \begin{array}{l} b \in \mathbb{N} \\ a+b+c=n \\ a=0 \;\vee\; c=0 \\ 0 < c \\ \vdash \\ 0 < n \end{array} \quad \text{OR\_L} \quad \left\{ \begin{array}{l} \begin{array}{l} b \in \mathbb{N} \\ a+b+c=n \\ a=0 \\ 0 < c \\ \vdash \\ 0 < n \end{array} \quad \text{EQ\_LR} \quad \ldots \\ \\ \begin{array}{l} b \in \mathbb{N} \\ a+b+c=n \\ c=0 \\ 0 < c \\ \vdash \\ 0 < n \end{array} \quad \text{EQ\_LR} \quad \ldots \end{array} \right.$$

$$\begin{array}{l} d \in \mathbb{N} \\ 0 < d \\ n \in \mathbb{N} \\ n \leq d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ c \in \mathbb{N} \\ a+b+c=n \\ a=0 \ \lor \ c=0 \\ 0 < c \\ \vdash \\ 0 < n \end{array}$$

MON

$$\begin{array}{l} b \in \mathbb{N} \\ a+b+c=n \\ a=0 \ \lor \ c=0 \\ 0 < c \\ \vdash \\ 0 < n \end{array}$$

OR_L

$$\begin{array}{l} b \in \mathbb{N} \\ a+b+c=n \\ a=0 \\ 0 < c \\ \vdash \\ 0 < n \end{array}$$

EQ_LR . . .

$$\begin{array}{l} b \in \mathbb{N} \\ a+b+c=n \\ c=0 \\ 0 < c \\ \vdash \\ 0 < n \end{array}$$

EQ_LR . . .

. . .

$$\begin{array}{l} b \in \mathbb{N} \\ \underline{0+b+c=n} \\ 0 < c \\ \vdash \\ 0 < n \end{array}$$

ARITH

$$\begin{array}{l} \underline{b \in \mathbb{N}} \\ \underline{b+c=n} \\ 0 < c \\ \vdash \\ 0 < n \end{array}$$

ARITH

$$\begin{array}{l} \underline{c \leq n} \\ \underline{0 < c} \\ \vdash \\ 0 < n \end{array}$$

ARITH

$$\begin{array}{l} 0 < n \\ \vdash \\ 0 < n \end{array}$$

HYP

. . .

$$\begin{array}{l} b \in \mathbb{N} \\ \underline{a+b+0=n} \\ 0 < 0 \\ \vdash \\ 0 < n \end{array}$$

ARITH

$$\begin{array}{l} b \in \mathbb{N} \\ a+b=n \\ 0 < 0 \\ \vdash \\ 0 < n \end{array}$$

MON

$$\begin{array}{l} \underline{0 < 0} \\ \vdash \\ 0 < n \end{array}$$

ARITH

$$\begin{array}{l} \perp \\ \vdash \\ 0 < n \end{array}$$

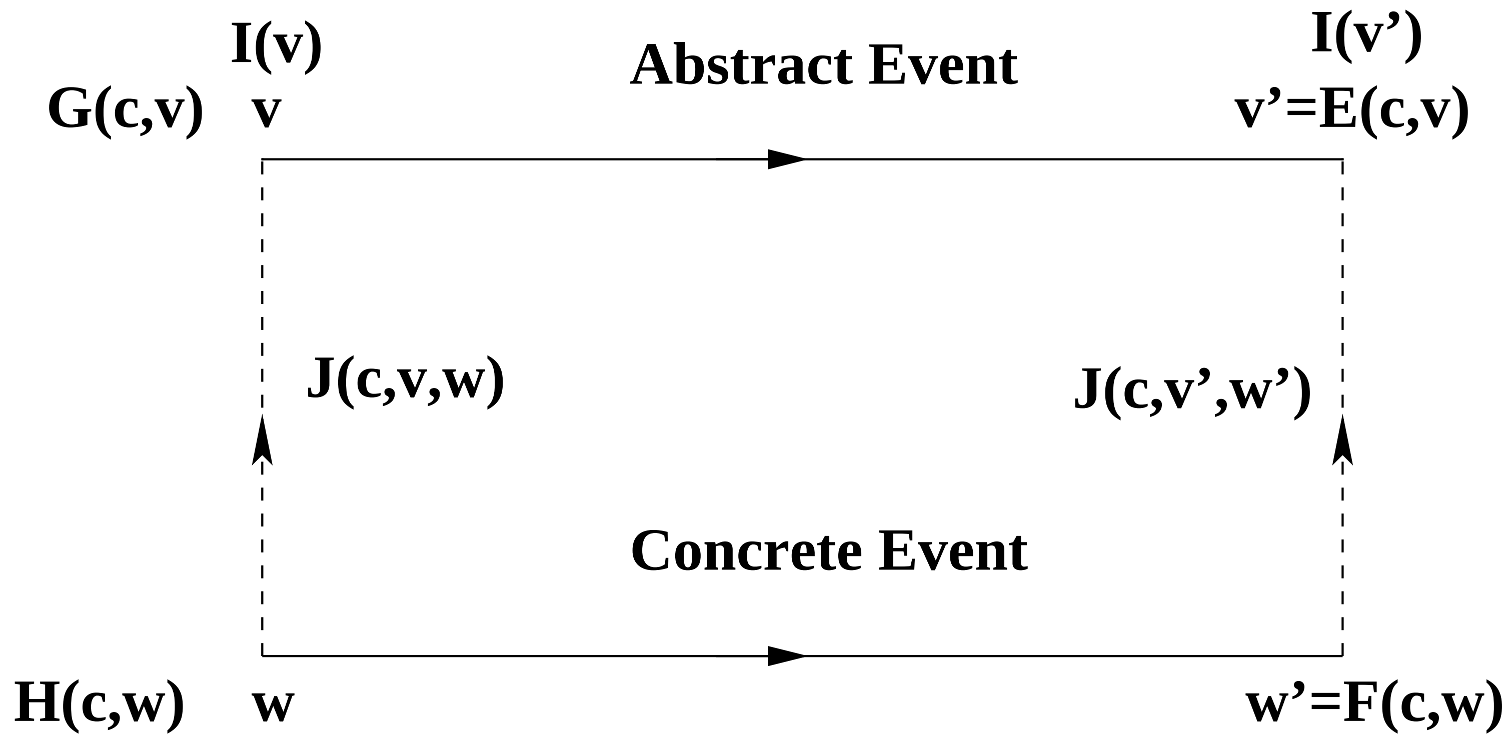
CNTR

An Additional Rule: the Contradiction Rule

- In the previous proof, we have used and additional inference rule

- It says that a false hypothesis entails any goal

$$\frac{}{\perp \vdash \mathbf{P}} \quad \text{CNTR}$$


I(v)
Abstract Event
I(v')
G(c,v)
v
v'=E(c,v)
J(c,v,w)
J(c,v',w')
Concrete Event
H(c,w)
w
w'=F(c,w)

Invariant Refinement Rule

Axioms Abstract Invariants Concrete Invariants Concrete Guards $\vdash$ Modified Concrete Invariant	$A(c)$ $I(c, v)$ $J(c, v, w)$ $H(c, w)$ $\vdash$ $J_j(c, E(c, v), F(c, w))$	INV

What we Have to Prove

- ML_out / GRD done

- ML_in / GRD done

- ML_out / **inv1_4** / INV

- ML_out / **inv1_5** / INV

- ML_in / **inv1_4** / INV

- ML_in / **inv1_5** / INV

Applying Invariant Refinement to Event ML_out

axm0_1
axm0_2
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
inv1_4
inv1_5
Concrete guards of ML_out

$\vdash$

Modified Invariant **inv1_4**

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \quad \lor \quad c = 0 \\
a + b < d \\
c = 0 \\
\vdash \\
a + 1 + b + c = n + 1
\end{array}
$$

ML_out / **inv1_4** / INV

(abstract-)ML_out
when
$n < d$
then
$n := n + 1$
end

(concrete-)ML_out
when
$a + b < d$
$c = 0$
then
$a := a + 1$
end

$$
\boxed{\begin{array}{l}
\quad d \in \mathbb{N} \\
\quad 0 < d \\
\quad n \in \mathbb{N} \\
\quad n \leq d \\
\quad a \in \mathbb{N} \\
\quad b \in \mathbb{N} \\
\quad c \in \mathbb{N} \\
\quad a + b + c = n \\
\quad a = 0 \quad \vee \quad c = 0 \\
\quad a + b < d \\
\quad c = 0 \\
\vdash \\
\quad a + 1 + b + c = n + 1
\end{array}}
\quad \text{MON} \quad
\boxed{\begin{array}{l}
\quad a + b + c = n \\
\vdash \\
\quad \underline{a + 1 + b + c = n + 1}
\end{array}}
\quad \text{ARITH} \quad \ldots
$$

$$
\ldots \quad
\boxed{\begin{array}{l}
\quad a + b + c = n \\
\vdash \\
\quad a + b + c + 1 = n + 1
\end{array}}
\quad \text{EQ\_LR} \quad
\boxed{\vdash \quad n + 1 = n + 1}
\quad \text{EQL}
$$

An Additional Rule: the Equality Rule

- In the previous proof, we have used and additional inference rule

- It says that a any term is equal to itself

$$\frac{}{\vdash \ \mathbf{E} = \mathbf{E}} \ \text{EQL}$$

- Remark: we have not yet formally defined "term" and "predicate"

- This will be done in later lectures

Applying Invariant Refinement to Event ML_out

axm0_1
axm0_2
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
inv1_4
inv1_5
Concrete guards of ML_out

$\vdash$

Modified Invariant **inv1_5**

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \ \lor \ c = 0 \\
a + b < d \\
c = 0 \\
\vdash \\
a + 1 = 0 \ \lor \ c = 0
\end{array}
$$

ML_out / **inv1_5** / INV

(abstract-)ML_out
when
$n < d$
then
$n := n + 1$
end

(concrete-)ML_out
when
$a + b < d$
$c = 0$
then
$a := a + 1$
end

$$\begin{array}{l} d \in \mathbb{N} \\ 0 < d \\ n \in \mathbb{N} \\ n \leq d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ c \in \mathbb{N} \\ a + b + c = n \\ a = 0 \ \lor \ c = 0 \\ a + b < d \\ c = 0 \\ \vdash \\ a + 1 = 0 \ \lor \ c = 0 \end{array}$$

MON

$$\begin{array}{l} c = 0 \\ \vdash \\ a + 1 = 0 \ \lor \ c = 0 \end{array}$$

OR_R2

$$\begin{array}{l} c = 0 \\ \vdash \\ c = 0 \end{array}$$

HYP

Applying Invariant Refinement to Event ML_in

axm0_1
axm0_2
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
inv1_4
inv1_5
Concrete guards of ML_in
$\vdash$
Modified Invariant **inv1_4**

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \quad \lor \quad c = 0 \\
0 < c \\
\vdash \\
a + b + c - 1 = n - 1
\end{array}
$$

ML_in / **inv1_4** / INV

(abstract-)ML_in
when
$0 < n$
then
$n := n - 1$
end

(concrte-)ML_in
when
$0 < c$
then
$c := c - 1$
end

$$\begin{array}{l} \quad d \in \mathbb{N} \\ \quad 0 < d \\ \quad n \in \mathbb{N} \\ \quad n \leq d \\ \quad a \in \mathbb{N} \\ \quad b \in \mathbb{N} \\ \quad c \in \mathbb{N} \\ \quad a + b + c = n \\ \quad a = 0 \;\lor\; c = 0 \\ \quad 0 < c \\ \vdash \\ \quad a + b + c - 1 = n - 1 \end{array}$$

MON

$$\begin{array}{l} \quad a + b + c = n \\ \vdash \\ \quad a + b + c - 1 = n - 1 \end{array}$$

EQ_LR

$$\vdash \quad n - 1 = n - 1$$

EQL

axm0_1
axm0_2
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
inv1_4
inv1_5
Concrete guards of ML_in
⊢
Modified Invariant **inv1_5**

$$\begin{array}{l} d \in \mathbb{N} \\ 0 < d \\ n \in \mathbb{N} \\ n \leq d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ c \in \mathbb{N} \\ a + b + c = n \\ a = 0 \ \lor \ c = 0 \\ 0 < c \\ \vdash \\ a = 0 \ \lor \ c - 1 = 0 \end{array}$$

ML_in / **inv1_5** / INV

(abstract-)ML_in
when
$0 < n$
then
$n := n - 1$
end

(concrete-)ML_in
when
$0 < c$
then
$c := c - 1$
end

Proof

$$\begin{array}{l} d \in \mathbb{N} \\ 0 < d \\ n \in \mathbb{N} \\ n \leq d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ c \in \mathbb{N} \\ a + b + c = n \\ a = 0 \lor c = 0 \\ 0 < c \\ \vdash \\ a = 0 \lor c - 1 = 0 \end{array}$$

MON

$$\begin{array}{l} a = 0 \lor c = 0 \\ 0 < c \\ \vdash \\ a = 0 \lor c - 1 = 0 \end{array}$$

OR_L

$$\left\{ \begin{array}{l} \boxed{\begin{array}{l} a = 0 \\ 0 < c \\ \vdash \\ a = 0 \lor c - 1 = 0 \end{array}} \ \text{MON} \cdots \\ \\ \boxed{\begin{array}{l} c = 0 \\ 0 < c \\ \vdash \\ a = 0 \lor c - 1 = 0 \end{array}} \ \text{EQ\_LR} \cdots \end{array} \right.$$

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \ \vee \ c = 0 \\
0 < c \\
\vdash \\
a = 0 \ \vee \ c - 1 = 0
\end{array}
$$

MON

$$
\begin{array}{l}
a = 0 \ \vee \ c = 0 \\
0 < c \\
\vdash \\
a = 0 \ \vee \ c - 1 = 0
\end{array}
$$

OR_L

$$
\left\{
\begin{array}{ll}
\boxed{\begin{array}{l} a = 0 \\ 0 < c \\ \vdash \\ a = 0 \ \vee \ c - 1 = 0 \end{array}} & \text{MON} \cdots \\[2ex]
\boxed{\begin{array}{l} c = 0 \\ 0 < c \\ \vdash \\ a = 0 \ \vee \ c - 1 = 0 \end{array}} & \text{EQ\_LR} \cdots
\end{array}
\right.
$$

$$
\left\{
\begin{array}{lllll}
\cdots & \boxed{\begin{array}{l} a = 0 \\ \vdash \\ a = 0 \ \vee \ c - 1 = 0 \end{array}} & \textbf{OR\_R1} & \boxed{a = 0 \ \vdash \ a = 0} & \textbf{HYP} \\[2ex]
\cdots & \boxed{\underline{0 < 0} \ \vdash \ a = 0 \ \vee \ -1 = 0} & \textbf{ARITH} & \boxed{\bot \ \vdash \ a = 0 \ \vee \ -1 = 0} & \textbf{CNTR}
\end{array}
\right.
$$

Refining the Initialization Event init

- Concrete initialization

init
$$a, b, c := 0, 0, 0$$

- Corresponding after predicate

$$a' = 0 \quad \wedge \quad b' = 0 \quad \wedge \quad c' = 0$$

Refinement Rule for Initialization

Constants c with axioms $A(c)$

Concrete invariant $J(c, v, w)$

Abstract initialization with after predicate $v' = K(c)$

Concrete initialization with after predicate $w' = L(c)$

Axioms $\vdash$ Modified concrete invariants		$A(c)$ $\vdash$ $J_j(c, K(c), L(c))$	INV

What we Have to Prove

- ML_out / GRD done
- ML_in / GRD done
- ML_out / **inv1_4** / INV done
- ML_out / **inv1_5** / INV done
- ML_in / **inv1_4** / INV done
- ML_in / **inv1_5** / INV done
- **inv1_4** / INV
- **inv1_5** / INV

Applying the Initialization Refinement Rule

axm0_1
axm0_2
$\vdash$
Modified concrete invariant **inv1_4**

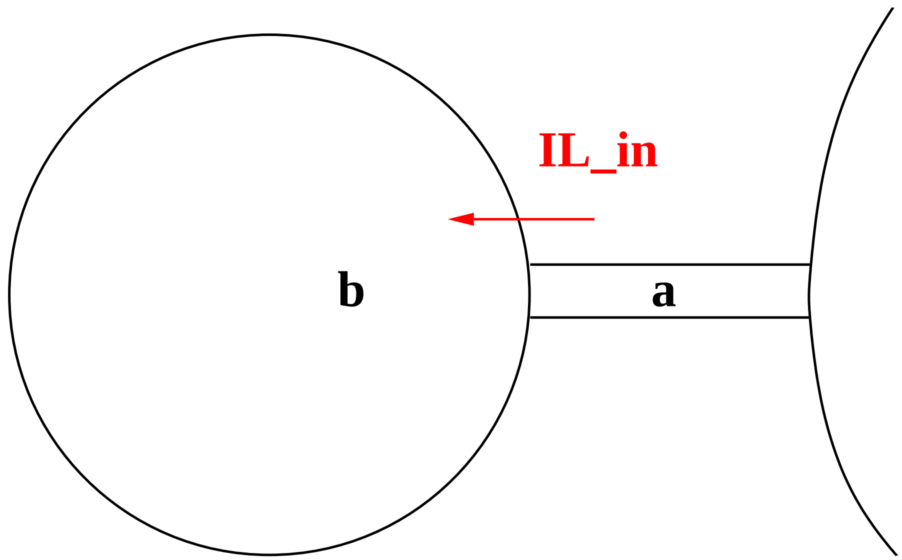
$$(a + b + c = n)$$

$$d \in \mathbb{N}$$
$$d > 0$$
$$\vdash$$
$$0 + 0 + 0 = 0$$

axm0_1
axm0_2
$\vdash$
Modified concrete invariant **inv1_5**

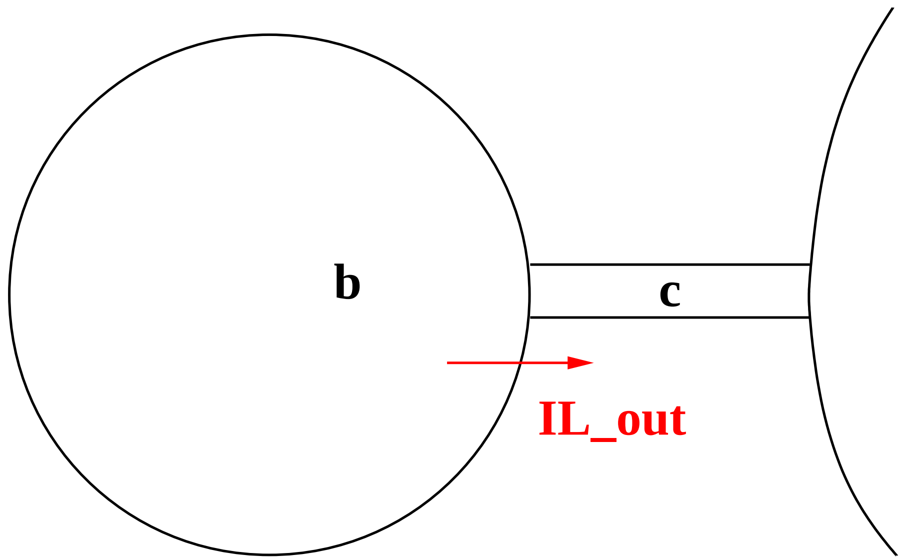
$$(a = 0 \ \lor \ c = 0)$$

$$d \in \mathbb{N}$$
$$d > 0$$
$$\vdash$$
$$0 = 0 \ \lor \ 0 = 0$$



IL_in
when
$0 < a$
then
$a, b := a - 1, b + 1$
end

New Event IL_out



IL_out
when
$0 < b$
$a = 0$
then
$b, c := b - 1, c + 1$
end

IL_in
  when
    $0 < a$
  then
    $a := a - 1$
    $b := b + 1$
  end

IL_out
  when
    $0 < b$
    $a = 0$
  then
    $b := b - 1$
    $c := c + 1$
  end

Before-after predicates

$$a' = a + 1 \quad \wedge \quad b' = b + 1 \quad \wedge \quad c' = c$$

$$a' = a \quad \wedge \quad b' = b - 1 \quad \wedge \quad c' = c + 1$$

The empty assignment skip

The before-after predicate of skip in the initial model

$$\boxed{n' = n}$$

The before-after predicate of skip in the first refinement

$$\boxed{a' = a \quad \wedge \quad b' = b \quad \wedge \quad c' = c}$$

Proving Correctness of New Events

(1) A new event must refine an implicit event, made of a skip action

(2) The new events must not diverge

- For this one has to exhibit a variant
- The variant is a natural number (could be more complicated)
- Each new event must decrease this variant

What we Have to Prove

- ML_out / GRD done
- ML_in / GRD done
- ML_out / **inv1_4** / INV done
- ML_out / **inv1_5** / INV done
- ML_in / **inv1_4** / INV done
- ML_in / **inv1_5** / INV done
- **inv1_4** / INV done
- **inv1_5** / INV done
- IL_in / **inv1_4** / INV
- IL_in / **inv1_5** / INV
- IL_out / **inv1_4** / INV
- IL_out / **inv1_5** / INV

Event IL_in Refines skip (1)

axm0_1
axm0_2
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
inv1_4
inv1_5
Concrete guards of IL_in
$\vdash$
Modified Invariant **inv1_4**

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \quad \lor \quad c = 0 \\
0 < a \\
\vdash \\
a - 1 + b + 1 + c = n
\end{array}
$$

