

# Event-B Course

## 2. Controlling Cars on a Bridge

Jean-Raymond Abrial

September-October-November 2011

# Purpose of this Lecture (1)

---

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**
- With each refinement, the **observer “zooms in”** to see more details

- 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 criteria** are formulated as **proof obligations**
- **Proofs** will be performed by using the **sequent calculus**
- **Inference rules** used in the sequent calculus will be **reviewed**

- The concepts of **state** and **events** for defining models
- Some **principles** of system development: **invariants** and **refinement**
- A refresher of **classical logic** and **simple arithmetic foundations**
- A refresher of **formal proofs**

1. Presentation of the **requirement document**
2. Defining the **refinement strategy**
3. Development of the **initial model** and the **refinements**

**Remark:** **Theoretical background** provided during development

# 1. The Requirements Document

- We show the **embedding** of the Explanations and the References
- **Explanation:**
  - 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**.

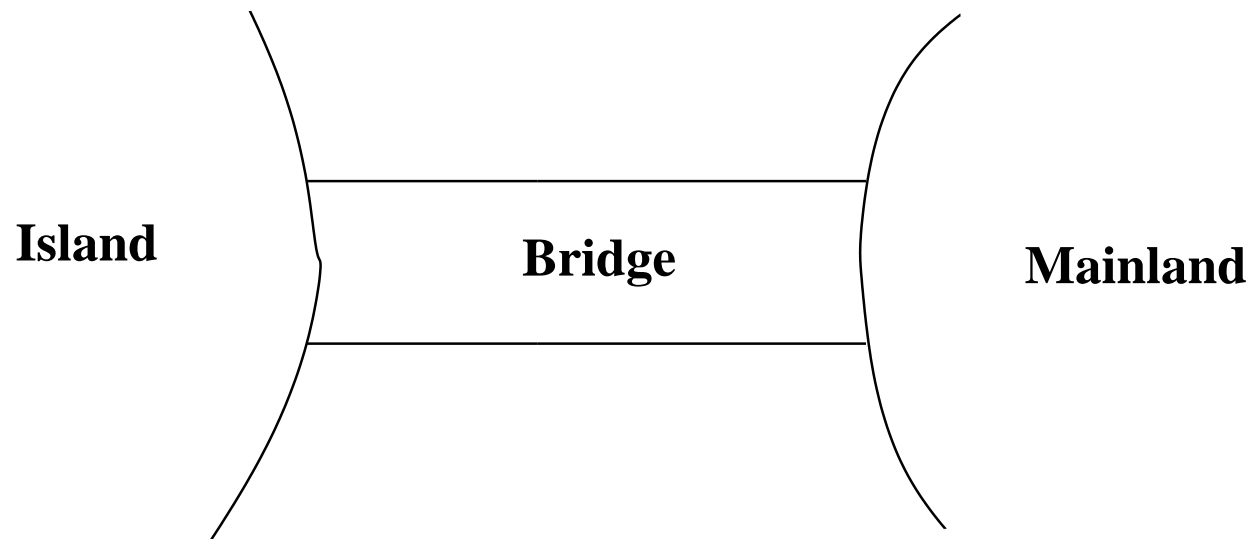
- There are **two kinds** of requirements:
  - the **equipment** (environment) labeled **EQP**,
  - the **function** of the system, labeled **FUN**.



## - Reference:

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

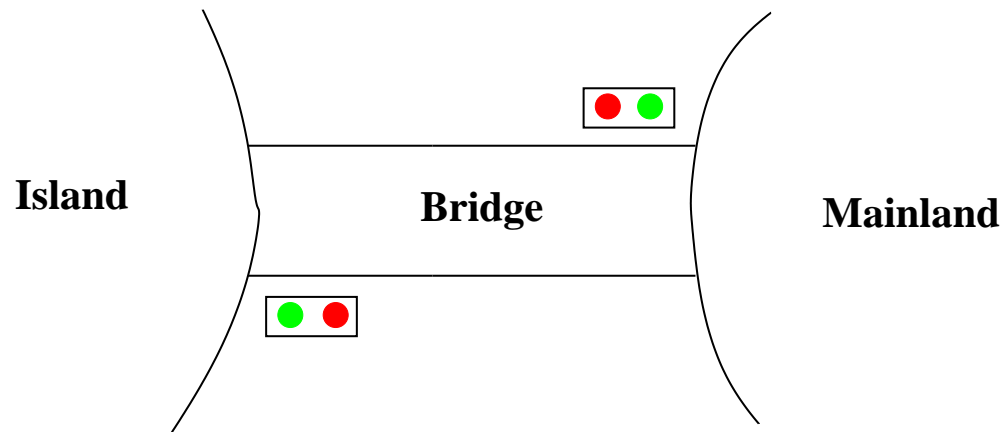
## - Explanation: This can be illustrated as follows



- **Explanation:** The controller is equipped with two **traffic lights**.
- **Reference:**

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

- **Explanation:**
- One of the traffic lights is situated on the **mainland**.
- The other one on the **island**.
- This can be illustrated as follows:



- Reference:

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

- Explanation: Drivers are supposed to obey the traffic light

- Reference:

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

- Explanation:
- There are also four car sensors
- These sensors are situated at both ends of the bridge.
- They are supposed to detect the presence of cars
- Reference:

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

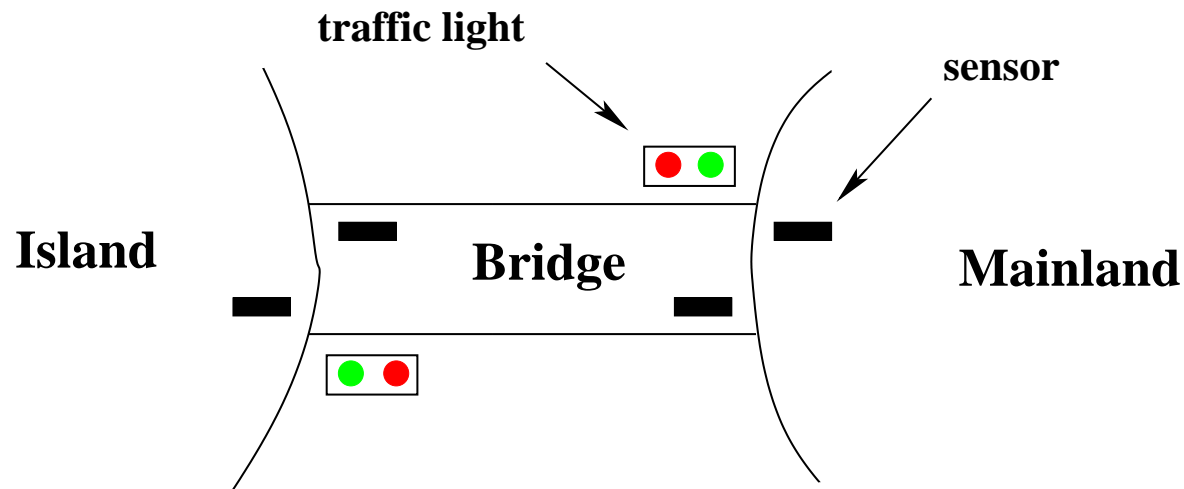
EQP-4

- **Reference:**

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

EQP-5

- **Explanation:** The pieces of equipment can be illustrated as follows:



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

## - Reference:

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



# Summary of the References (1)

---

15

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

## Summary of the References (2)

---

16

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 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

## 2. The Refinement Strategy

- Defining **a priori** the various refinement steps
- **Linking** these steps **with the requirements**
- Goal: starting the **traceability** of the requirements
- **Might be modified** during the formal development

- **Initial model:** Limiting the number of cars (FUN-2)

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

- **First refinement:** Introducing the one way bridge (FUN-3)

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

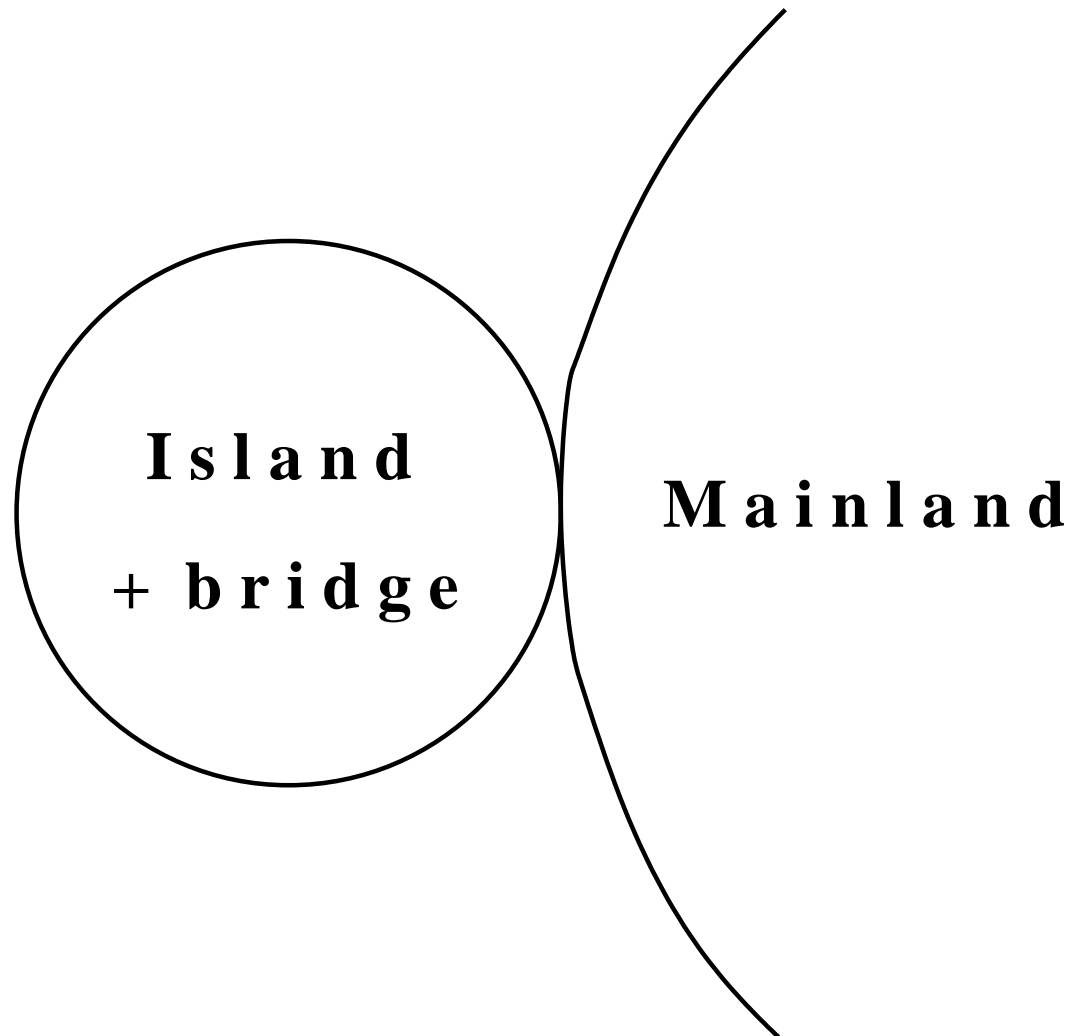
- **Second refinement:** Introducing the traffic lights (EQP-1,2,3)
- **Third refinement:** Introducing the sensors (EQP-4,5)