IL_in / **inv1_4** / INV

IL_in
when
$0 < a$
then
$a := a - 1$
$b := b + 1$
end

Proof

$$
\begin{array}{l}
\boldsymbol{d} \in \mathbb{N} \\
0 < \boldsymbol{d} \\
\boldsymbol{n} \in \mathbb{N} \\
\boldsymbol{n} \leq \boldsymbol{d} \\
\boldsymbol{a} \in \mathbb{N} \\
\boldsymbol{b} \in \mathbb{N} \\
\boldsymbol{c} \in \mathbb{N} \\
\boldsymbol{a} + \boldsymbol{b} + \boldsymbol{c} = \boldsymbol{n} \\
\boldsymbol{a} = 0 \quad \lor \quad \boldsymbol{c} = 0 \\
0 < \boldsymbol{a} \\
\vdash \\
\boldsymbol{a} - 1 + \boldsymbol{b} + 1 + \boldsymbol{c} = \boldsymbol{n}
\end{array}
$$

MON

$$
\begin{array}{l}
\boldsymbol{a} + \boldsymbol{b} + \boldsymbol{c} = \boldsymbol{n} \\
\vdash \\
\underline{\boldsymbol{a} - 1 + \boldsymbol{b} + 1 + \boldsymbol{c} = \boldsymbol{n}}
\end{array}
$$

ARITH

$$
\begin{array}{l}
\boldsymbol{a} + \boldsymbol{b} + \boldsymbol{c} = \boldsymbol{n} \\
\vdash \\
\boldsymbol{a} + \boldsymbol{b} + \boldsymbol{c} = \boldsymbol{n}
\end{array}
$$

HYP

Event IL_in Refines skip (2)

axm0_1
axm0_2
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
inv1_4
inv1_5
Concrete guards of IL_in
$\vdash$
Modified Invariant **inv1_5**

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \quad \lor \quad c = 0 \\
0 < a \\
\vdash \\
a - 1 = 0 \quad \lor \quad c = 0
\end{array}
$$

IL_in / **inv1_5** / INV

IL_in
when
$0 < a$
then
$a := a - 1$
$b := b + 1$
end

Proof

$$\begin{array}{l} \quad d \in \mathbb{N} \\ \quad 0 < d \\ \quad n \in \mathbb{N} \\ \quad n \leq d \\ \quad a \in \mathbb{N} \\ \quad b \in \mathbb{N} \\ \quad c \in \mathbb{N} \\ \quad a + b + c = n \\ \quad a = 0 \;\vee\; c = 0 \\ \quad 0 < a \\ \vdash \\ \quad a - 1 = 0 \;\vee\; c = 0 \end{array} \quad \text{MON} \quad \begin{array}{l} \quad a = 0 \;\vee\; c = 0 \\ \quad 0 < a \\ \vdash \\ \quad a - 1 = 0 \;\vee\; c = 0 \end{array} \quad \text{OR\_L} \quad \cdots$$

$$\boxed{\begin{array}{l} \quad d \in \mathbb{N} \\ \quad 0 < d \\ \quad n \in \mathbb{N} \\ \quad n \leq d \\ \quad a \in \mathbb{N} \\ \quad b \in \mathbb{N} \\ \quad c \in \mathbb{N} \\ \quad a + b + c = n \\ \quad a = 0 \quad \vee \quad c = 0 \\ \quad 0 < a \\ \vdash \\ \quad a - 1 = 0 \quad \vee \quad c = 0 \end{array}} \quad \text{MON} \quad \boxed{\begin{array}{l} \quad a = 0 \quad \vee \quad c = 0 \\ \quad 0 < a \\ \vdash \\ \quad a - 1 = 0 \quad \vee \quad c = 0 \end{array}} \quad \text{OR\_L} \quad \cdots$$

$$\cdots \left\{ \begin{array}{l} \boxed{\begin{array}{l} \quad a = 0 \\ \quad 0 < a \\ \vdash \\ \quad a - 1 = 0 \quad \vee \quad c = 0 \end{array}} \quad \textbf{EQ\_LR} \quad \boxed{\begin{array}{l} \quad \underline{0 < 0} \\ \vdash \\ \quad -1 = 0 \quad \vee \quad c = 0 \end{array}} \quad \textbf{ARITH} \quad \boxed{\begin{array}{l} \quad \perp \\ \vdash \\ \quad -1 = 0 \quad \vee \quad c = 0 \end{array}} \quad \textbf{CNTR} \\ \\ \boxed{\begin{array}{l} \quad c = 0 \\ \quad 0 < a \\ \vdash \\ \quad a - 1 = 0 \quad \vee \quad c = 0 \end{array}} \quad \textbf{MON} \quad \boxed{\begin{array}{l} \quad c = 0 \\ \vdash \\ \quad a - 1 = 0 \quad \vee \quad c = 0 \end{array}} \quad \textbf{OR\_R2} \quad \boxed{c = 0 \quad \vdash \quad c = 0} \quad \textbf{HYP} \end{array} \right.$$

Non-divergence of New Events (1)

Axioms $A(c)$, invariants $I(c, v)$, concrete invariant $J(c, v, w)$

New event with guard $H(c, w)$

Variant $V(c, w)$

Axioms Abstract invariants Concrete invariants Concrete guard of a new event $\vdash$ Variant $\in \mathbb{N}$	$A(c)$ $I(c, v)$ $J(c, v, w)$ $H(c, w)$ $\vdash$ $V(c, w) \in \mathbb{N}$	NAT

Axioms $A(c)$, invariants $I(c, v)$, concrete invariant $J(c, v, w)$

New event with guard $H(c, w)$ and b-a predicate $w' = F(c, w)$

Variant $V(c, w)$

Axioms Abstract invariant Concrete invariant Concrete guard $\vdash$ Modified Var. $<$ Var.	$A(c)$ $I(c, v)$ $J(c, v, w)$ $H(c, w)$ $\vdash$ $V(c, F(c, w)) < V(c, w)$	VAR

variant_1: $2 * a + b$

What we Have to Prove

—ML_out / GRD done
—ML_in / GRD done
—ML_out / **inv1_4** / INV done
—ML_out / **inv1_5** / INV done
—ML_in / **inv1_4** / INV done
—ML_in / **inv1_5** / INV done
—**inv1_4** / INV done
—**inv1_5** / INV done
—IL_in / **inv1_4** / INV done
—IL_in / **inv1_5** / INV done
—IL_out / **inv1_4** / INV done
—IL_out / **inv1_5** / INV done

—NAT
—IL_in / VAR
—IL_out / VAR

axm0_1	$d \in \mathbb{N}$
axm0_2	$0 < d$
inv0_1	$n \in \mathbb{N}$
inv0_2	$n \leq d$
inv1_1	$a \in \mathbb{N}$
inv1_2	$b \in \mathbb{N}$
inv1_3	$c \in \mathbb{N}$
inv1_4	$a + b + c = n$
inv1_5	$a = 0 \quad \vee \quad c = 0$
Concrete guard of IL_in	$0 < a$
$\vdash$	$\vdash$
Modified variant $<$ Variant	$2 * (a - 1) + b + 1 < 2 * a + b$

IL_in / VAR

IL_in
when
$0 < a$
then
$a := a - 1$
$b := b + 1$
end

axm0_1
axm0_2
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
inv1_4
inv1_5
Concrete guards of IL_out

$\vdash$
Modified variant $<$ Variant

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \quad \lor \quad c = 0 \\
0 < b \\
a = 0 \\
\vdash \\
2 * a + b - 1 < 2 * a + b
\end{array}
$$

IL_out / VAR

IL_out
when
$0 < b$
$a = 0$
then
$b := b - 1$
$c := c + 1$
end

The $G_i(c, v)$ are the abstract guards

The $H_i(c, v)$ are the concrete guards

We have to prove the following Deadlock Freedom Rule

$A(c)$ $I(c, v)$ $J(c, v, w)$ $G_1(c, v) \ \vee \ \ldots \ \vee \ G_m(c, v)$ $\vdash$ $H_1(c, w) \ \vee \ \ldots \ \vee \ H_n(c, w)$	DLF

axm0_1
axm0_2
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
inv1_4
inv1_5
$\vdash$
Disjunction of concrete guards

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \quad \lor \quad c = 0 \\
\vdash \\
(a + b < d \;\land\; c = 0) \;\lor \\
c > 0 \quad \lor \quad a > 0 \\
(b > 0 \;\land\; a = 0)
\end{array}
$$

DLF

ML_out
  when
    $a + b < d$
    $c = 0$
  then
    $a := a + 1$
  end

ML_in
  when
    $c > 0$
  then
    $c := c - 1$
  end

IL_in
  when
    $a > 0$
  then
    $a := a - 1$
    $b := b + 1$
  end

IL_out
  when
    $b > 0$
    $a = 0$
  then
    $b := b - 1$
    $c := c + 1$
  end

More Inference Rule

$$\frac{\mathbf{H}, \neg\,\mathbf{P} \;\vdash\; \mathbf{Q}}{\mathbf{H} \;\vdash\; \mathbf{P} \vee \mathbf{Q}} \quad \text{NEG}$$

$$\frac{\mathbf{H}, \mathbf{P}, \mathbf{Q} \;\vdash\; \mathbf{R}}{\mathbf{H},\; \mathbf{P} \wedge \mathbf{Q} \;\vdash\; \mathbf{R}} \quad \text{AND\_L}$$

$$\frac{\mathbf{H} \;\vdash\; \mathbf{P} \qquad \mathbf{H} \;\vdash\; \mathbf{Q}}{\mathbf{H} \;\vdash\; \mathbf{P} \wedge \mathbf{Q}} \quad \text{AND\_R}$$

$$\begin{array}{l} d \in \mathbb{N} \\ 0 < d \\ n \in \mathbb{N} \\ n \leq d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ c \in \mathbb{N} \\ a + b + c = n \\ a = 0 \ \vee \ c = 0 \\ \vdash \\ (a + b < d \ \wedge \ c = 0) \ \vee \\ c > 0 \ \vee \\ a > 0 \ \vee \\ (b > 0 \ \wedge \ a = 0) \end{array}$$

MON

$$\begin{array}{l} 0 < d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ c \in \mathbb{N} \\ \vdash \\ (a + b < d \ \wedge \ c = 0) \ \vee \\ c > 0 \ \vee \\ a > 0 \ \vee \\ (b > 0 \ \wedge \ a = 0) \end{array}$$

NEG

$$\begin{array}{l} 0 < d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ \underline{c \in \mathbb{N}} \\ \underline{\neg\,(c > 0)} \\ \vdash \\ (a + b < d \ \wedge \ c = 0) \ \vee \\ a > 0 \ \vee \\ (b > 0 \ \wedge \ a = 0) \end{array}$$

ARITH $\cdots$

$$\begin{array}{l} d \in \mathbb{N} \\ 0 < d \\ n \in \mathbb{N} \\ n \leq d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ c \in \mathbb{N} \\ a + b + c = n \\ a = 0 \;\vee\; c = 0 \\ \vdash \\ (a + b < d \;\wedge\; c = 0) \;\vee \\ c > 0 \;\vee \\ a > 0 \;\vee \\ (b > 0 \;\wedge\; a = 0) \end{array}$$

MON

$$\begin{array}{l} 0 < d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ c \in \mathbb{N} \\ \vdash \\ (a + b < d \;\wedge\; c = 0) \;\vee \\ c > 0 \;\vee \\ a > 0 \;\vee \\ (b > 0 \;\wedge\; a = 0) \end{array}$$

NEG

$$\begin{array}{l} 0 < d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ \underline{c \in \mathbb{N}} \\ \underline{\neg (c > 0)} \\ \vdash \\ (a + b < d \;\wedge\; c = 0) \;\vee \\ a > 0 \;\vee \\ (b > 0 \;\wedge\; a = 0) \end{array}$$

ARITH $\cdots$

$\cdots$

$$\begin{array}{l} 0 < d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ c = 0 \\ \vdash \\ (a + b < d \;\wedge\; c = 0) \;\vee \\ a > 0 \;\vee \\ (b > 0 \;\wedge\; a = 0) \end{array}$$

EQ_LR

$$\begin{array}{l} 0 < d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ \vdash \\ (a + b < d \;\wedge\; 0 = 0) \;\vee \\ a > 0 \;\vee \\ (b > 0 \;\wedge\; a = 0) \end{array}$$

NEG

$$\begin{array}{l} 0 < d \\ \underline{a \in \mathbb{N}} \\ \underline{b \in \mathbb{N}} \\ \underline{\neg (a > 0)} \\ \vdash \\ (a + b < d \;\wedge\; 0 = 0) \;\vee \\ (b > 0 \;\wedge\; a = 0) \end{array}$$

ARITH $\cdots$

Proof (cont’d)

$$
\cdots \;
\boxed{\begin{array}{l}
\;\; 0 < d \\
\;\; a = 0 \\
\;\; b \in \mathbb{N} \\
\vdash \\
\;\; (a + b < d \ \wedge\ 0 = 0) \ \vee \\
\;\; (b > 0 \ \wedge\ a = 0)
\end{array}}
\ \text{EQ\_LR}\
\boxed{\begin{array}{l}
\;\; 0 < d \\
\;\; \underline{b \in \mathbb{N}} \\
\vdash \\
\;\; (\underline{0 + b} < d \ \wedge\ 0 = 0) \ \vee \\
\;\; (b > 0 \ \wedge\ 0 = 0)
\end{array}}
\ \text{ARITH}\
\boxed{\begin{array}{l}
\;\; 0 < d \\
\;\; b = 0 \ \vee\ b > 0 \\
\vdash \\
\;\; (b < d \ \wedge\ 0 = 0) \ \vee \\
\;\; (b > 0 \ \wedge\ 0 = 0)
\end{array}}
\ \textbf{OR\_L} \cdots
$$

Proof (cont'd)

$\ldots$

$$\begin{array}{l} 0 < d \\ a = 0 \\ b \in \mathbb{N} \\ \vdash \\ (a + b < d \ \wedge \ 0 = 0) \ \vee \\ (b > 0 \ \wedge \ a = 0) \end{array}$$

EQ_LR

$$\begin{array}{l} 0 < d \\ \underline{b \in \mathbb{N}} \\ \vdash \\ (\underline{0 + b} < d \ \wedge \ 0 = 0) \ \vee \\ (b > 0 \ \wedge \ 0 = 0) \end{array}$$

ARITH

$$\begin{array}{l} 0 < d \\ b = 0 \ \vee \ b > 0 \\ \vdash \\ (b < d \ \wedge \ 0 = 0) \ \vee \\ (b > 0 \ \wedge \ 0 = 0) \end{array}$$

OR_L $\ldots$

$\ldots$

$$\begin{array}{l} 0 < d \\ b = 0 \\ \vdash \\ (b < d \ \wedge \ 0 = 0) \ \vee \\ (b > 0 \ \wedge \ 0 = 0) \end{array}$$

OR_R1

$$\begin{array}{l} 0 < d \\ b = 0 \\ \vdash \\ b < d \ \wedge \ 0 = 0 \end{array}$$

EQ_LR

$$\begin{array}{l} 0 < d \\ \vdash \\ 0 < d \ \wedge \ 0 = 0 \end{array}$$

AND_R

$$\begin{array}{l} 0 < d \\ \vdash \\ 0 < d \end{array}$$

HYP

$$\begin{array}{l} 0 < d \\ \vdash \\ 0 = 0 \end{array}$$

EQL

$$\begin{array}{l} 0 < d \\ b > 0 \\ \vdash \\ (b < d \ \wedge \ 0 = 0) \ \vee \\ (b > 0 \ \wedge \ 0 = 0) \end{array}$$

OR_R2

$$\begin{array}{l} 0 < d \\ b > 0 \\ \vdash \\ b > 0 \ \wedge \ 0 = 0 \end{array}$$

MON

$$\begin{array}{l} b > 0 \\ \vdash \\ b > 0 \ \wedge \ 0 = 0 \end{array}$$

AND_R

$$\begin{array}{l} b > 0 \\ \vdash \\ b > 0 \end{array}$$

HYP

$$\begin{array}{l} b > 0 \\ \vdash \\ 0 = 0 \end{array}$$

EQL

What we Proved

- ML_out / GRD done
- ML_in / GRD done
- ML_out / **inv1_4** / INV done
- ML_out / **inv1_5** / INV done
- ML_in / **inv1_4** / INV done
- ML_in / **inv1_5** / INV done
- **inv1_4** / INV done
- **inv1_5** / INV done
- IL_in / **inv1_4** / INV done
- IL_in / **inv1_5** / INV done
- IL_out / **inv1_4** / INV done
- IL_out / **inv1_5** / INV done
- NAT done
- IL_in / VAR done
- IL_out / VAR done
- DLF done

What we Proved

- For old events:
  - Strengthening of guards: rule GRD
  - Concrete invariant preservation: rule INV
- For new events:
  - Refining the implicit skip event: rule INV
  - Absence of divergence: rules NAT and VAR
- For all events:
  - Relative deadlock freeness: rule DLF

Refinement Rules (1)

Axioms Abstract invariants Concrete invariants Concrete guards $\vdash$ Abstract guard	GRD

Axioms Abstract invariants Concrete invariants Concrete guard $\vdash$ Modified concrete invariant	INV

Axioms $\vdash$ Modified concrete invariant	INV

Refinement Rules (2)

Axioms Abstract invariants Concrete invariants Concrete guards of a new event $\vdash$ Variant $\in \mathbb{N}$	NAT

Axioms Abstract invariants Concrete invariants Concrete guards of a new event $\vdash$ Modified variant $<$ Variant	VAR

Axioms Abstract invariants Concrete invariants Disjunction of abstract events guards $\vdash$ Disjunction of concrete events guards	DLF

variables: a, b, c

inv1_1: $a \in \mathbb{N}$

inv1_2: $b \in \mathbb{N}$

inv1_3: $c \in \mathbb{N}$

inv1_4: $a + b + c = n$

inv1_5: $a = 0 \;\vee\; c = 0$

variant1: $2 * a + b$

Events of the First Refinement

```
init
  a := 0
  b := 0
  c := 0
```

```
ML_in
  when
    0 < c
  then
    c := c − 1
  end
```

```
ML_out
  when
    a + b < d
    c = 0
  then
    a := a + 1
  end
```

```
IL_in
  when
    0 < a
  then
    a := a − 1
    b := b + 1
  end
```

```
IL_out
  when
    0 < b
    a = 0
  then
    b := b − 1
    c := c + 1
  end
```

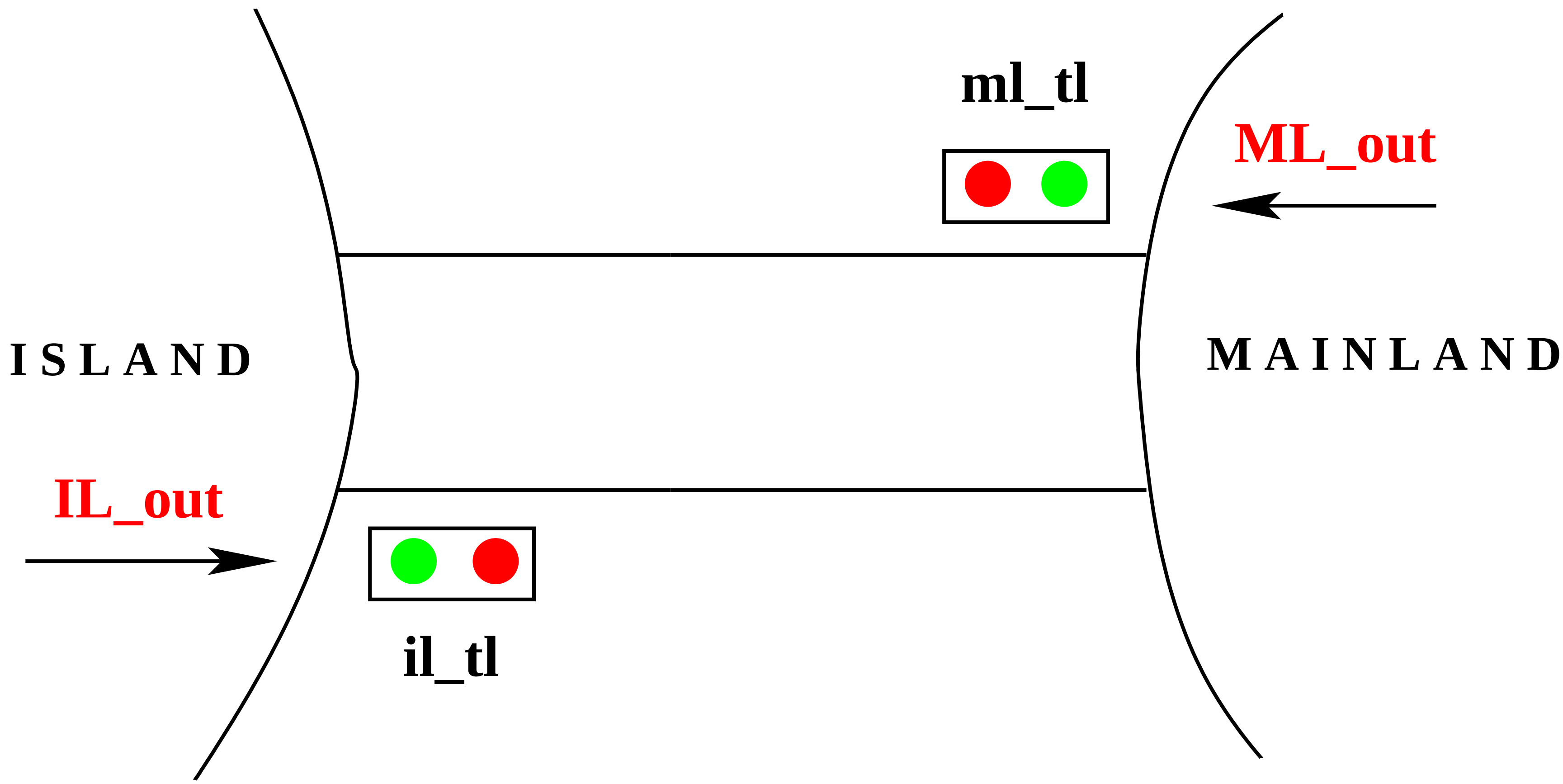
- Initial model: Limiting the number of cars (FUN_2)

- First refinement: Introducing the one way bridge (FUN_3)

- Second refinement: Introducing the traffic lights (EQP_1,2,3)

- Third refinement: Introducing the sensors (EQP_4,5)

Second Refinement: Introducing Traffic Lights



set: $COLOR$

constants: $red, green$

axm2_1: $COLOR = \{green, red\}$

axm2_2: $green \neq red$

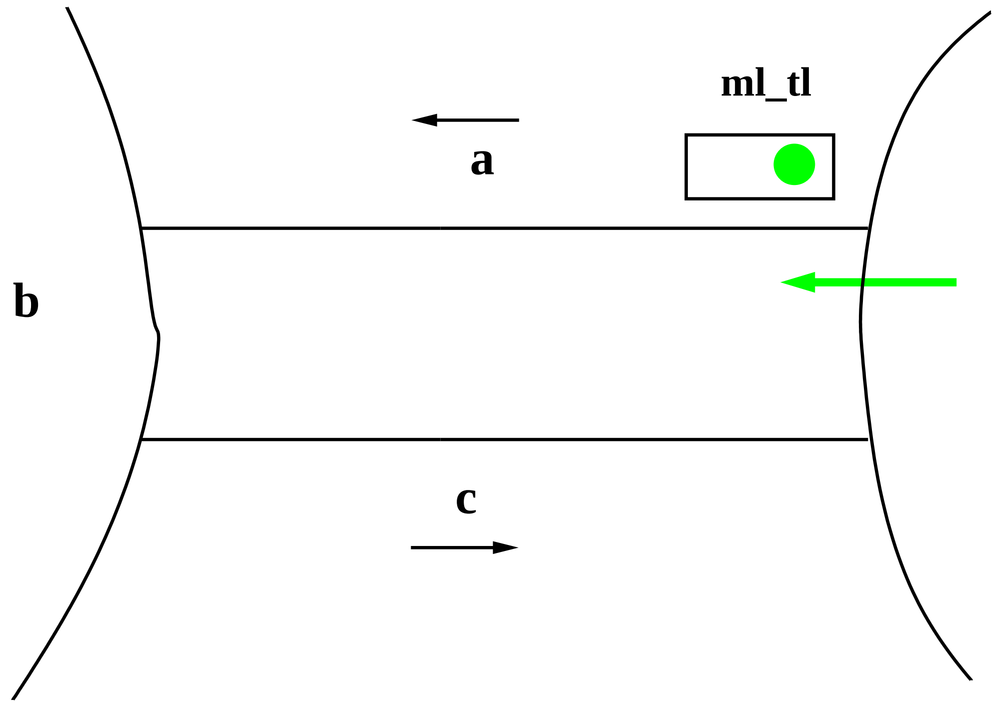
Extending the Variables

$$il\_tl \in COLOR$$

$$ml\_tl \in COLOR$$

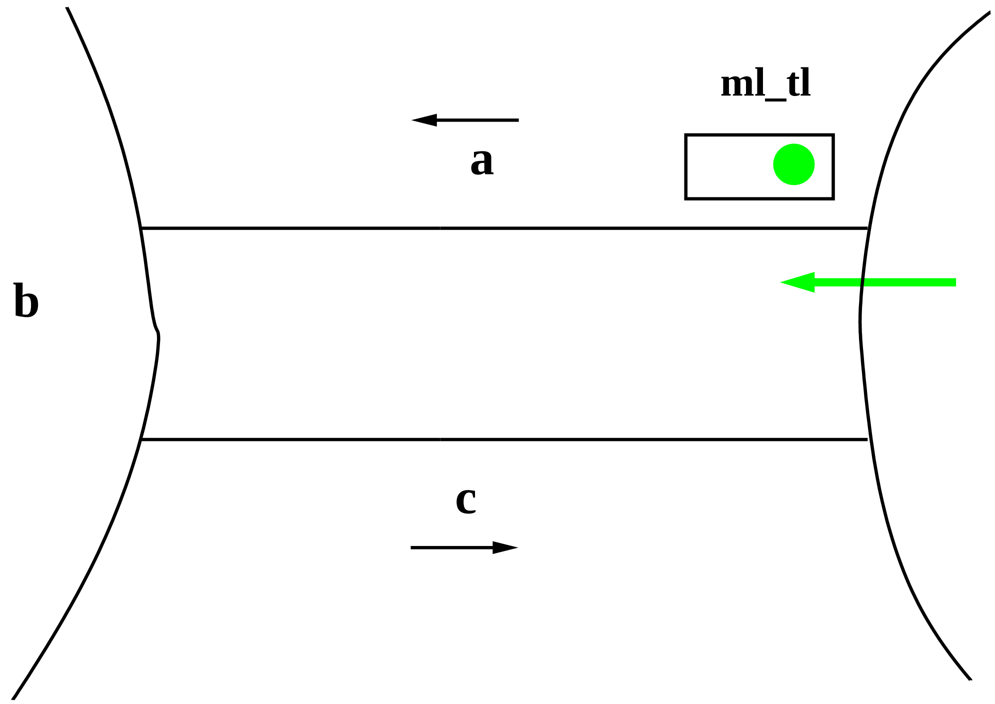
Remark: Events IL_in and ML_in are not modified in this refinement

Extending the Invariant (1)



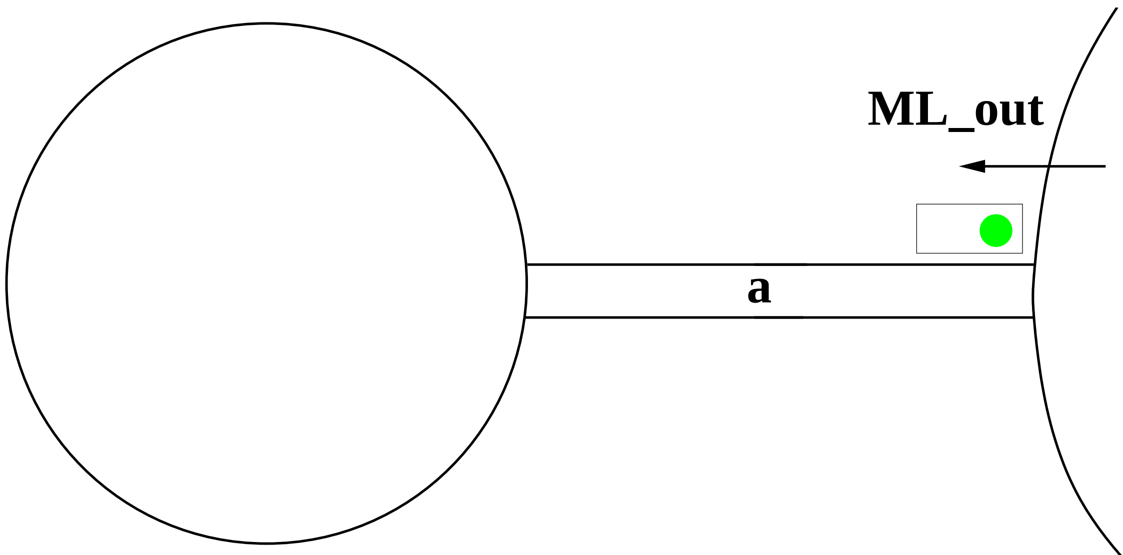
- A green mainland traffic light implies safe access to the bridge

Extending the Invariant (1)

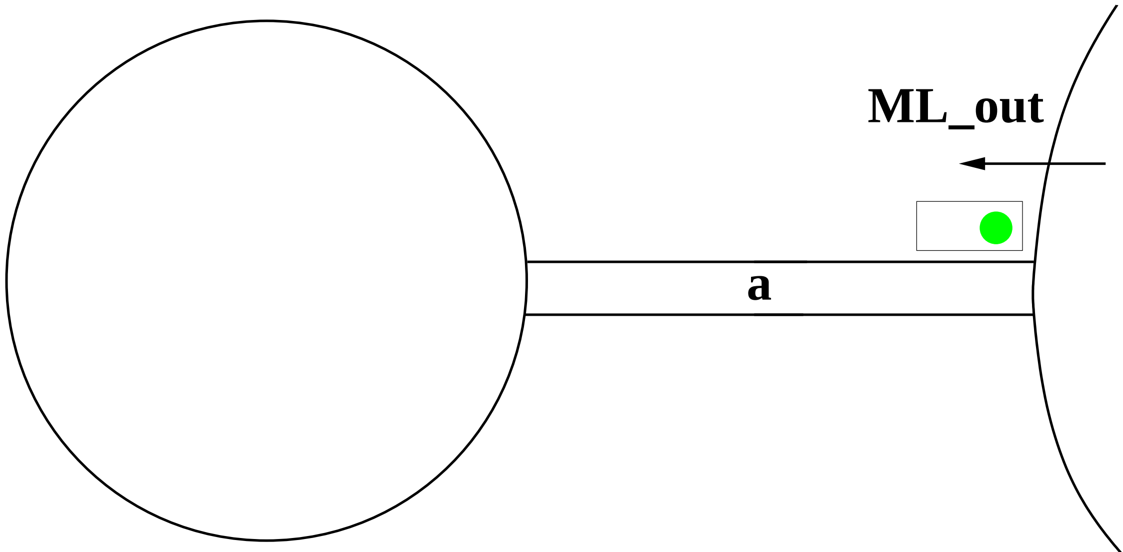


- A green mainland traffic light implies safe access to the bridge

$$ml\_tl = \mathbf{green} \;\Rightarrow\; c = 0 \;\wedge\; a + b < d$$



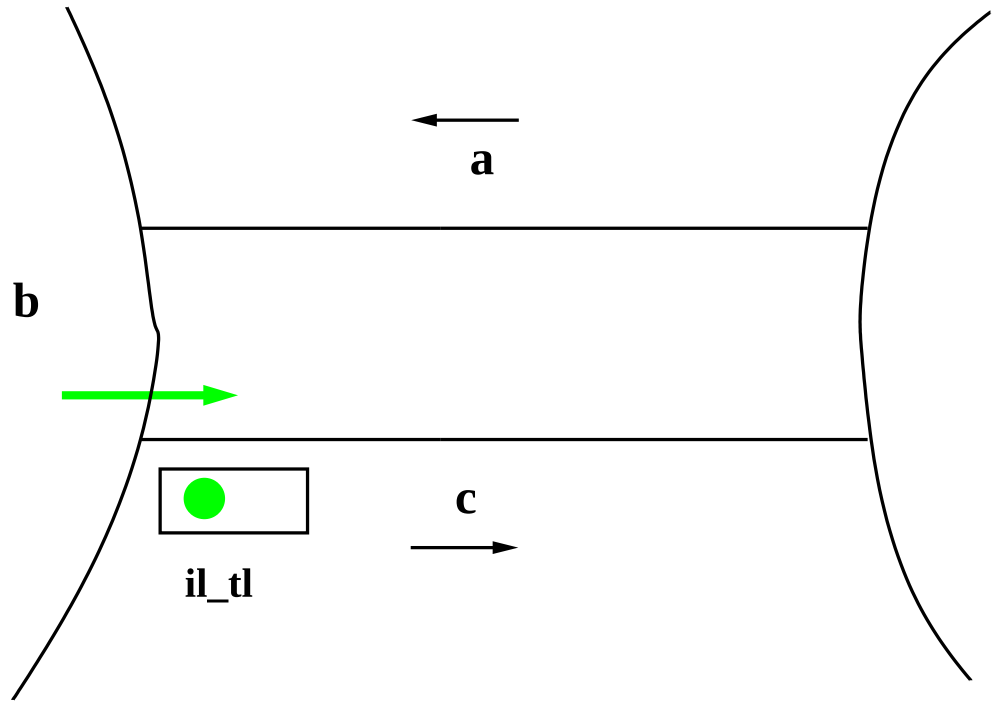
(abstract_)ML_out
when
$c = 0$
$a + b < d$
then
$a := a + 1$
end



(abstract_)ML_out
when
$c = 0$
$a + b < d$
then
$a := a + 1$
end

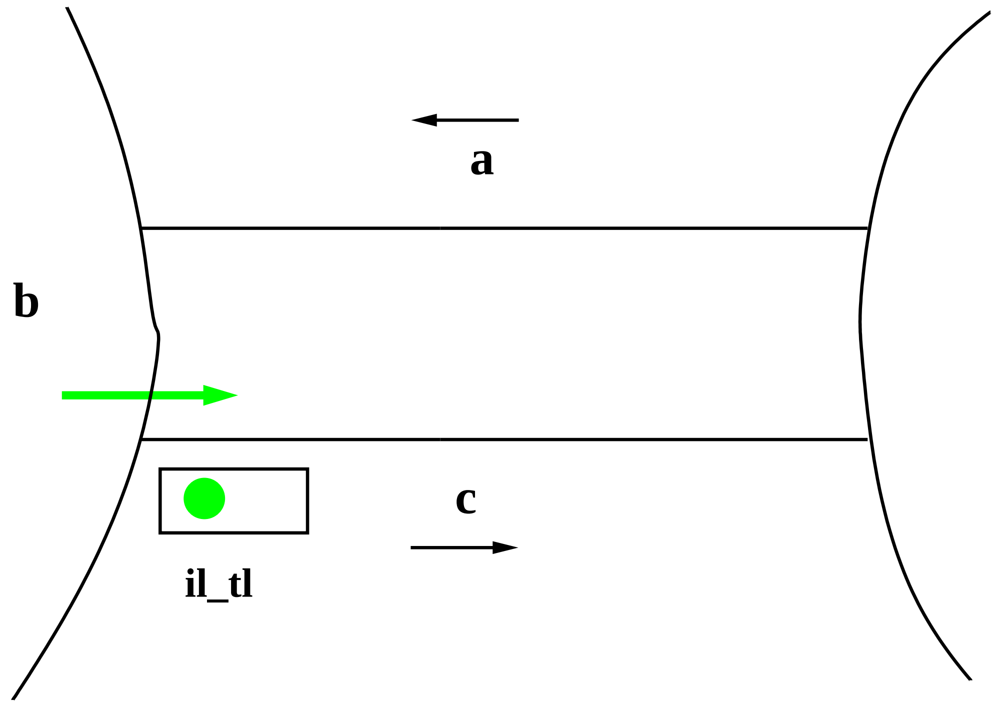
(concrete_)ML_out
when
$ml\_tl = \mathbf{green}$
then
$a := a + 1$
end

Extending the Invariant (2)



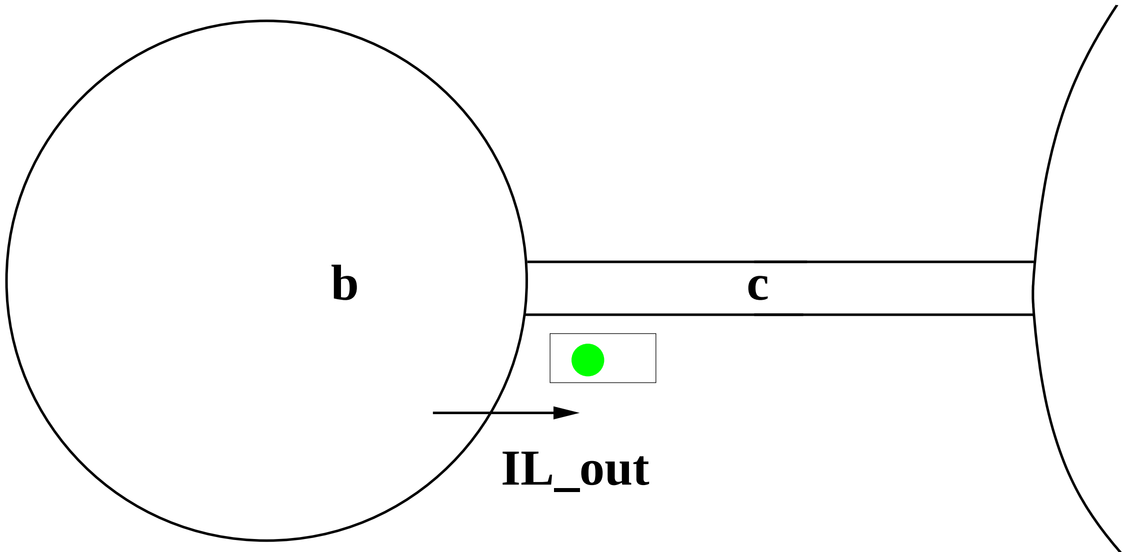
- A green island traffic light implies safe access to the bridge

Extending the Invariant (2)

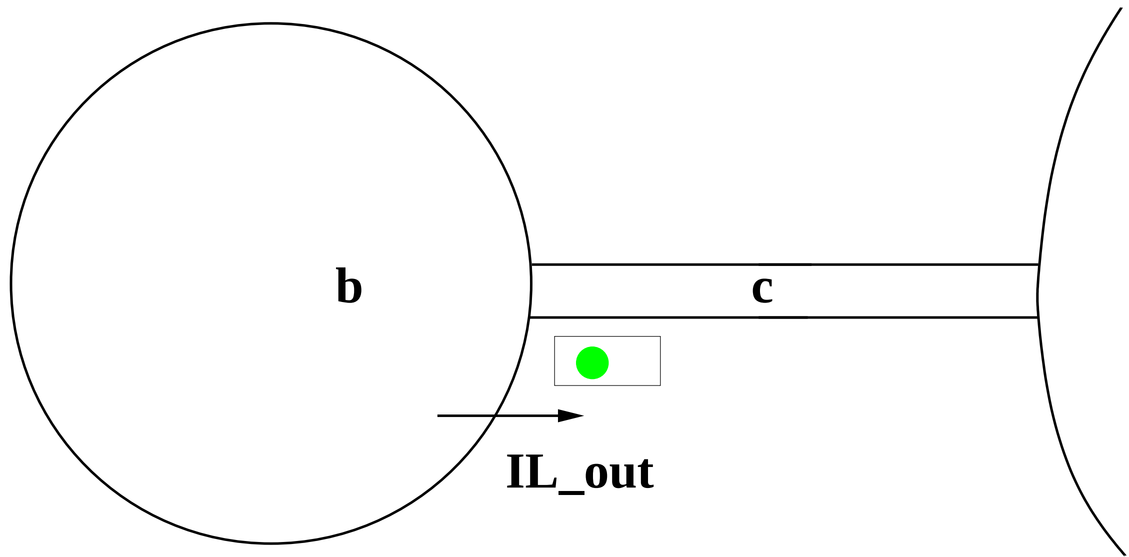


- A green island traffic light implies safe access to the bridge

$$\boxed{il\_tl = \mathbf{green} \;\Rightarrow\; a = 0 \;\wedge\; 0 < b}$$



(abstract_)IL_out
when
$a = 0$
$0 < b$
then
$b, c := b - 1, c + 1$
end



(abstract_)IL_out
  when
    $a = 0$
    $0 < b$
  then
    $b, c := b - 1, c + 1$
  end

(concrete_)IL_out
  when
    $il\_tl =$ **green**
  then
    $b, c := b - 1, c + 1$
  end

New Events ML_tl_green and IL_tl_green

ML_tl_green
when
$ml\_tl = \textbf{red}$
$c = 0$
$a + b < d$
then
$ml\_tl := \textbf{green}$
end

IL_tl_green
when
$il\_tl = \textbf{red}$
$a = 0$
$0 < b$
then
$il\_tl := \textbf{green}$
end

- Turning lights to green when proper conditions hold

Summary of State Refinement (so far)

variables: a, b, c, ml_tl, il_tl

inv2_1: $ml\_tl \in COLOR$

inv2_2: $il\_tl \in COLOR$

inv2_3: $ml\_tl = \textbf{green} \Rightarrow a + b < d \wedge c = 0$

inv2_4: $il\_tl = \textbf{green} \Rightarrow 0 < b \wedge a = 0$

ML_out
  when
    $ml\_tl = \textbf{green}$
  then
    $a := a + 1$
  end

IL_out
  when
    $il\_tl = \textbf{green}$
  then
    $b := b - 1$
    $c := c + 1$
  end

Events ML_in and IL_ in are unchanged

ML_in
  when
    $0 < c$
  then
    $c := c - 1$
  end

IL_in
  when
    $0 < a$
  then
    $a := a - 1$
    $b := b + 1$
  end

variables: a, b, c, ml_tl, il_tl

- Variables a, b, and c were present in the previous refinement

- Variables ml_tl and il_tl are superposed to a, b, and c

- We have thus to extend rule INV

Superposition: Introduction of a new Rule

Abstract_Event
when
$G(c, u, v)$
then
$u := E(c, u, v)$
$v := M(c, u, v)$
end

Concrete_Event
when
$H(c, v, w)$
then
$v := N(c, v, w)$
$w := F(c, v, w)$
end

Axioms Abstract invariants Concrete invariants Concrete guards $\Rightarrow$ Same actions on common variables	$A(c)$ $I(c, u, v)$ $J(c, u, v, w)$ $H(c, v, w)$ $\Rightarrow$ $M(c, u, v) = N(c, v, w)$	SIM

- We have to apply 3 Proof Obligations:
  - GRD,
  - SIM,
  - INV

- On 4 events: ML_out, IL_out, ML_in, IL_in

- And 2 main invariants:

inv2_3:	$ml\_tl = \textbf{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0$
inv2_4:	$il\_tl = \textbf{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0$

Proving the Refinement of the Four Old Events

ML_out
when
$c = 0$
$a + b < d$
then
$a := a + 1$
end

IL_out
when
$a = 0$
$0 < b$
then
$b := b - 1$
$c := c + 1$
end

ML_in
when
$0 < c$
then
$c := c - 1$
end

IL_in
when
$0 < a$
then
$a := a - 1$
$b := b + 1$
end

ML_out
when
$ml\_tl = \textbf{green}$
then
$a := a + 1$
end

IL_out
when
$il\_tl = \textbf{green}$
then
$b := b - 1$
$c := c + 1$
end

ML_in
when
$0 < c$
then
$c := c - 1$
end

IL_in
when
$0 < a$
then
$a := a - 1$
$b := b + 1$
end

- SIM is completely trivial since the actions are the same

ML_out	IL_out	ML_in	IL_in
when	**when**	**when**	**when**
$c = 0$	$a = 0$	$0 < c$	$0 < a$
$a + b < d$	$0 < b$	**then**	**then**
then	**then**	$c := c - 1$	$a := a - 1$
$a := a + 1$	$b := b - 1$	**end**	$b := b + 1$
end	$c := c + 1$		**end**
	end		

ML_out	IL_out	ML_in	IL_in
when	**when**	**when**	**when**
$ml\_tl = \textbf{green}$	$il\_tl = \textbf{green}$	$0 < c$	$0 < a$
then	**then**	**then**	**then**
$a := a + 1$	$b := b - 1$	$c := c - 1$	$a := a - 1$
end	$c := c + 1$	**end**	$b := b + 1$
	end		**end**

- GRD is also trivial

inv2_3: $ml\_tl = \textbf{green} \Rightarrow a + b < d \wedge c = 0$

inv2_4: $il\_tl = \textbf{green} \Rightarrow 0 < b \wedge a = 0$

Proving the Refinement of the Four Old Events

ML_out	IL_out	ML_in	IL_in
when $c = 0$, $a + b < d$ **then** $a := a + 1$ **end**	**when** $a = 0$, $0 < b$ **then** $b := b - 1$, $c := c + 1$ **end**	**when** $0 < c$ **then** $c := c - 1$ **end**	**when** $0 < a$ **then** $a := a - 1$, $b := b + 1$ **end**

ML_out	IL_out	ML_in	IL_in
when $ml\_tl = \textbf{green}$ **then** $a := a + 1$ **end**	**when** $il\_tl = \textbf{green}$ **then** $b := b - 1$, $c := c + 1$ **end**	**when** $0 < c$ **then** $c := c - 1$ **end**	**when** $0 < a$ **then** $a := a - 1$, $b := b + 1$ **end**

- INV applied to ML_in and IL_in holds trivially

inv2_3: $ml\_tl = \textbf{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0$

inv2_4: $il\_tl = \textbf{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0$

ML_out	IL_out	ML_in	IL_in
when $c = 0$, $a + b < d$ **then** $a := a + 1$ **end**	**when** $a = 0$, $0 < b$ **then** $b := b - 1$, $c := c + 1$ **end**	**when** $0 < c$ **then** $c := c - 1$ **end**	**when** $0 < a$ **then** $a := a - 1$, $b := b + 1$ **end**

ML_out	IL_out	ML_in	IL_in
when $ml\_tl = \textbf{green}$ **then** $a := a + 1$ **end**	**when** $il\_tl = \textbf{green}$ **then** $b := b - 1$, $c := c + 1$ **end**	**when** $0 < c$ **then** $c := c - 1$ **end**	**when** $0 < a$ **then** $a := a - 1$, $b := b + 1$ **end**

- INV applied to ML_out and IL_out raise some difficulties

What we Have to Prove

- ML_out / **inv2_4** / INV

- IL_out / **inv2_3** / INV

- ML_out / **inv2_3** / INV

- IL_out / **inv2_4** / INV

- Rules about implication

$$\frac{\mathbf{H, P, Q} \;\vdash\; \mathbf{R}}{\mathbf{H,\ P,\ P \Rightarrow Q} \;\vdash\; \mathbf{R}} \quad \mathbf{IMP\_L}$$

$$\frac{\mathbf{H, P} \;\vdash\; \mathbf{Q}}{\mathbf{H} \;\vdash\; \mathbf{P \Rightarrow Q}} \quad \mathbf{IMP\_R}$$

- Rules about negation

$$\frac{\mathbf{H} \;\vdash\; \mathbf{P}}{\mathbf{H,\ \neg P} \;\vdash\; \mathbf{Q}} \quad \mathbf{NOT\_L}$$

$$\frac{\mathbf{H, P} \;\vdash\; \mathbf{Q} \qquad \mathbf{H, P} \;\vdash\; \mathbf{\neg Q}}{\mathbf{H} \;\vdash\; \mathbf{\neg P}} \quad \mathbf{NOT\_R}$$

axm0_1
axm0_2
axm2_1
axm2_2
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
inv1_4
inv1_5
inv2_1
inv2_2
inv2_3
inv2_4
Guard of event ML_out
$\vdash$
Modified invariant **inv2_4**

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
COLOR = \{\mathbf{green}, \mathbf{red}\} \\
\mathbf{green} \neq \mathbf{red} \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \;\lor\; c = 0 \\
ml\_tl \in COLOR \\
il\_tl \in COLOR \\
ml\_tl = \mathbf{green} \;\Rightarrow\; a + b < d \;\land\; c = 0 \\
il\_tl = \mathbf{green} \;\Rightarrow\; 0 < b \;\land\; a = 0 \\
ml\_tl = \mathbf{green} \\
\vdash \\
il\_tl = \mathbf{green} \;\Rightarrow\; 0 < b \;\land\; a + 1 = 0
\end{array}
$$

ML_out / **inv2_4** / INV

ML_out
when
$ml\_tl = \mathbf{green}$
then
$a := a + 1$
end

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
COLOR = \{\textbf{green}, \textbf{red}\} \\
\textbf{green} \neq \textbf{red} \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \ \lor \ c = 0 \\
ml\_tl \in COLOR \\
il\_tl \in COLOR \\
ml\_tl = \textbf{green} \ \Rightarrow \ a + b < d \ \land \ c = 0 \\
il\_tl = \textbf{green} \ \Rightarrow \ 0 < b \ \land \ a = 0 \\
ml\_tl = \textbf{green} \\
\vdash \\
il\_tl = \textbf{green} \ \Rightarrow \ 0 < b \ \land \ a + 1 = 0
\end{array}
$$

MON

$$
\begin{array}{l}
\textbf{green} \neq \textbf{red} \\
il\_tl = \textbf{green} \ \Rightarrow \ 0 < b \ \land \ a = 0 \\
ml\_tl = \textbf{green} \\
\vdash \\
il\_tl = \textbf{green} \ \Rightarrow \ 0 < b \ \land \ a + 1 = 0
\end{array}
$$

IMP_R $\cdots$

$$
\begin{array}{|l|}
\hline
d \;\in\; \mathbb{N} \\
0 < d \\
COLOR = \{\textbf{green}, \textbf{red}\} \\
\textbf{green} \neq \textbf{red} \\
n \;\in\; \mathbb{N} \\
n \leq d \\
a \;\in\; \mathbb{N} \\
b \;\in\; \mathbb{N} \\
c \;\in\; \mathbb{N} \\
a + b + c = n \\
a = 0 \;\vee\; c = 0 \\
ml\_tl \;\in\; COLOR \\
il\_tl \;\in\; COLOR \\
ml\_tl = \textbf{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0 \\
il\_tl = \textbf{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0 \\
ml\_tl = \textbf{green} \\
\vdash \\
il\_tl = \textbf{green} \;\Rightarrow\; 0 < b \;\wedge\; a + 1 = 0 \\
\hline
\end{array}
\quad \text{MON} \quad
\begin{array}{|l|}
\hline
\textbf{green} \neq \textbf{red} \\
il\_tl = \textbf{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0 \\
ml\_tl = \textbf{green} \\
\vdash \\
il\_tl = \textbf{green} \;\Rightarrow\; 0 < b \;\wedge\; a + 1 = 0 \\
\hline
\end{array}
\quad \text{IMP\_R} \cdots
$$

$$
\cdots
\begin{array}{|l|}
\hline
\textbf{green} \neq \textbf{red} \\
il\_tl = \textbf{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0 \\
ml\_tl = \textbf{green} \\
il\_tl = \textbf{green} \\
\vdash \\
0 < b \;\wedge\; a + 1 = 0 \\
\hline
\end{array}
\quad \text{IMP\_L} \quad
\begin{array}{|l|}
\hline
\textbf{green} \neq \textbf{red} \\
0 < b \;\wedge\; a = 0 \\
ml\_tl = \textbf{green} \\
il\_tl = \textbf{green} \\
\vdash \\
0 < b \;\wedge\; a + 1 = 0 \\
\hline
\end{array}
\quad \text{AND\_L} \cdots
$$

$\cdots$

$$\begin{array}{l} \text{green} \neq \text{red} \\ 0 < b \\ a = 0 \\ ml\_tl = \text{green} \\ il\_tl = \text{green} \\ \vdash \\ 0 < b \ \wedge \ a + 1 = 0 \end{array}$$

AND_R

$$\begin{array}{l} \text{green} \neq \text{red} \\ 0 < b \\ a = 0 \\ ml\_tl = \text{green} \\ il\_tl = \text{green} \\ \vdash \\ 0 < b \end{array}$$

MON

$$0 < b \ \vdash \ 0 < b$$

HYP

$$\begin{array}{l} \text{green} \neq \text{red} \\ 0 < b \\ a = 0 \\ ml\_tl = \text{green} \\ il\_tl = \text{green} \\ \vdash \\ a + 1 = 0 \end{array}$$

EQ_LR

$$\begin{array}{l} \text{green} \neq \text{red} \\ ml\_tl = \text{green} \\ il\_tl = \text{green} \\ \vdash \\ \underline{0 + 1 = 0} \end{array}$$

ARITH

$$\begin{array}{l} \text{green} \neq \text{red} \\ ml\_tl = \text{green} \\ il\_tl = \text{green} \\ \vdash \\ 1 = 0 \end{array}$$

?

axm0_1	$d \in \mathbb{N}$
axm0_2	$0 < d$
axm2_1	$COLOR = \{\text{green}, \text{red}\}$
axm2_2	$\text{green} \neq \text{red}$
inv0_1	$n \in \mathbb{N}$
inv0_2	$n \leq d$
inv1_1	$a \in \mathbb{N}$
inv1_2	$b \in \mathbb{N}$
inv1_3	$c \in \mathbb{N}$
inv1_4	$a + b + c = n$
inv1_5	$a = 0 \lor c = 0$
inv2_1	$ml\_tl \in COLOR$
inv2_2	$il\_tl \in COLOR$
inv2_3	$ml\_tl = \text{green} \Rightarrow a + b < d \land c = 0$
inv2_4	$il\_tl = \text{green} \Rightarrow 0 < b \land a = 0$
Guard of IL_out	$il\_tl = \text{green}$
$\vdash$	$\vdash$
Modified **inv2_3**	$ml\_tl = \text{green} \Rightarrow a + b - 1 < d \land c + 1 = 0$

IL_out / **inv2_3** / INV

IL_out
 when
  $il\_tl = \text{green}$
 then
  $b := b - 1$
  $c := c + 1$
 end

$$
\begin{array}{l}
d \ \in \ \mathbb{N} \\
0 < d \\
COLOR = \{\text{green}, \text{red}\} \\
\text{green} \neq \text{red} \\
n \ \in \ \mathbb{N} \\
n \leq d \\
a \ \in \ \mathbb{N} \\
b \ \in \ \mathbb{N} \\
c \ \in \ \mathbb{N} \\
a + b + c = n \\
a = 0 \ \vee \ c = 0 \\
ml\_tl \ \in \ COLOR \\
il\_tl \ \in \ COLOR \\
ml\_tl = \text{green} \ \Rightarrow \ a + b < d \ \wedge \ c = 0 \\
il\_tl = \text{green} \ \Rightarrow \ 0 < b \ \wedge \ a = 0 \\
il\_tl = \text{green} \\
\vdash \\
ml\_tl = \text{green} \ \Rightarrow \ a + b - 1 < d \ \wedge \ c + 1 = 0
\end{array}
$$

MON

$$
\begin{array}{l}
\text{green} \neq \text{red} \\
ml\_tl = \text{green} \ \Rightarrow \ a + b < d \ \wedge \ c = 0 \\
il\_tl = \text{green} \\
\vdash \\
ml\_tl = \text{green} \ \Rightarrow \ a + b - 1 < d \ \wedge \\
\qquad c + 1 = 0
\end{array}
$$

IMP_R $\cdots$

$$
\begin{array}{l}
d \;\in\; \mathbb{N} \\
0 < d \\
COLOR = \{\textbf{green}, \textbf{red}\} \\
\textbf{green} \neq \textbf{red} \\
n \;\in\; \mathbb{N} \\
n \leq d \\
a \;\in\; \mathbb{N} \\
b \;\in\; \mathbb{N} \\
c \;\in\; \mathbb{N} \\
a + b + c = n \\
a = 0 \;\;\vee\;\; c = 0 \\
ml\_tl \;\in\; COLOR \\
il\_tl \;\in\; COLOR \\
ml\_tl = \textbf{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0 \\
il\_tl = \textbf{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0 \\
il\_tl = \textbf{green} \\
\vdash \\
ml\_tl = \textbf{green} \;\Rightarrow\; a + b - 1 < d \;\wedge\; c + 1 = 0
\end{array}
$$

MON

$$
\begin{array}{l}
\textbf{green} \neq \textbf{red} \\
ml\_tl = \textbf{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0 \\
il\_tl = \textbf{green} \\
\vdash \\
ml\_tl = \textbf{green} \;\Rightarrow\; a + b - 1 < d \;\wedge \\
\qquad c + 1 = 0
\end{array}
$$

IMP_R ⋯

⋯

$$
\begin{array}{l}
\textbf{green} \neq \textbf{red} \\
ml\_tl = \textbf{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0 \\
il\_tl = \textbf{green} \\
ml\_tl = \textbf{green} \\
\vdash \\
a + b - 1 < d \;\wedge\; c + 1 = 0
\end{array}
$$

IMP_L

$$
\begin{array}{l}
\textbf{green} \neq \textbf{red} \\
a + b < d \;\wedge\; c = 0 \\
il\_tl = \textbf{green} \\
ml\_tl = \textbf{green} \\
\vdash \\
a + b - 1 < d \;\wedge \\
c + 1 = 0
\end{array}
$$

AND_L ⋯

Tentative Proof (cont'd)

. . .

green $\neq$ red
$a + b < d$
$c = 0$
$il\_tl =$ green
$ml\_tl =$ green
$\vdash$
$a + b - 1 < d \;\wedge$
$c + 1 = 0$

AND_R

green $\neq$ red
$a + b < d$
$c = 0$
$il\_tl =$ green
$ml\_tl =$ green
$\vdash$
$a + b - 1 < d$

MON

$a + b < d \;\vdash\; a + b - 1 < d$

DEC

green $\neq$ red
$c = 0$
$il\_tl =$ green
$ml\_tl =$ green
$\vdash$
$c + 1 = 0$

EQ_LR

green $\neq$ red
$il\_tl =$ green
$ml\_tl =$ green
$\vdash$
$\underline{0 + 1 = 0}$

ARITH

green $\neq$ red
$il\_tl =$ green
$ml\_tl =$ green
$\vdash$
$1 = 0$

?

- In both cases, we were stopped by attempting to prove the following

$$\begin{array}{l} \quad \mathbf{green} \neq \mathbf{red} \\ \quad il\_tl = \mathbf{green} \\ \quad ml\_tl = \mathbf{green} \\ \vdash \\ \quad 1 = 0 \end{array}$$

Both traffic lights are assumed to be green!

- This indicates that an "obvious" invariant was missing
- In fact, at least one of the two traffic lights must be red

inv2_5: $ml\_tl = \mathbf{red} \quad \lor \quad il\_tl = \mathbf{red}$

Completing the Proof

$$\begin{array}{l} \textbf{green} \neq \textbf{red} \\ \color{red}{ml\_tl = \textbf{red} \quad \vee \quad il\_tl = \textbf{red}} \\ il\_tl = \textbf{green} \\ ml\_tl = \textbf{green} \\ \vdash \\ 1 = 0 \end{array} \quad \text{OR\_L} \left\{ \begin{array}{l} \begin{array}{l} \textbf{green} \neq \textbf{red} \\ \color{red}{ml\_tl = \textbf{red}} \\ il\_tl = \textbf{green} \\ ml\_tl = \textbf{green} \\ \vdash \\ 1 = 0 \end{array} \quad \textbf{EQ\_LR} \quad \begin{array}{l} \textbf{green} \neq \textbf{red} \\ \textbf{green} = \textbf{red} \\ il\_tl = \textbf{green} \\ \vdash \\ 1 = 0 \end{array} \quad \textbf{NOT\_L} \ \cdots \\ \\ \begin{array}{l} \textbf{green} \neq \textbf{red} \\ \color{red}{il\_tl = \textbf{red}} \\ il\_tl = \textbf{green} \\ ml\_tl = \textbf{green} \\ \vdash \\ 1 = 0 \end{array} \quad \textbf{EQ\_LR} \quad \begin{array}{l} \textbf{green} \neq \textbf{red} \\ \textbf{green} = \textbf{red} \\ ml\_tl = \textbf{green} \\ \vdash \\ 1 = 0 \end{array} \quad \textbf{NOT\_L} \ \cdots \end{array} \right.$$

$$\left\{ \begin{array}{l} \cdots \quad \begin{array}{l} \textbf{green} = \textbf{red} \\ il\_tl = \textbf{green} \\ \vdash \\ \textbf{green} = \textbf{red} \end{array} \quad \textbf{MON} \quad \textbf{green} = \textbf{red} \ \vdash \ \textbf{green} = \textbf{red} \quad \textbf{HYP} \\ \\ \cdots \quad \begin{array}{l} \textbf{green} = \textbf{red} \\ ml\_tl = \textbf{green} \\ \vdash \\ \textbf{green} = \textbf{red} \end{array} \quad \textbf{MON} \quad \textbf{green} = \textbf{red} \ \vdash \ \textbf{green} = \textbf{red} \quad \textbf{HYP} \end{array} \right.$$

$$\textbf{inv2\_5:} \quad ml\_tl = \textbf{red} \;\; \lor \;\; il\_tl = \textbf{red}$$

This could have been deduced from these requirements

The bridge is one way or the other, not both at the same time	FUN-3

Cars are not supposed to pass on a red traffic light, only on a green one	EQP-3

What we Have to Prove

- ML_out / **inv2_4** / INV done

- IL_out / **inv2_3** / INV done

- ML_out / **inv2_3** / INV

- IL_out / **inv2_4** / INV

- ML_tl_green / **inv2_5** / INV

- IL_tl_green / **inv2_5** / INV

Proving Preservation of inv2_3 by Event ML_out

axm0_1	$d \in \mathbb{N}$
axm0_2	$0 < d$
axm2_1	$COLOR = \{\text{green}, \text{red}\}$
axm2_2	$\text{green} \neq \text{red}$
inv0_1	$n \in \mathbb{N}$
inv0_2	$n \leq d$
inv1_1	$a \in \mathbb{N}$
inv1_2	$b \in \mathbb{N}$
inv1_3	$c \in \mathbb{N}$
inv1_4	$a + b + c = n$
inv1_5	$a = 0 \lor c = 0$
inv2_1	$ml\_tl \in COLOR$
inv2_2	$il\_tl \in COLOR$
inv2_3	$ml\_tl = \text{green} \Rightarrow a + b < d \land c = 0$
inv2_4	$il\_tl = \text{green} \Rightarrow 0 < b \land a = 0$
Guard of ML_out	$ml\_tl = \text{green}$
$\vdash$	$\vdash$
Modified **inv2_3**	$ml\_tl = \text{green} \Rightarrow a + 1 + b < d \land c = 0$

ML_out / inv2_3 / **INV**

ML_out
when
$ml\_tl = \text{green}$
then
$a := a + 1$
end

$$
\begin{array}{l}
d \;\in\; \mathbb{N} \\
0 < d \\
COLOR = \{\textsf{green}, \textsf{red}\} \\
\textsf{green} \neq \textsf{red} \\
n \;\in\; \mathbb{N} \\
n \leq d \\
a \;\in\; \mathbb{N} \\
b \;\in\; \mathbb{N} \\
c \;\in\; \mathbb{N} \\
a + b + c = n \\
a = 0 \;\;\vee\;\; c = 0 \\
ml\_tl \;\in\; COLOR \\
il\_tl \;\in\; COLOR \\
ml\_tl = \textsf{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0 \\
il\_tl = \textsf{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0 \\
ml\_tl = \textsf{green} \\
\vdash \\
\quad ml\_tl = \textsf{green} \;\Rightarrow\; a + 1 + b < d \;\wedge\; c = 0
\end{array}
$$

MON

$$
\begin{array}{l}
\quad ml\_tl = \textsf{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0 \\
\vdash \\
\quad ml\_tl = \textsf{green} \;\Rightarrow\; a + 1 + b < d \;\wedge\; c = 0
\end{array}
$$

IMP_R $\cdots$

$$
\begin{array}{l}
d \;\in\; \mathbb{N} \\
0 < d \\
COLOR = \{\text{green}, \text{red}\} \\
\text{green} \neq \text{red} \\
n \;\in\; \mathbb{N} \\
n \leq d \\
a \;\in\; \mathbb{N} \\
b \;\in\; \mathbb{N} \\
c \;\in\; \mathbb{N} \\
a + b + c = n \\
a = 0 \;\;\lor\;\; c = 0 \\
ml\_tl \;\in\; COLOR \\
il\_tl \;\in\; COLOR \\
ml\_tl = \text{green} \;\Rightarrow\; a + b < d \;\land\; c = 0 \\
il\_tl = \text{green} \;\Rightarrow\; 0 < b \;\land\; a = 0 \\
ml\_tl = \text{green} \\
\vdash \\
\quad ml\_tl = \text{green} \;\Rightarrow\; a + 1 + b < d \;\land \\
\qquad\qquad\qquad\qquad\qquad\qquad c = 0
\end{array}
$$

MON

$$
\begin{array}{l}
ml\_tl = \text{green} \;\Rightarrow\; a + b < d \;\land\; c = 0 \\
\vdash \\
ml\_tl = \text{green} \;\Rightarrow\; a + 1 + b < d \;\land\; c = 0
\end{array}
$$

IMP_R $\cdots$

$\cdots$

$$
\begin{array}{l}
ml\_tl = \text{green} \;\;\Rightarrow\;\; a + b < d \;\;\land\;\; c = 0 \\
ml\_tl = \text{green} \\
\vdash \\
a + 1 + b < d \;\;\land\;\; c = 0
\end{array}
$$

IMP_L

$$
\begin{array}{l}
a + b < d \;\;\land\;\; c = 0 \\
ml\_tl = \text{green} \\
\vdash \\
a + 1 + b < d \;\;\land\;\; c = 0
\end{array}
$$

AND_L $\cdots$

Tentative Proof (cont'd)

$$\ldots\quad \boxed{\begin{array}{l} a+b<d \\ c=0 \\ ml\_tl = \textbf{green} \\ \vdash \\ a+1+b<d \;\;\wedge\;\; c=0 \end{array}} \quad \text{AND\_R} \left\{ \begin{array}{l} \boxed{\begin{array}{l} a+b<d \\ c=0 \\ ml\_tl = \textbf{green} \\ \vdash \\ a+1+b<d \end{array}} \;\; \textbf{MON} \;\; \boxed{\begin{array}{l} a+b<d \\ ml\_tl = \textbf{green} \\ \vdash \\ a+1+b<d \end{array}} \;\; \textbf{?} \\ \\ \boxed{\begin{array}{l} a+b<d \\ c=0 \\ ml\_tl = \textbf{green} \\ \vdash \\ c=0 \end{array}} \;\; \textbf{MON} \;\; \boxed{c=0 \;\;\vdash\;\; c=0} \;\; \textbf{HYP} \end{array} \right.$$

- This requires splitting the ML_out in two separate events ML_out_1 and ML_out_2

ML_out_1
when
$ml\_tl = \textbf{green}$
$a+1+b<d$
then
$a := a+1$
end

ML_out_2
when
$ml\_tl = \textbf{green}$
$a+1+b=d$
then
$a := a+1$
$ml\_tl := \textbf{red}$
end

ML_out_1
when
$ml\_tl = \mathbf{green}$
$a + 1 + b < d$
then
$a := a + 1$
end

ML_out_2
when
$ml\_tl = \mathbf{green}$
$a + 1 + b = d$
then
$a := a + 1$
$ml\_tl := \mathbf{red}$
end

- When $a + 1 + b = d$ then only one more car can enter the island

- Consequently, the traffic light ml_tl must be turned red

  (while the car enters the bridge)

axm0_1	$d \in \mathbb{N}$
axm0_2	$0 < d$
axm2_1	$COLOR = \{\text{green}, \text{red}\}$
axm2_2	$\text{green} \neq \text{red}$
inv0_1	$n \in \mathbb{N}$
inv0_2	$n \leq d$
inv1_1	$a \in \mathbb{N}$
inv1_2	$b \in \mathbb{N}$
inv1_3	$c \in \mathbb{N}$
inv1_4	$a + b + c = n$
inv1_5	$a = 0 \quad \vee \quad c = 0$
inv2_1	$ml\_tl \in COLOR$
inv2_2	$il\_tl \in COLOR$
inv2_3	$ml\_tl = \text{green} \Rightarrow a + b < d \wedge c = 0$
inv2_4	$il\_tl = \text{green} \Rightarrow 0 < b \wedge a = 0$
Guard of ML_out_1	$ml\_tl = \text{green}$
	$a + 1 + b < d$
$\vdash$	$\vdash$
Modified **inv2_3**	$ml\_tl = \text{green} \Rightarrow a + 1 + b < d \wedge c = 0$

ML_out_1 / inv2_3 / INV

ML_out_1
  when
    $ml\_tl = \text{green}$
    $a + 1 + b < d$
  then
    $a := a + 1$
  end

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
COLOR = \{\text{green}, \text{red}\} \\
\text{green} \neq \text{red} \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \;\vee\; c = 0 \\
ml\_tl \in COLOR \\
il\_tl \in COLOR \\
ml\_tl = \text{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0 \\
il\_tl = \text{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0 \\
ml\_tl = \text{green} \\
a + 1 + b < d \\
\vdash \\
ml\_tl = \text{green} \;\Rightarrow\; a + 1 + b < d \;\wedge\; c = 0
\end{array}
$$

MON

$$
\begin{array}{l}
ml\_tl = \text{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0 \\
a + 1 + b < d \\
\vdash \\
ml\_tl = \text{green} \;\Rightarrow\; a + 1 + b < d \;\wedge\; c = 0
\end{array}
$$

IMP_R $\cdots$

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
COLOR = \{\textbf{green}, \textbf{red}\} \\
\textbf{green} \neq \textbf{red} \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \ \lor \ c = 0 \\
ml\_tl \in COLOR \\
il\_tl \in COLOR \\
ml\_tl = \textbf{green} \Rightarrow a + b < d \ \land \ c = 0 \\
il\_tl = \textbf{green} \Rightarrow 0 < b \ \land \ a = 0 \\
ml\_tl = \textbf{green} \\
a + 1 + b < d \\
\vdash \\
ml\_tl = \textbf{green} \Rightarrow a + 1 + b < d \ \land \ c = 0
\end{array}
$$

MON

$$
\begin{array}{l}
ml\_tl = \textbf{green} \Rightarrow a + b < d \ \land \ c = 0 \\
a + 1 + b < d \\
\vdash \\
ml\_tl = \textbf{green} \Rightarrow a + 1 + b < d \ \land \ c = 0
\end{array}
$$

IMP_R $\cdots$

$\cdots$

$$
\begin{array}{l}
ml\_tl = \textbf{green} \Rightarrow a + b < d \ \land \ c = 0 \\
ml\_tl = \textbf{green} \\
a + 1 + b < d \\
\vdash \\
a + 1 + b < d \ \land \ c = 0
\end{array}
$$

IMP_L

$$
\begin{array}{l}
a + b < d \ \land \ c = 0 \\
ml\_tl = \textbf{green} \\
a + 1 + b < d \\
\vdash \\
a + 1 + b < d \ \land \ c = 0
\end{array}
$$

AND_L $\cdots$

Proof (cont'd)

$\cdots$

$$\begin{array}{l} a+b<d \\ c=0 \\ ml\_tl = \textbf{green} \\ \color{red}{a+1+b<d} \\ \vdash \\ a+1+b<d \quad \wedge \quad c=0 \end{array}$$

AND_R

$$\begin{array}{l} a+b<d \\ c=0 \\ ml\_tl = \textbf{green} \\ a+1+b<d \\ \vdash \\ a+1+b<d \end{array}$$

MON

$$\begin{array}{l} \color{red}{a+1+b<d} \\ \vdash \\ a+1+b<d \end{array}$$

HYP

$$\begin{array}{l} a+b<d \\ c=0 \\ ml\_tl = \textbf{green} \\ \color{red}{a+1+b<d} \\ \vdash \\ c=0 \end{array}$$

MON

$$c=0 \quad \vdash \quad c=0$$

HYP

Proving Preservation of inv2_3 by Event ML_out_2

axm0_1	$d \in \mathbb{N}$	
axm0_2	$0 < d$	
axm2_1	$COLOR = \{\text{green}, \text{red}\}$	
axm2_2	$\text{green} \neq \text{red}$	
inv0_1	$n \in \mathbb{N}$	
inv0_2	$n \leq d$	
inv1_1	$a \in \mathbb{N}$	
inv1_2	$b \in \mathbb{N}$	
inv1_3	$c \in \mathbb{N}$	
inv1_4	$a + b + c = n$	ML_out_2 / inv2_3 / INV
inv1_5	$a = 0 \;\vee\; c = 0$	
inv2_1	$ml\_tl \in COLOR$	
inv2_2	$il\_tl \in COLOR$	
inv2_3	$ml\_tl = \text{green} \Rightarrow a + b < d \;\wedge\; c = 0$	
inv2_4	$il\_tl = \text{green} \Rightarrow 0 < b \;\wedge\; a = 0$	
Guard of ML_out_2	$ml\_tl = \text{green}$	
	$a + 1 + b = d$	
$\vdash$	$\vdash$	
Modified **inv2_3**	$\text{red} = \text{green} \Rightarrow a + 1 + b < d \;\wedge\; c = 0$	

$$
\begin{array}{l}
\text{ML\_out\_2} \\
\quad \textbf{when} \\
\qquad ml\_tl = \text{green} \\
\qquad a + 1 + b = d \\
\quad \textbf{then} \\
\qquad a := a + 1 \\
\qquad ml\_tl := red \\
\quad \textbf{end}
\end{array}
$$

$d \in \mathbb{N}$
$0 < d$
$COLOR = \{\text{green}, \text{red}\}$
$\text{green} \neq \text{red}$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \;\vee\; c = 0$
$ml\_tl \in COLOR$
$il\_tl \in COLOR$
$ml\_tl = \text{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0$
$il\_tl = \text{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0$
$ml\_tl = \text{green}$
$a + 1 + b = d$
$\vdash$
$\text{red} = \text{green} \;\Rightarrow\; a + 1 + b < d \;\wedge\; c = 0$

MON

$\text{green} \neq \text{red}$
$\vdash$
$\text{red} = \text{green} \;\Rightarrow\; a + 1 + b < d \;\wedge\; c = 0$

IMP_R

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
COLOR = \{\textsf{green}, \textsf{red}\} \\
\textsf{green} \neq \textsf{red} \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \;\vee\; c = 0 \\
ml\_tl \in COLOR \\
il\_tl \in COLOR \\
ml\_tl = \textsf{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0 \\
il\_tl = \textsf{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0 \\
ml\_tl = \textsf{green} \\
a + 1 + b = d \\
\vdash \\
\textsf{red} = \textsf{green} \;\Rightarrow\; a + 1 + b < d \;\wedge\; c = 0
\end{array}
$$

MON

$$
\begin{array}{l}
\textsf{green} \neq \textsf{red} \\
\vdash \\
\textsf{red} = \textsf{green} \;\Rightarrow\; a + 1 + b < d \;\wedge\; c = 0
\end{array}
$$

IMP_R

…

$$
\begin{array}{l}
\textsf{green} \neq \textsf{red} \\
\textsf{red} = \textsf{green} \\
\vdash \\
a + 1 + b < d \;\wedge\; c = 0
\end{array}
$$

EQ_LR

$$
\begin{array}{l}
\textsf{green} \neq \textsf{green} \\
\vdash \\
a + 1 + b < d \;\wedge\; c = 0
\end{array}
$$

NOT_L

$$
\begin{array}{l}
\vdash \\
\textsf{green} = \textsf{green}
\end{array}
$$

EQL

What we Have to Prove

- ML_out / **inv2_4** / INV done

- IL_out / **inv2_3** / INV done

- ML_out / **inv2_3** / INV done

- IL_out / **inv2_4** / INV

- ML_tl_green / **inv2_5** / INV

- IL_tl_green / **inv2_5** / INV

axm0_1 **axm0_2** **axm2_1** **axm2_2** **inv0_1** **inv0_2** **inv1_1** **inv1_2** **inv1_3** **inv1_4** **inv1_5** **inv2_1** **inv2_2** **inv2_3** **inv2_4** Guard of event IL_out $\vdash$ Modified invariant **inv2_4**	$d \in \mathbb{N}$ $0 < d$ $COLOR = \{\text{green}, \text{red}\}$ $\text{green} \neq \text{red}$ $n \in \mathbb{N}$ $n \leq d$ $a \in \mathbb{N}$ $b \in \mathbb{N}$ $c \in \mathbb{N}$ $a + b + c = n$ $a = 0 \;\lor\; c = 0$ $ml\_tl \in COLOR$ $il\_tl \in COLOR$ $ml\_tl = \text{green} \;\Rightarrow\; a + b < d \;\land\; c = 0$ $il\_tl = \text{green} \;\Rightarrow\; 0 < b \;\land\; a = 0$ $il\_tl = \text{green}$ $\vdash$ $il\_tl = \text{green} \;\Rightarrow\; 0 < b - 1 \;\land\; a = 0$	IL_out / **inv2_4** / INV

IL_out
  when
    $il\_tl = \text{green}$
  then
    $b := b - 1$
    $c := c + 1$
  end

$$
\begin{array}{l}
d \;\in\; \mathbb{N} \\
0 < d \\
COLOR = \{\text{green}, \text{red}\} \\
\text{green} \neq \text{red} \\
n \;\in\; \mathbb{N} \\
n \leq d \\
a \;\in\; \mathbb{N} \\
b \;\in\; \mathbb{N} \\
c \;\in\; \mathbb{N} \\
a + b + c = n \\
a = 0 \;\;\lor\;\; c = 0 \\
ml\_tl \;\in\; COLOR \\
il\_tl \;\in\; COLOR \\
ml\_tl = \text{green} \;\Rightarrow\; a + b < d \;\land\; c = 0 \\
il\_tl = \text{green} \;\Rightarrow\; 0 < b \;\land\; a = 0 \\
il\_tl = \text{green} \\
\vdash \\
il\_tl = \text{green} \;\Rightarrow\; 0 < b - 1 \;\land\; a = 0
\end{array}
$$

MON

$$
\begin{array}{l}
il\_tl = \text{green} \;\Rightarrow\; 0 < b \;\land\; a = 0 \\
il\_tl = \text{green} \\
\vdash \\
il\_tl = \text{green} \;\Rightarrow\; 0 < b - 1 \;\land \\
\qquad a = 0
\end{array}
$$

IMP_R

. . .

$$
\begin{array}{l}
il\_tl = \text{green} \;\Rightarrow\; 0 < b \;\land\; a = 0 \\
il\_tl = \text{green} \\
\vdash \\
0 < b - 1 \;\land\; a = 0
\end{array}
$$

IMP_L

$$
\begin{array}{l}
0 < b \;\land\; a = 0 \\
\vdash \\
0 < b - 1 \;\land\; a = 0
\end{array}
$$

AND_L

Tentative Proof (cont’d)

$$\ldots \quad \boxed{\begin{array}{l} 0 < b \\ a = 0 \\ \vdash \\ \quad 0 < b - 1 \ \wedge \ a = 0 \end{array}} \quad \text{AND\_R} \left\{ \begin{array}{l} \boxed{\begin{array}{l} 0 < b \\ a = 0 \\ \vdash \\ \quad 0 < b - 1 \end{array}} \quad \text{MON} \quad \boxed{\begin{array}{l} 0 < b \\ \vdash \\ \quad 0 < b - 1 \end{array}} \quad ? \\ \\ \boxed{\begin{array}{l} 0 < b \\ a = 0 \\ \vdash \\ a = 0 \end{array}} \quad \text{MON} \quad \boxed{\begin{array}{l} a = 0 \\ \vdash \\ a = 0 \end{array}} \quad \text{HYP} \end{array} \right.$$

- This requires splitting the concrete IL_out in two separate events IL_out_1 and IL_out_2

IL_out_1
  when
    $il\_tl = \textbf{green}$
    $b \neq 1$
  then
    $b, c := b - 1, c + 1$
  end

IL_out_2
  when
    $il\_tl = \textbf{green}$
    $b = 1$
  then
    $b, c := b - 1, c + 1$
    $il\_tl := \textbf{red}$
  end

IL_out_1
when
$il\_tl = \text{green}$
$b \neq 1$
then
$b, c := b - 1, c + 1$
end

IL_out_2
when
$il\_tl = \text{green}$
$b = 1$
then
$b, c := b - 1, c + 1$
$il\_tl := \text{red}$
end

- When b=1, then only one car remains in the island

- Consequently, the traffic light il_tl can be turned red
(after this car has left)

axm0_1 **axm0_2** **axm2_1** **axm2_2** **inv0_1** **inv0_2** **inv1_1** **inv1_2** **inv1_3** **inv1_4** **inv1_5** **inv2_1** **inv2_2** **inv2_3** **inv2_4** Guard of event IL_out_1 $\vdash$ Modified invariant **inv2_4**	$d \in \mathbb{N}$ $0 < d$ $COLOR = \{\text{green}, \text{red}\}$ $\text{green} \neq \text{red}$ $n \in \mathbb{N}$ $n \leq d$ $a \in \mathbb{N}$ $b \in \mathbb{N}$ $c \in \mathbb{N}$ $a + b + c = n$ $a = 0 \;\vee\; c = 0$ $ml\_tl \in COLOR$ $il\_tl \in COLOR$ $ml\_tl = \text{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0$ $il\_tl = \text{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0$ $il\_tl = \text{green}$ $b \neq 1$ $\vdash$ $il\_tl = \text{green} \;\Rightarrow\; 0 < b - 1 \;\wedge\; a = 0$	IL_out_1 / **inv2_4** / INV

IL_out_1
  when
    $il\_tl = \text{green}$
    $b \neq 1$
  then
    $b, c := b - 1, c + 1$
  end

$$
\begin{array}{l}
d \;\in\; \mathbb{N} \\
0 < d \\
COLOR = \{\text{green}, \text{red}\} \\
\text{green} \neq \text{red} \\
n \;\in\; \mathbb{N} \\
n \leq d \\
a \;\in\; \mathbb{N} \\
b \;\in\; \mathbb{N} \\
c \;\in\; \mathbb{N} \\
a + b + c = n \\
a = 0 \;\;\vee\;\; c = 0 \\
ml\_tl \;\in\; COLOR \\
il\_tl \;\in\; COLOR \\
ml\_tl = \text{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0 \\
il\_tl = \text{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0 \\
il\_tl = \text{green} \\
b \neq 1 \\
\vdash \\
il\_tl = \text{green} \;\Rightarrow\; 0 < b - 1 \;\wedge\; a = 0
\end{array}
$$

MON

$$
\begin{array}{l}
il\_tl = \text{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0 \\
il\_tl = \text{green} \\
b \neq 1 \\
\vdash \\
il\_tl = \text{green} \;\Rightarrow\; 0 < b - 1 \;\wedge\; a = 0
\end{array}
$$

IMP_R

. . .

$$
\begin{array}{l}
il\_tl = \text{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0 \\
il\_tl = \text{green} \\
b \neq 1 \\
\vdash \\
0 < b - 1 \;\wedge\; a = 0
\end{array}
$$

IMP_L

$$
\begin{array}{l}
0 < b \;\wedge\; a = 0 \\
b \neq 1 \\
\vdash \\
0 < b - 1 \;\wedge\; a = 0
\end{array}
$$

AND_L

Proof (cont'd)

$$\ldots \quad \boxed{\begin{array}{l} 0 < b \\ a = 0 \\ b \neq 1 \\ \vdash \\ \quad 0 < b - 1 \;\; \wedge \;\; a = 0 \end{array}} \;\; \text{AND\_R} \left\{ \begin{array}{l} \boxed{\begin{array}{l} 0 < b \\ a = 0 \\ b \neq 1 \\ \vdash \\ \quad 0 < b - 1 \end{array}} \;\; \text{MON} \;\; \boxed{\begin{array}{l} 0 < b \\ b \neq 1 \\ \vdash \\ 0 < b - 1 \end{array}} \;\; \text{ARITH} \;\; \boxed{\begin{array}{l} 0 < b - 1 \\ \vdash \\ 0 < b - 1 \end{array}} \;\; \text{HYP} \\ \\ \boxed{\begin{array}{l} 0 < b \\ a = 0 \\ b \neq 1 \;\; \vdash \\ a = 0 \end{array}} \;\; \text{MON} \;\; \boxed{\begin{array}{l} a = 0 \\ \vdash \\ a = 0 \end{array}} \;\; \text{HYP} \end{array} \right.$$

axm0_1	$d \in \mathbb{N}$	
axm0_2	$0 < d$	
axm2_1	$COLOR = \{\text{green}, \text{red}\}$	
axm2_2	$\text{green} \neq \text{red}$	
inv0_1	$n \in \mathbb{N}$	
inv0_2	$n \leq d$	
inv1_1	$a \in \mathbb{N}$	
inv1_2	$b \in \mathbb{N}$	
inv1_3	$c \in \mathbb{N}$	
inv1_4	$a + b + c = n$	IL_out_1 / **inv2_4** / INV
inv1_5	$a = 0 \lor c = 0$	
inv2_1	$ml\_tl \in COLOR$	
inv2_2	$il\_tl \in COLOR$	
inv2_3	$ml\_tl = \text{green} \Rightarrow a + b < d \land c = 0$	
inv2_4	$il\_tl = \text{green} \Rightarrow 0 < b \land a = 0$	
Guard of event IL_out_2	$il\_tl = \text{green}$ $b = 1$	
$\vdash$	$\vdash$	
Modified invariant **inv2_4**	$\text{red} = \text{green} \Rightarrow 0 < b - 1 \land a = 0$	

IL_out_2
when
$il\_tl = \text{green}$
$b = 1$
then
$b, c, il\_tl := b - 1, c + 1, red$
end

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
COLOR = \{\text{green}, \text{red}\} \\
\text{green} \neq \text{red} \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \;\vee\; c = 0 \\
ml\_tl \in COLOR \\
il\_tl \in COLOR \\
ml\_tl = \text{green} \;\Rightarrow\; a + b < d \;\wedge\; c = 0 \\
il\_tl = \text{green} \;\Rightarrow\; 0 < b \;\wedge\; a = 0 \\
il\_tl = \text{green} \\
b = 1 \\
\vdash \\
\text{red} = \text{green} \;\Rightarrow\; 0 < b - 1 \;\wedge\; a = 0
\end{array}
$$

MON

$$
\begin{array}{l}
\text{green} \neq \text{red} \\
\vdash \\
\text{red} = \text{green} \;\Rightarrow\; 0 < b - 1 \;\wedge \\
\qquad a = 0
\end{array}
$$

IMP_R

...

$$
\begin{array}{l}
\text{green} \neq \text{red} \\
\text{red} = \text{green} \\
\vdash \\
0 < b - 1 \;\wedge\; a = 0
\end{array}
$$

EQ_LR

$$
\begin{array}{l}
\text{green} \neq \text{green} \\
\vdash \\
0 < b - 1 \;\wedge \\
a = 0
\end{array}
$$

NOT_L

$$
\begin{array}{l}
\vdash \\
\text{green} = \text{green}
\end{array}
$$

EQL

What we Have to Prove

- ML_out / **inv2_4** / INV done

- IL_out / **inv2_3** / INV done

- ML_out / **inv2_3** / INV done

- IL_out / **inv2_4** / INV done

- ML_tl_green / **inv2_5** / INV

- IL_tl_green / **inv2_5** / INV

But the new invariant **inv2_5** is not preserved by the new events

inv2_5: $ml\_tl = \textbf{red} \;\lor\; il\_tl = \textbf{red}$

Unless we correct them as follows:

ML_tl_green
when
$ml\_tl = \textbf{red}$
$a + b < d$
$c = 0$
then
$ml\_tl := \textbf{green}$
$il\_tl := \textbf{red}$
end

IL_tl_green
when
$il\_tl = \textbf{red}$
$0 < b$
$a = 0$
then
$il\_tl := \textbf{green}$
$ml\_tl := \textbf{red}$
end

Summary of the Proof Situation

- Correct event refinement: OK

- Absence of divergence of new events: FAILURE

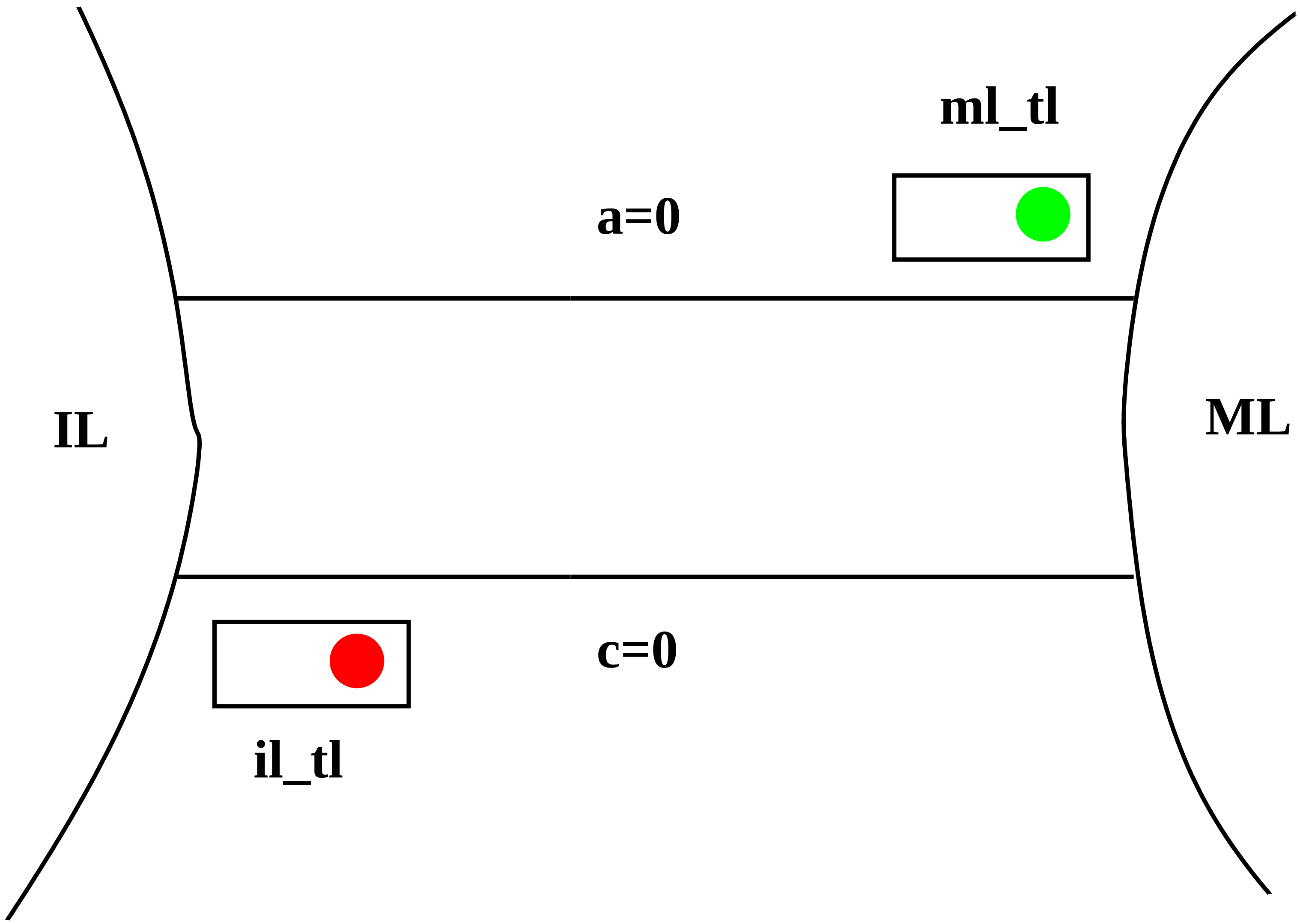
- Absence of deadlock: ?

ML_tl_green **when** $ml\_tl = \textbf{red}$ $a + b < d$ $c = 0$ **then** $ml\_tl := \textbf{green}$ $il\_tl := \textbf{red}$ **end**	IL_tl_green **when** $il\_tl = \textbf{red}$ $0 < b$ $a = 0$ **then** $il\_tl := \textbf{green}$ $ml\_tl := \textbf{red}$ **end**

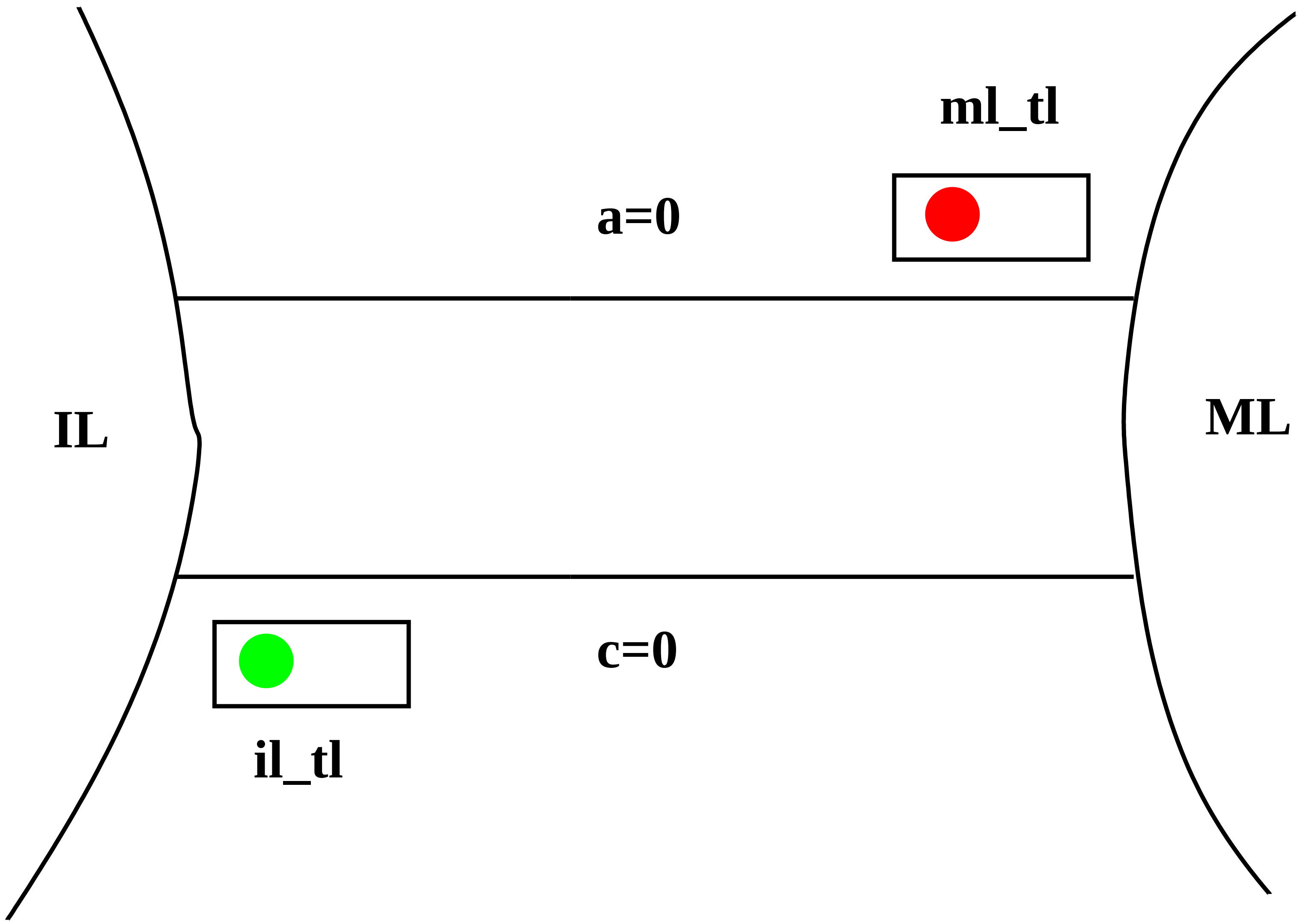
When a and c are both equal to 0 and b is positive, then both events are always alternatively enabled

The lights can change colors very rapidly

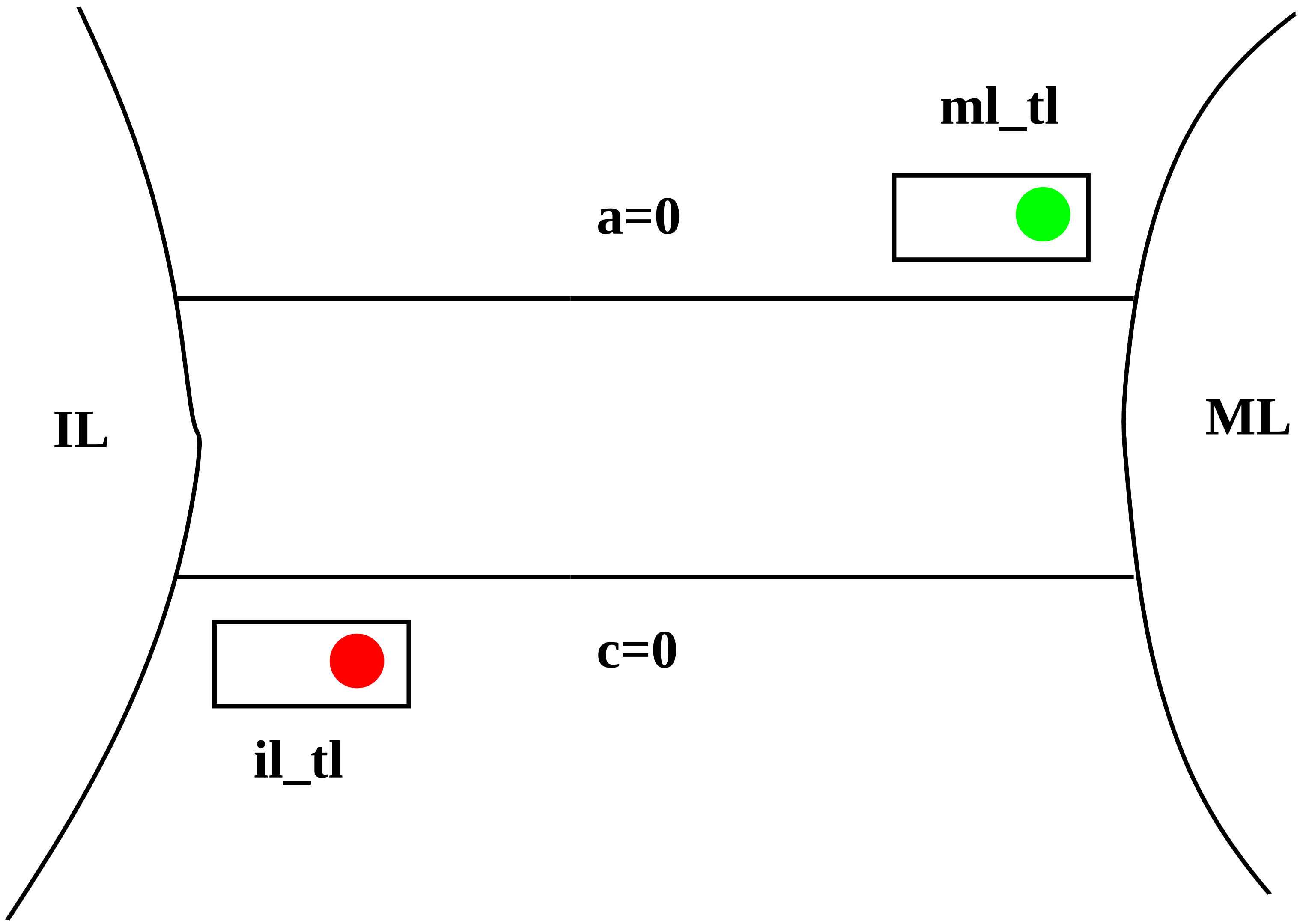
ML_tl_green and IL_tl_green can run for ever

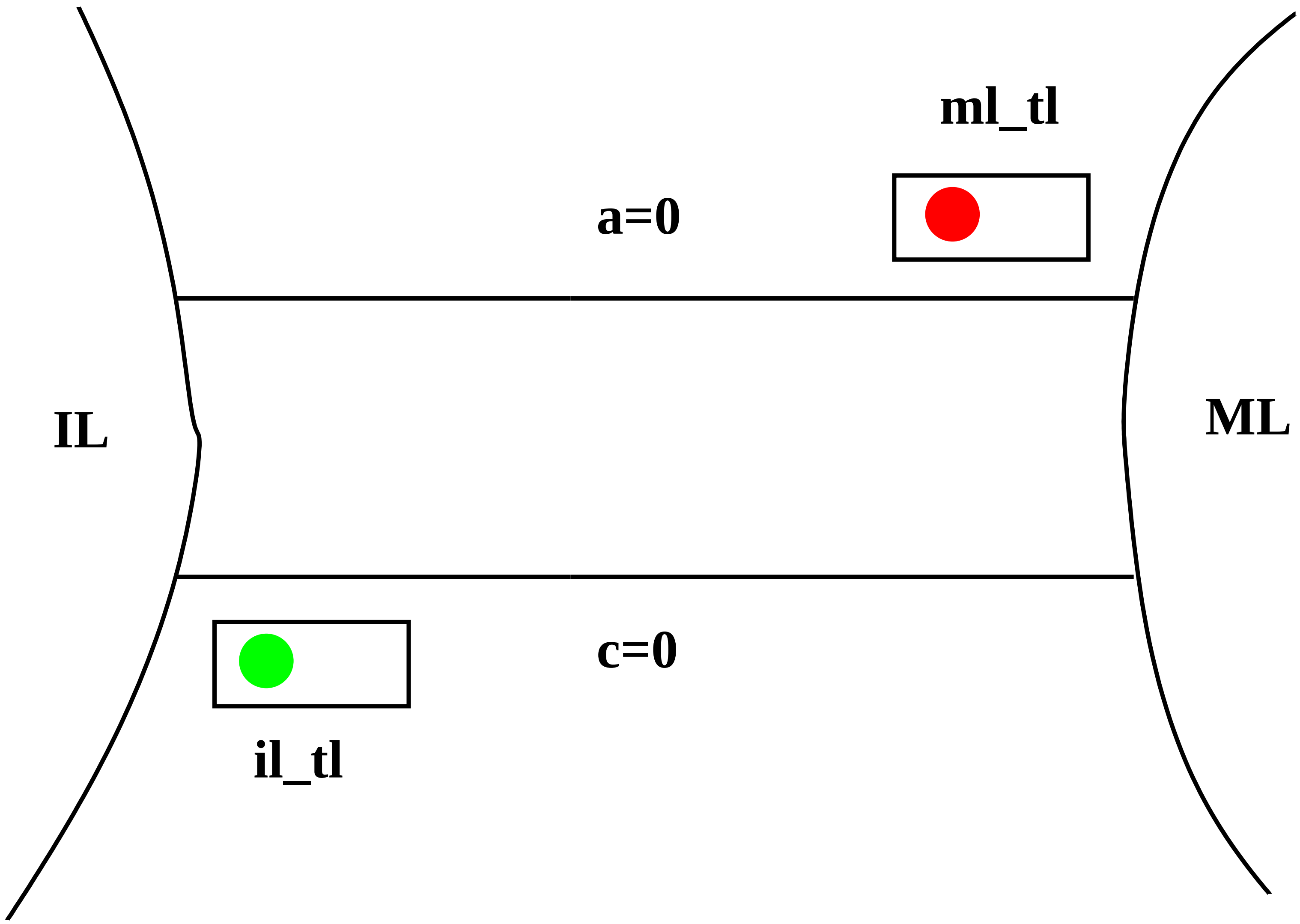


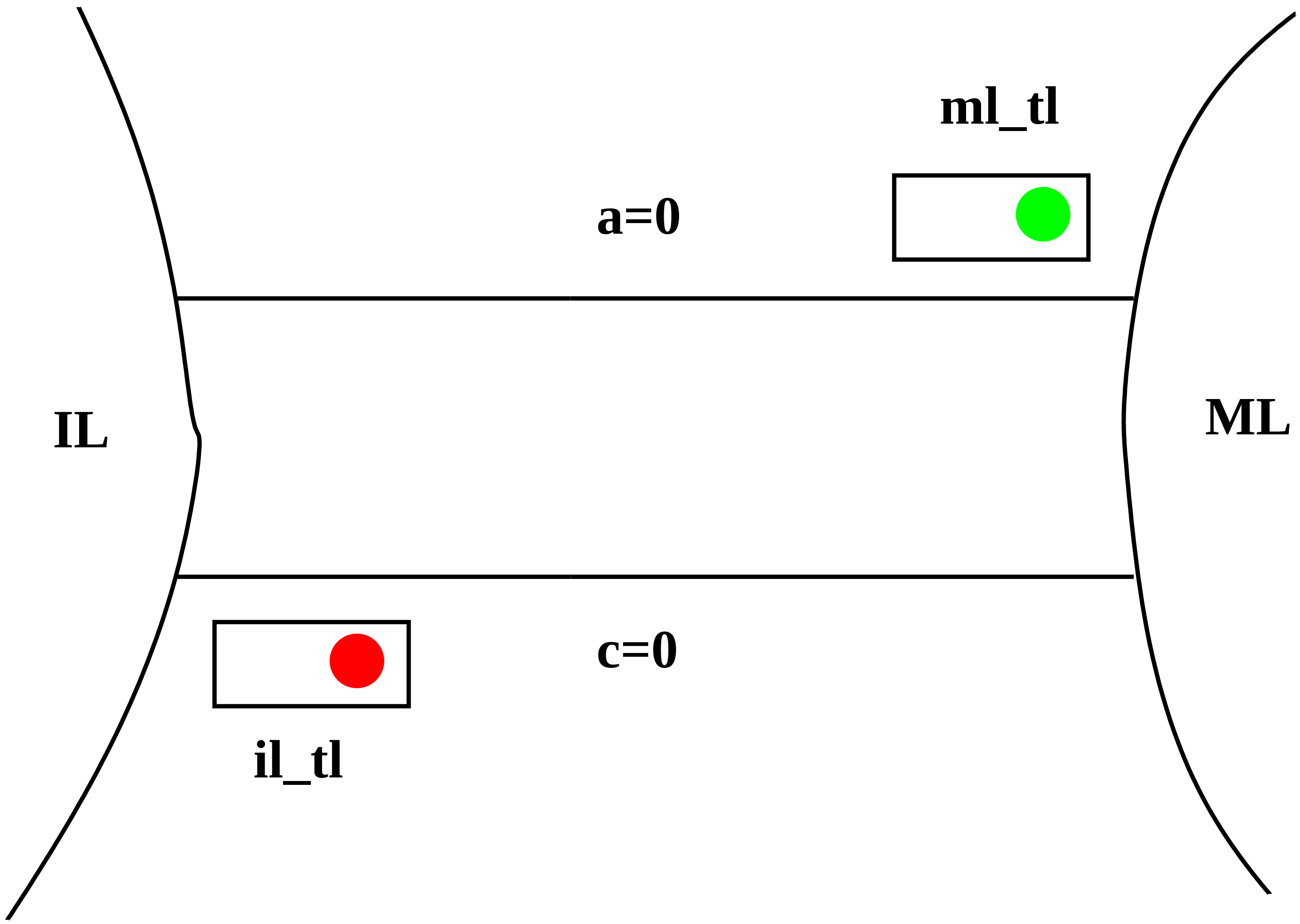
ML_tl_green and IL_tl_green can run for ever



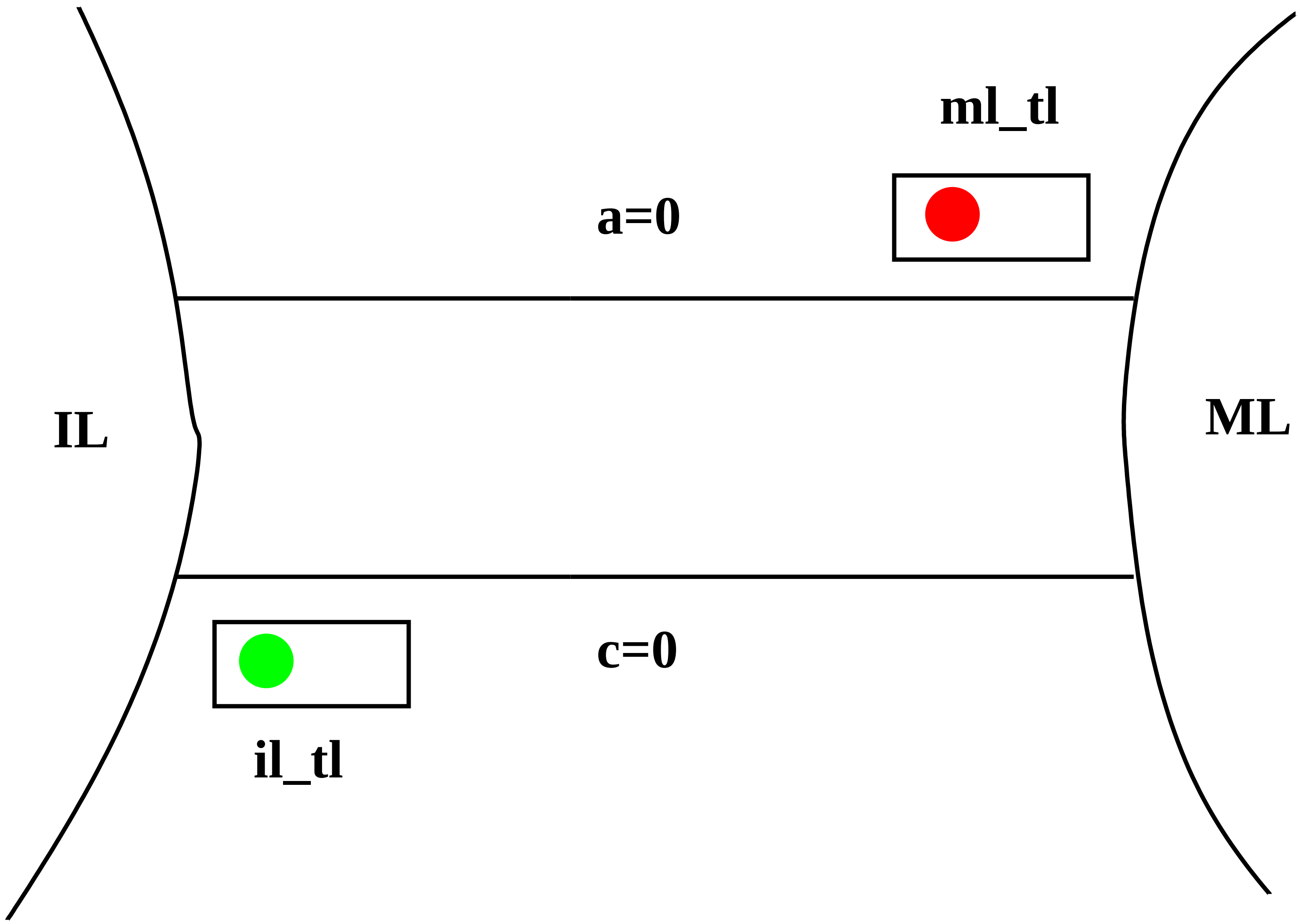
ML_tl_green and IL_tl_green can run for ever




ml_tl
a=0
IL
ML
c=0
il_tl


ml_tl
a=0
IL
ML
c=0
il_tl

ML_tl_green and IL_tl_green can run for ever



- Allowing each light to turn green only when at least one car has passed in the other direction

- For this, we introduce two additional variables:

inv2_6: $ml\_pass \in \{0, 1\}$

inv2_7: $il\_pass \in \{0, 1\}$

ML_out_1
when
$ml\_tl = \textbf{green}$
$a + 1 + b < d$
then
$a := a + 1$
$ml\_pass := 1$
end

ML_out_2
when
$ml\_tl = \textbf{green}$
$a + 1 + b = d$
then
$a := a + 1$
$ml\_tl := \textbf{red}$
$ml\_pass := 1$
end

IL_out_1
when
$il\_tl = \textbf{green}$
$b \neq 1$
then
$b := b - 1$
$c := c + 1$
$il\_pass := 1$
end

IL_out_2
when
$il\_tl = \textbf{green}$
$b = 1$
then
$b := b - 1$
$c := c + 1$
$il\_tl := \textbf{red}$
$il\_pass := 1$
end

ML_tl_green
when
$ml\_tl = \textbf{red}$
$a + b < d$
$c = 0$
$il\_pass = 1$
then
$ml\_tl := \textbf{green}$
$il\_tl := \textbf{red}$
$ml\_pass := 0$
end

IL_tl_green
when
$il\_tl = \textbf{red}$
$0 < b$
$a = 0$
$ml\_pass = 1$
then
$il\_tl := \textbf{green}$
$ml\_tl := \textbf{red}$
$il\_pass := 0$
end

We exhibit the following variant

variant_2: $ml\_pass + il\_pass$

$$
\begin{array}{l}
\quad ml\_tl = \textbf{red} \\
\quad a + b < d \\
\quad c = 0 \\
\quad il\_pass = 1 \\
\Rightarrow \\
\quad il\_pass + 0 < \\
\quad ml\_pass + il\_pass
\end{array}
\qquad
\begin{array}{l}
\quad il\_tl = \textbf{red} \\
\quad b > 0 \\
\quad a = 0 \\
\quad ml\_pass = 1 \\
\Rightarrow \\
\quad ml\_pass + 0 < \\
\quad ml\_pass + il\_pass
\end{array}
$$

This cannot be proved. This suggests the following invariants:

inv2_8: $ml\_tl = \textbf{red} \;\Rightarrow\; ml\_pass = 1$

inv2_9: $il\_tl = \textbf{red} \;\Rightarrow\; il\_pass = 1$

No Deadlock (1)

$$
\begin{array}{l}
\quad 0 < d \\
\quad ml\_tl \ \in \{\textsf{red}, \textsf{green}\} \\
\quad il\_tl \ \in \{\textsf{red}, \textsf{green}\} \\
\quad ml\_pass \ \in \{0, 1\} \\
\quad il\_pass \ \in \{0, 1\} \\
\quad a \ \in \ \mathbb{N} \\
\quad b \ \in \ \mathbb{N} \\
\quad c \ \in \ \mathbb{N} \\
\quad ml\_tl = \textsf{red} \ \Rightarrow \ ml\_pass = 1 \\
\quad il\_tl = \textsf{red} \ \Rightarrow \ il\_pass = 1 \\
\Rightarrow \\
\quad (ml\_tl = \textsf{red} \ \land \ a + b < d \ \land \ c = 0 \ \land \ il\_pass = 1) \ \lor \\
\quad (il\_tl = \textsf{red} \ \land \ a = 0 \ \land \ b > 0 \ \land \ ml\_pass = 1) \ \lor \\
\quad ml\_tl = \textsf{green} \ \lor \ il\_tl = \textsf{green} \ \lor \ a > 0 \ \lor \ c > 0
\end{array}
$$

The previous statement reduces to the following, which is true

$$\begin{array}{l} \quad 0 < d \\ \quad a \in \mathbb{N} \\ \quad b \in \mathbb{N} \\ \quad c \in \mathbb{N} \\ \Rightarrow \\ \quad (a + b < d \ \wedge \ c = 0) \ \vee \\ \quad (a = 0 \ \wedge \ b > 0) \ \vee \\ \quad a > 0 \ \vee \\ \quad c > 0 \end{array} \quad \rightsquigarrow \quad \begin{array}{l} \quad 0 < d \\ \quad b \in \mathbb{N} \\ \Rightarrow \\ \quad b < d \ \vee \ b > 0 \end{array}$$

- Thanks to the proofs:
    - We discovered 4 errors
    - We introduced several additional invariants
    - We corrected 4 events
    - We introduced 2 more variables

Conclusion: we Introduced the Superposition Rule

Axioms Abstract invariants Concrete invariants Concrete guards $\vdash$ Same actions on common variables	SIM

Summary of Second Refinement: the State (1)

variables: $a, b, c,$
$ml\_tl, il\_tl, ml\_pass, il\_pass$

inv2_1: $ml\_tl \in \{\mathbf{red}, \mathbf{green}\}$

inv2_2: $il\_tl \in \{\mathbf{red}, \mathbf{green}\}$

inv2_3: $ml\_tl = 1 \Rightarrow a + b < d \wedge c = 0$

inv2_4: $il\_tl = 1 \Rightarrow 0 < b \wedge a = 0$

inv2_5: $ml\_tl = \textbf{red} \quad \vee \quad il\_tl = \textbf{red}$

inv2_6: $ml\_pass \in \{0, 1\}$

inv2_7: $il\_pass \in \{0, 1\}$

inv2_8: $ml\_tl = \textbf{red} \Rightarrow ml\_pass = 1$

inv2_9: $il\_tl = \textbf{red} \Rightarrow il\_pass = 1$

variant2: $ml\_pass + il\_pass$

ML_out_1
when
$ml\_tl = \textbf{green}$
$a + 1 + b < d$
then
$a := a + 1$
$ml\_pass := 1$
end

ML_out_2
when
$ml\_tl = \textbf{green}$
$a + 1 + b = d$
then
$a := a + 1$
$ml\_pass := 1$
$ml\_tl := \textbf{red}$
end

Summary of Second Refinement: the Event (2)

IL_out_1
  when
    $il\_tl = \mathbf{green}$
    $b \neq 1$
  then
    $b := b - 1$
    $c := c + 1$
    $il\_pass := 1$
  end

IL_out_2
  when
    $il\_tl = \mathbf{green}$
    $b = 1$
  then
    $b := b - 1$
    $c := c + 1$
    $il\_pass := 1$
    $il\_tl := \mathbf{red}$
  end

Summary of Second Refinement: the Event (3)

ML_tl_green **when** $ml\_tl = \textbf{red}$ $a + b < d$ $c = 0$ $il\_pass = 1$ **then** $ml\_tl := \textbf{green}$ $il\_tl := \textbf{red}$ $ml\_pass := 0$ **end**	IL_tl_green **when** $il\_tl = \textbf{red}$ $0 < b$ $a = 0$ $ml\_pass = 1$ **then** $il\_tl := \textbf{green}$ $ml\_tl := \textbf{red}$ $il\_pass := 0$ **end**

- These events are identical to their abstract versions

ML_in
  when
    $0 < c$
  then
    $c := c - 1$
  end

IL_in
  when
    $0 < a$
  then
    $a := a - 1$
    $b := b + 1$
  end

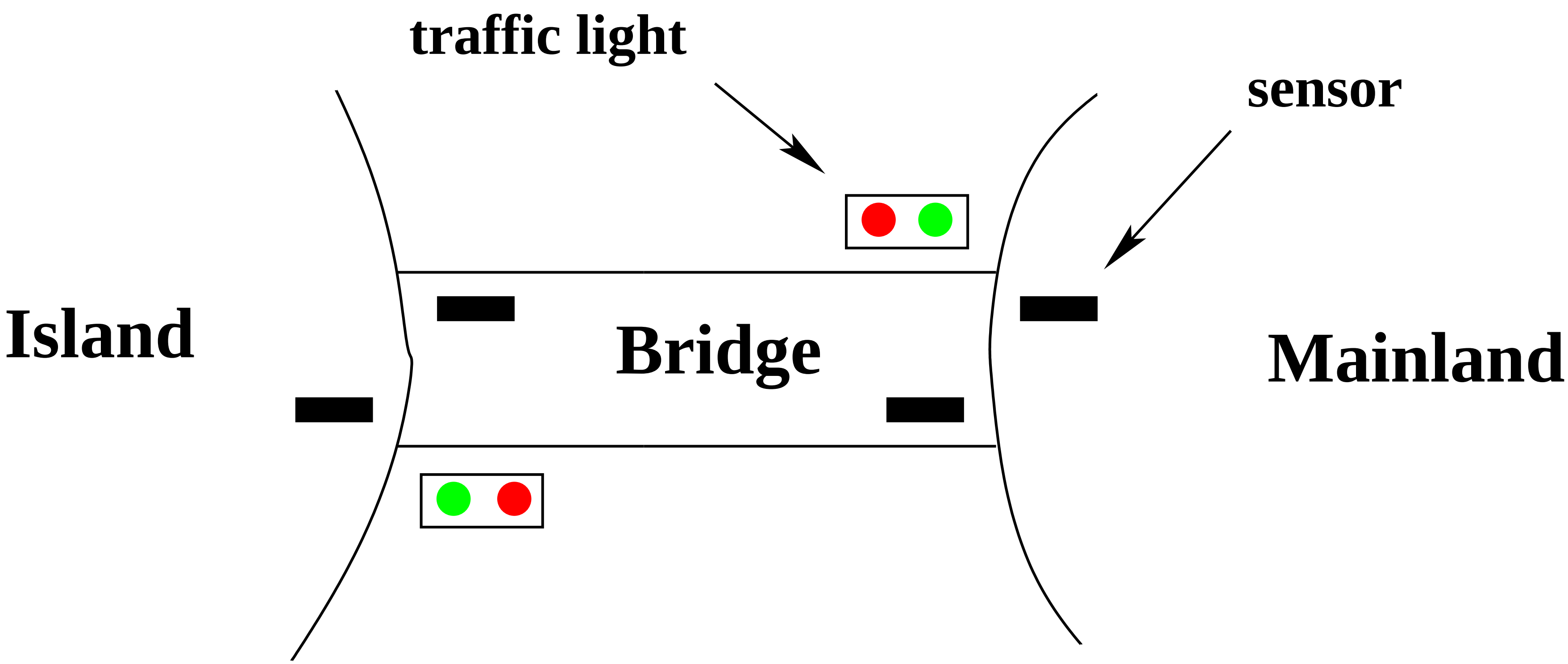
- Initial model: Limiting the number of cars (FUN_2)

- First refinement: Introducing the one way bridge (FUN_3)

- Second refinement: Introducing the traffic lights (EQP_1,2,3)

- Third refinement: Introducing the sensors (EQP_4,5)

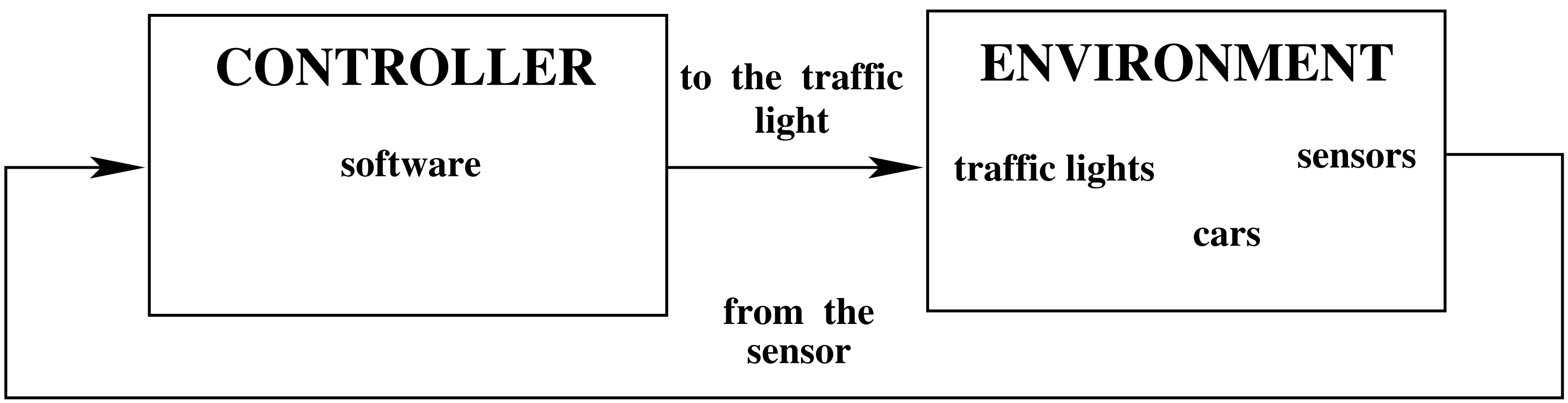
Reminder of the physical system



Closed Model

-We want to clearly identify in our model:

- The controller
- The environment
- The communication channels between the two



Contoller variables: $\boldsymbol{a},$

$\boldsymbol{b},$

$\boldsymbol{c},$

$\boldsymbol{ml\_pass},$

$\boldsymbol{il\_pass}$

These new variables denote physical objects

Environment variables: $\boldsymbol{A}$,

$\boldsymbol{B}$,

$\boldsymbol{C}$,

$\boldsymbol{ML\_OUT\_SR}$,

$\boldsymbol{ML\_IN\_SR}$,

$\boldsymbol{IL\_OUT\_SR}$,

$\boldsymbol{IL\_IN\_SR}$

Output channels: ml_tl,

il_tl

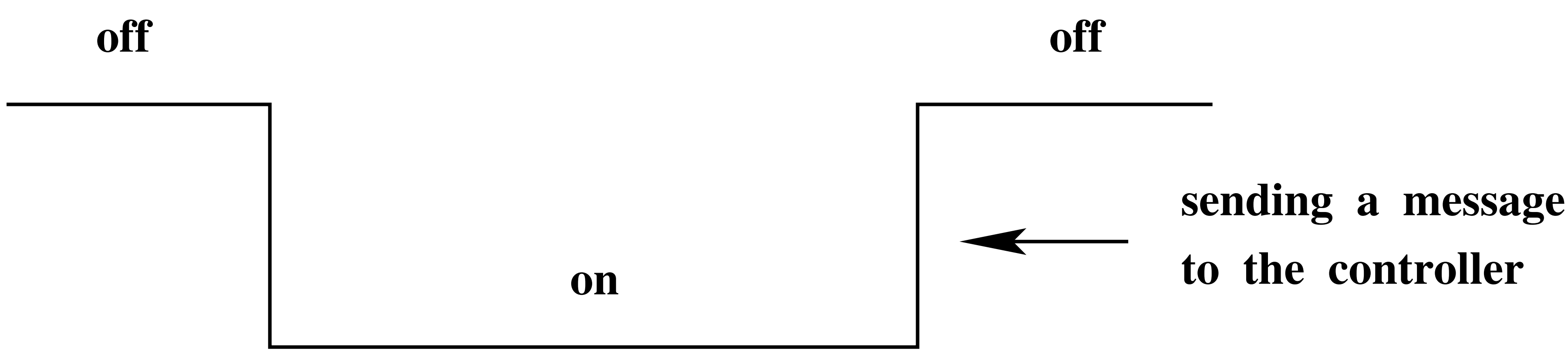
Input channels: $ml\_out\_10,$

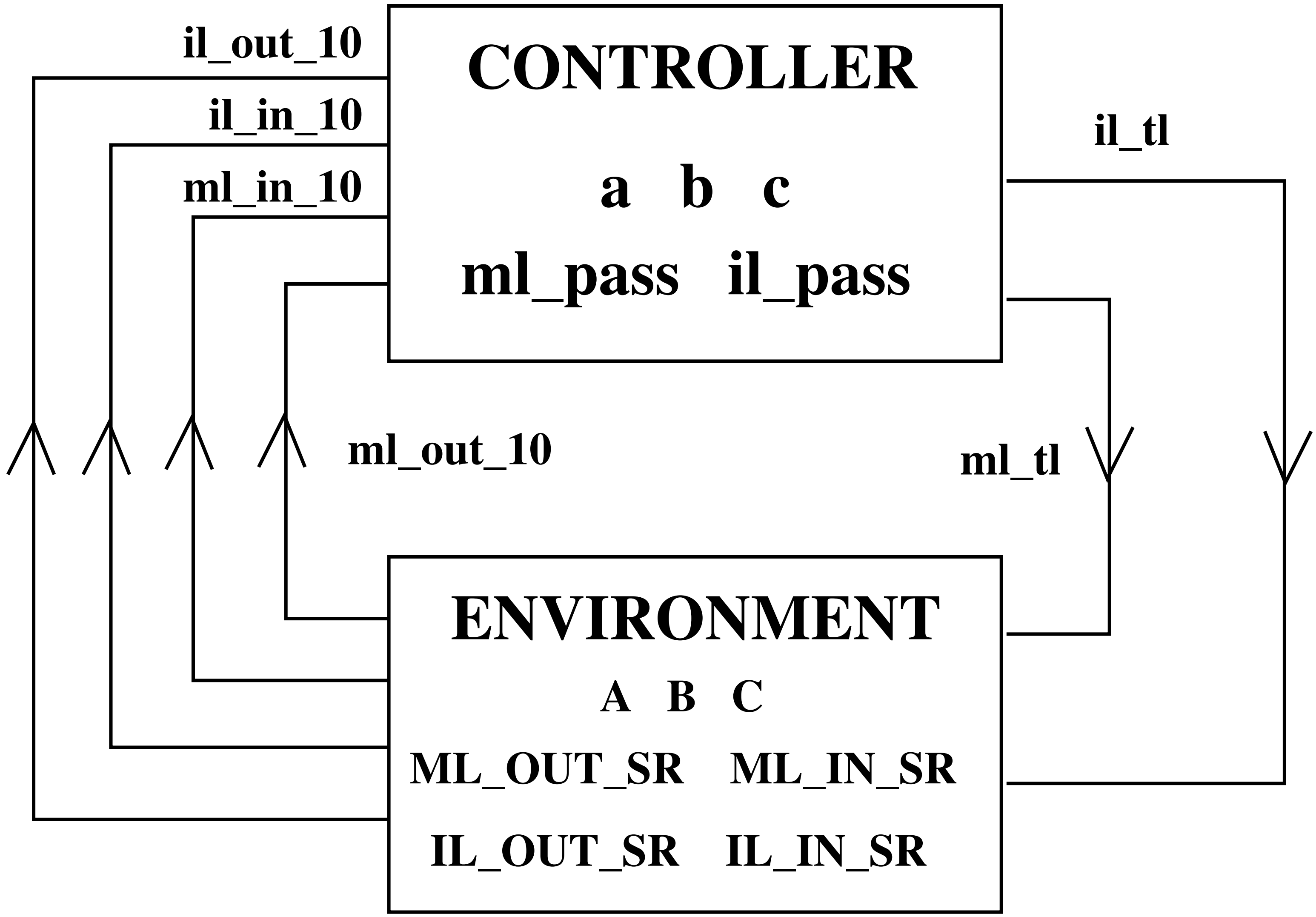
$ml\_in\_10,$

$il\_in\_10,$

il_out_10

A message is sent when a sensor moves from "on" to "off":




il_out_10
il_in_10
ml_in_10
CONTROLLER
a b c
ml_pass il_pass
il_tl
ml_out_10
ml_tl
ENVIRONMENT
A B C
ML_OUT_SR ML_IN_SR
IL_OUT_SR IL_IN_SR

carrier sets: $\ldots, SENSOR$

constants: $\ldots, on, off$

axm3_1: $SENSOR = \{on, off\}$

axm3_2: $on \neq off$

Variables (1)

$$\mathbf{inv3\_1:} \quad ML\_OUT\_SR \in SENSOR$$

$$\mathbf{inv3\_2:} \quad ML\_IN\_SR \in SENSOR$$

$$\mathbf{inv3\_3:} \quad IL\_OUT\_SR \in SENSOR$$

$$\mathbf{inv3\_4:} \quad IL\_IN\_SR \in SENSOR$$

$$\begin{array}{lrcl}
\textbf{inv3_5}: & \boldsymbol{A} & \in & \mathbb{N} \\
\textbf{inv3_6}: & \boldsymbol{B} & \in & \mathbb{N} \\
\textbf{inv3_7}: & \boldsymbol{C} & \in & \mathbb{N} \\
\textbf{inv3_8}: & \boldsymbol{ml_out_10} & \in & \mathbf{BOOL} \\
\textbf{inv3_9}: & \boldsymbol{ml_in_10} & \in & \mathbf{BOOL} \\
\textbf{inv3_10}: & \boldsymbol{il_out_10} & \in & \mathbf{BOOL} \\
\textbf{inv3_11}: & \boldsymbol{il_in_10} & \in & \mathbf{BOOL}
\end{array}$$

Invariants (1)

When sensors are on, there are cars on them

$$\mathbf{inv3\_12:}\quad IL\_IN\_SR = on \;\;\Rightarrow\;\; A > 0$$

$$\mathbf{inv3\_13:}\quad IL\_OUT\_SR = on \;\;\Rightarrow\;\; B > 0$$

$$\mathbf{inv3\_14:}\quad ML\_IN\_SR = on \;\;\Rightarrow\;\; C > 0$$

The sensors are used to detect the presence of cars entering or leaving the bridge	EQP-5

Invariants (2)

Drivers obey the traffic lights

$$\mathbf{inv3\_15:}\quad \boldsymbol{ml\_out\_10} = \mathbf{TRUE} \;\Rightarrow\; \boldsymbol{ml\_tl = green}$$

$$\mathbf{inv3\_16:}\quad \boldsymbol{il\_out\_10} = \mathbf{TRUE} \;\Rightarrow\; \boldsymbol{il\_tl = green}$$

Cars are not supposed to pass on a red traffic light, only on a green one	EQP-3

When a sensor is "on", the previous information is treated

$$\begin{array}{ll} \mathbf{inv3\_17:} & \boldsymbol{IL\_IN\_SR = on \;\Rightarrow\; il\_in\_10 = \mathrm{FALSE}} \\ \mathbf{inv3\_18:} & \boldsymbol{IL\_OUT\_SR = on \;\Rightarrow\; il\_out\_10 = \mathrm{FALSE}} \\ \mathbf{inv3\_19:} & \boldsymbol{ML\_IN\_SR = on \;\Rightarrow\; ml\_in\_10 = \mathrm{FALSE}} \\ \mathbf{inv3\_20:} & \boldsymbol{ML\_OUT\_SR = on \;\Rightarrow\; ml\_out\_10 = \mathrm{FALSE}} \end{array}$$

The controller must be fast enough so as to be able to treat all the information coming from the environment	FUN-5

Linking the physical and logical cars (1)

$$\mathbf{inv3\_21:}\quad il\_in\_10 = \mathrm{TRUE} \wedge ml\_out\_10 = \mathrm{TRUE} \;\Rightarrow\; A = a$$

$$\mathbf{inv3\_22:}\quad il\_in\_10 = \mathrm{FALSE} \wedge ml\_out\_10 = \mathrm{TRUE} \;\Rightarrow\; A = a + 1$$

$$\mathbf{inv3\_23:}\quad il\_in\_10 = \mathrm{TRUE} \wedge ml\_out\_10 = \mathrm{FALSE} \;\Rightarrow\; A = a - 1$$

$$\mathbf{inv3\_24:}\quad il\_in\_10 = \mathrm{FALSE} \wedge ml\_out\_10 = \mathrm{FALSE} \;\Rightarrow\; A = a$$

Linking the physical and logical cars (2)

$$\begin{array}{ll}
\mathbf{inv3_25:} & il_in_10 = \mathrm{TRUE} \;\wedge\; il_out_10 = \mathrm{TRUE} \;\;\Rightarrow\;\; B = b \\
\mathbf{inv3_26:} & il_in_10 = \mathrm{TRUE} \;\wedge\; il_out_10 = \mathrm{FALSE} \;\;\Rightarrow\;\; B = b + 1 \\
\mathbf{inv3_27:} & il_in_10 = \mathrm{FALSE} \;\wedge\; il_out_10 = \mathrm{TRUE} \;\;\Rightarrow\;\; B = b - 1 \\
\mathbf{inv3_28:} & il_in_10 = \mathrm{FALSE} \;\wedge\; il_out_10 = \mathrm{FALSE} \;\;\Rightarrow\;\; B = b
\end{array}$$

$$\begin{array}{ll}
\mathbf{inv3_29:} & il_out_10 = \mathrm{TRUE} \;\wedge\; ml_out_10 = \mathrm{TRUE} \;\;\Rightarrow\;\; C = c \\
\mathbf{inv3_30:} & il_out_10 = \mathrm{TRUE} \;\wedge\; ml_out_10 = \mathrm{FALSE} \;\;\Rightarrow\;\; C = c + 1 \\
\mathbf{inv3_31:} & il_out_10 = \mathrm{FALSE} \;\wedge\; ml_out_10 = \mathrm{TRUE} \;\;\Rightarrow\;\; C = c - 1 \\
\mathbf{inv3_32:} & il_out_10 = \mathrm{FALSE} \;\wedge\; ml_out_10 = \mathrm{FALSE} \;\;\Rightarrow\;\; C = c
\end{array}$$

The basic properties hold for the physical cars

$$\mathbf{inv3\_33:} \quad A = 0 \quad \vee \quad C = 0$$

$$\mathbf{inv3\_34:} \quad A + B + C \leq d$$

The number of cars on the bridge and the island is limited	FUN-2

The bridge is one way or the other, not both at the same time	FUN-3

Refining Abstract Events (1)

ML_out_1
when
$\underline{ml\_out\_10 = \mathrm{TRUE}}$
$a + b + 1 \neq d$
then
$a := a + 1$
$ml\_pass := 1$
$\underline{ml\_out\_10 := \mathrm{FALSE}}$
end

ML_out_2
when
$\underline{ml\_out\_10 = \mathrm{TRUE}}$
$a + b + 1 = d$
then
$a := a + 1$
$ml\_tl := red$
$ml\_pass := 1$
$\underline{ml\_out\_10 := \mathrm{FALSE}}$
end

(abstract-)ML_out_1
when
$ml\_tl = \textbf{green}$
$a + b + 1 \neq d$
then
$a := a + 1$
$ml\_pass := 1$
end

(abstract-)ML_out_2
when
$ml\_tl = \textbf{green}$
$a + b + 1 = d$
then
$a := a + 1$
$ml\_pass := 1$
$ml\_tl := \textbf{red}$
end

Refining Abstract Events (2)

IL_out_1
when
$\underline{il\_out\_10 = \mathrm{TRUE}}$
$b \neq 1$
then
$b := b - 1$
$c := c + 1$
$il\_pass := 1$
$\underline{il\_out\_10 := \mathrm{FALSE}}$
end

IL_out_2
when
$\underline{il\_out\_10 = \mathrm{TRUE}}$
$b = 1$
then
$b := b - 1$
$c := c + 1$
$il\_tl := red$
$il\_pass := 1$
$\underline{il\_out\_10 := \mathrm{FALSE}}$
end

(abstract-)IL_out_1
when
$il\_tl =$ green
$b \neq 1$
then
$b := b - 1$
$c := c + 1$
$il\_pass := 1$
end

(abstract-)IL_out_2
when
$il\_tl =$ green
$b = 1$
then
$b := b - 1$
$c := c + 1$
$il\_pass := 1$
$il\_tl :=$ red
end

Refining Abstract Events (3)

ML_in
when
$\underline{ml\_in\_10 = \mathrm{TRUE}}$
$0 < c$
then
$c := c - 1$
$\underline{ml\_in\_10 := \mathrm{FALSE}}$
end

IL_in
when
$\underline{il\_in\_10 = \mathrm{TRUE}}$
$0 < a$
then
$a := a - 1$
$b := b + 1$
$\underline{il\_in\_10 := \mathrm{FALSE}}$
end

(abstract-)ML_in
when
$0 < c$
then
$c := c - 1$
end

(abstract-)IL_in
when
$0 < a$
then
$a := a - 1$
$b := b + 1$
end

Refining Abstract Events (4)

ML_tl_green
when
$ml\_tl = red$
$a + b < d$
$c = 0$
$il\_pass = 1$
$\underline{il\_out\_10 = \text{FALSE}}$
then
$ml\_tl := green$
$il\_tl := red$
$ml\_pass := \text{FALSE}$
end

IL_tl_green
when
$il\_tl = red$
$a = 0$
$ml\_pass = 1$
$\underline{ml\_out\_10 = \text{FALSE}}$
then
$il\_tl := green$
$ml\_tl := red$
$il\_pass := \text{FALSE}$
end

(abstract-)ML_tl_green
when
$ml\_tl =$ **red**
$a + b < d$
$c = 0$
$il\_pass = 1$
then
$ml\_tl :=$ **green**
$il\_tl :=$ **red**
$ml\_pass := 0$
end

(abstract-)IL_tl_green
when
$il\_tl =$ **red**
$0 < b$
$a = 0$
$ml\_pass = 1$
then
$il\_tl :=$ **green**
$ml\_tl :=$ **red**
$il\_pass := 0$
end

Adding New PHYSICAL Events (1)

ML_out_arr
when
$ML\_OUT\_SR = off$
$ml\_out\_10 = \mathrm{FALSE}$
then
$ML\_OUT\_SR := on$
end

ML_in_arr
when
$ML\_IN\_SR = off$
$ml\_in\_10 = \mathrm{FALSE}$
$C > 0$
then
$ML\_IN\_SR := on$
end

IL_in_arr
when
$IL\_IN\_SR = off$
$il\_in\_10 = \mathrm{FALSE}$
$A > 0$
then
$IL\_IN\_SR := on$
end

IL_out_arr
when
$IL\_OUT\_SR = off$
$il\_out\_10 = \mathrm{FALSE}$
$B > 0$
then
$IL\_OUT\_SR := on$
end

Adding New PHYSICAL Events (2)

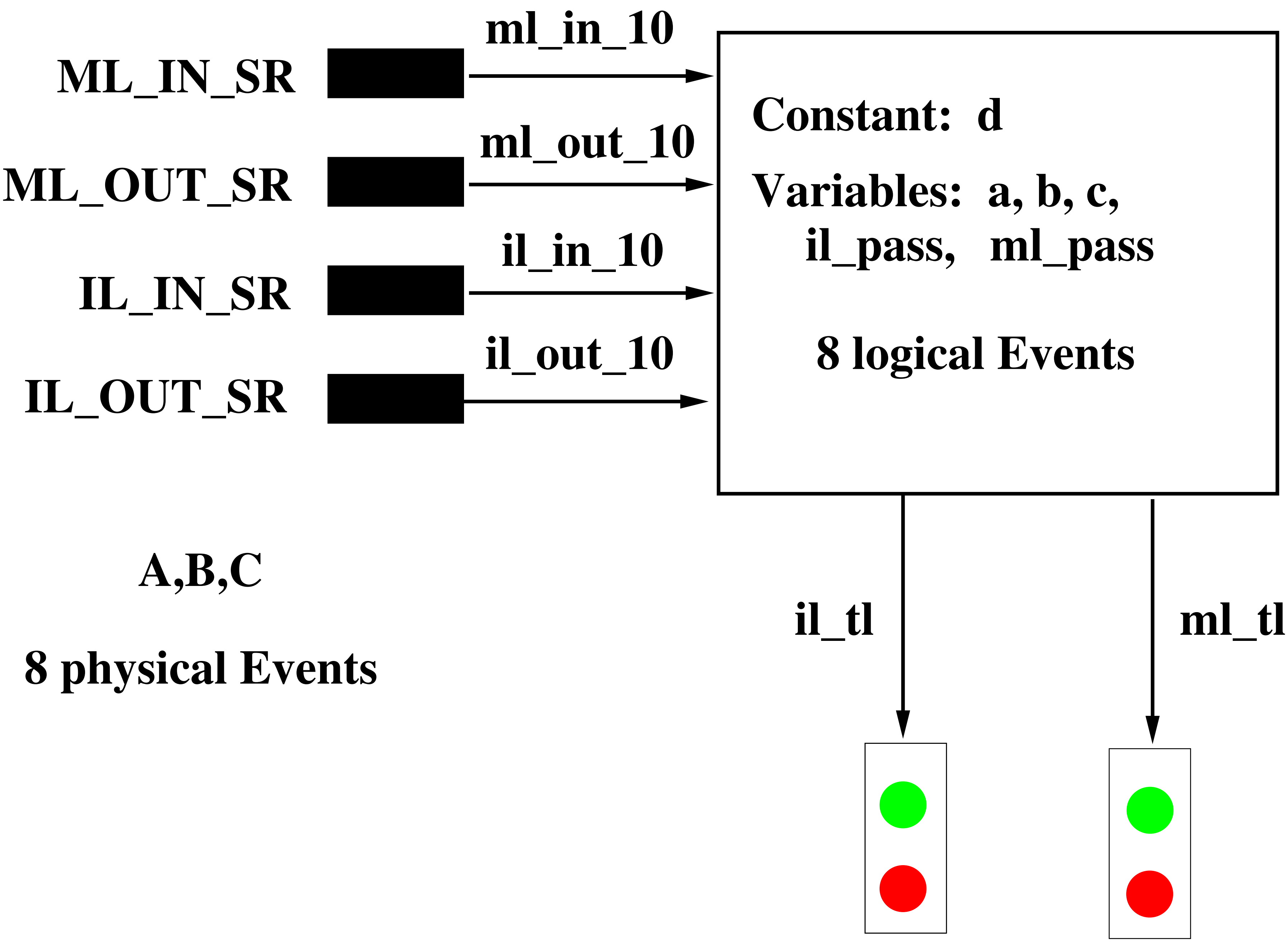
ML_out_dep
when
$ML\_OUT\_SR = on$
$ml\_tl = green$
then
$ML\_OUT\_SR := off$
$ml\_out\_10 := \mathrm{TRUE}$
end

ML_in_dep
when
$ML\_IN\_SR = on$
then
$ML\_IN\_SR := off$
$ml\_in\_10 := \mathrm{TRUE}$
$C = C - 1$
end

IL_in_dep
when
$IL\_IN\_SR = on$
then
$IL\_IN\_SR := off$
$il\_in\_10 := \mathrm{TRUE}$
$A = A - 1$
$B = B + 1$
end

IL_out_dep
when
$IL\_OUT\_SR = on$
$il\_tl = green$
then
$IL\_OUT\_SR := off$
$il\_out\_10 := \mathrm{TRUE}$
$B = B - 1$
$C = C + 1$
end

Final Structure of the Controller



- What is to be systematically proved?
  - Invariant preservation
  - Correct refinements of transitions
  - No divergence of new transitions
  - No deadlock introduced in refinements

- When are these proofs done?

Questions on Proving (cont'd)

- Who states what is to be proved?
  - An automatic tool: the Proof Obligation Generator

- Who is going to perform these proofs?
  - An automatic tool: the Prover
  - Sometimes helped by the Engineer (interactive proving)

- Three basic tools:
    - Proof Obligation Generator
    - Prover
    - Model translators into Hardware or Software languages
- These tools are embedded into a Development Data Base
- Such tools already exist in the Rodin Platform

Summary of Proofs on Example

- This development required 237 proofs
    - Initial model: 7
    - 1st refinement: 26
    - 2nd refinement: 66
    - 3rd refinement: 138

- All proved automatically by the Rodin Platform

Summary of Mathematical Notations (1)

$P \wedge Q$	conjunction
$P \vee Q$	disjunction
$P \Rightarrow Q$	implication
$\neg P$	negation
$x \in S$	set membership operator

$\mathbb{N}$	set of Natural Numbers: $\{0, 1, 2, 3, \ldots\}$
$\mathbb{Z}$	set of Integers: $\{0, 1, -1, 2, -2, \ldots\}$
$\{a, b, \ldots\}$	set defined in extension
$a + b$	addition of a and b
$a - b$	subtraction of a and b

$a * b$	product of a and b
$a = b$	equality relation
$a \leq b$	smaller than or equal relation
$a < b$	smaller than relation

Invariant Establishment Proof Rule

- For the init event in the initial model

Axioms of the constants $\Rightarrow$ Modified Invariants	INV

- For other events in the initial model

Axioms of the constants Invariants Guard of the event $\Rightarrow$ Modified Invariants	INV

- This rule is not mandatory

Axiom of the constant Invariants $\Rightarrow$ Disjunction of the guards	DLF

Refinement Rules (1): Guard Strengthening

- For old events only

Axioms of the constants Abstract invariants Concrete invariants Concrete guards $\Rightarrow$ Abstract guards	GRD

Refinement Rules (2): Invariant Establishment

- For init event only

Axioms of the constants $\Rightarrow$ Modified concrete invariants	INV

Refinement Rules (3): Invariant Preservation

- For all events (except init)

- New events refine an implicit non-guarded event with skip action

Axioms of the constants Abstract invariant Concrete invariant Concrete guard $\Rightarrow$ Modified concrete invariant	INV

- For new events only

Axioms of the constants Abstract invariants Concrete invariants Concrete guard of a new event $\Rightarrow$ Variant $\in$ $\mathbb{N}$	NAT

Refinement Rules (5): Non-divergence of New Events

- For new events only

Axioms of the constants Abstract invariants Concrete invariants Disj. of abs. guards $\Rightarrow$ Disj. of conc. guards	VAR

Refinement Rules (6): Relative Deadlock Freeness

- Global proof rule

Axioms of the constants Abstract invariants Concrete invariants Disjunction of abstract guards $\Rightarrow$ Disjunction of concrete guards	DLF

- For old events (in case of superposition)

Axioms of constants Abstract invariants Concrete invariants Concrete guards $\Rightarrow$ Same actions on common variables	SIM