### 3. The Formal Development



- **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)

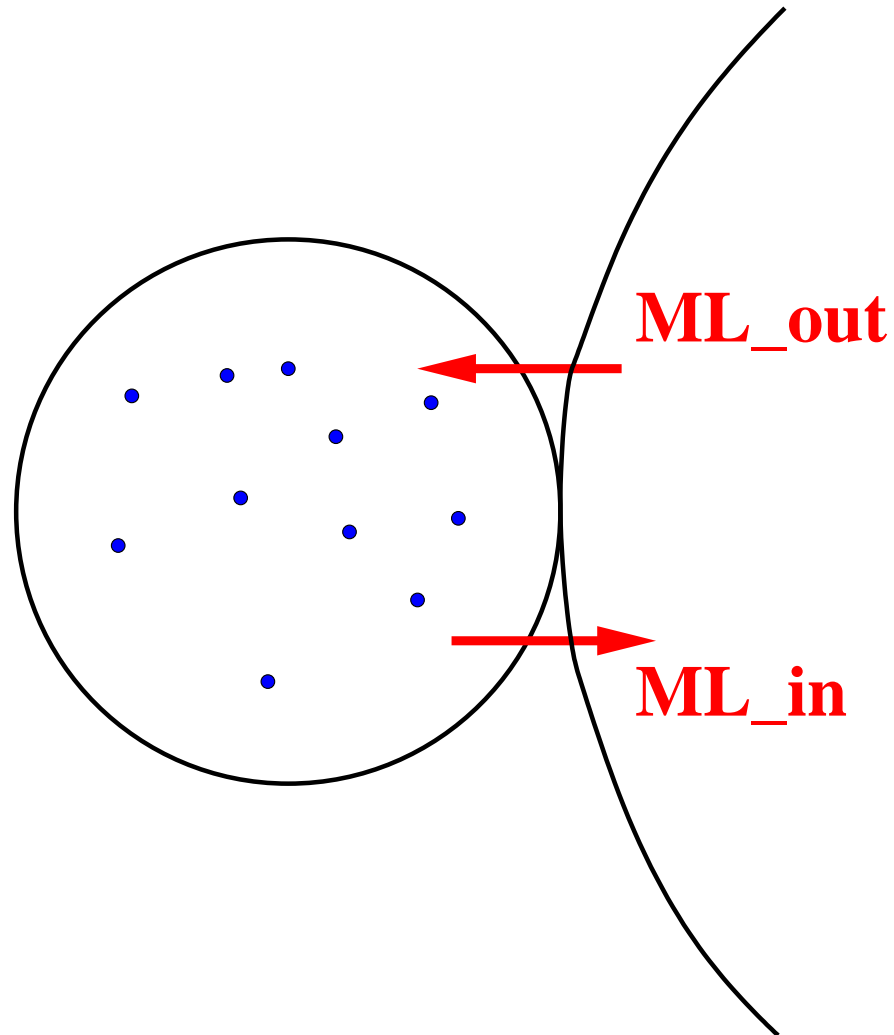
- It is **very simple**
- We completely ignore the equipment: traffic lights and sensors
- We do not even consider the bridge
- We are just interested in the **pair “island-bridge”**
- We are focusing **FUN-2**: limited number of cars on island-bridge



# Two Events that may be Observed

---

24



- **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 on the Island-Bridge
- **axm0\_1** states that  $d$  is a **natural number**
- Constant  $d$  is a member of the set  $\mathbb{N} = \{0, 1, 2, , \dots\}$

- **DYNAMIC PART**: variable  $n$  with invariants **inv0\_1** and **inv0\_2**

**variable:**  $n$

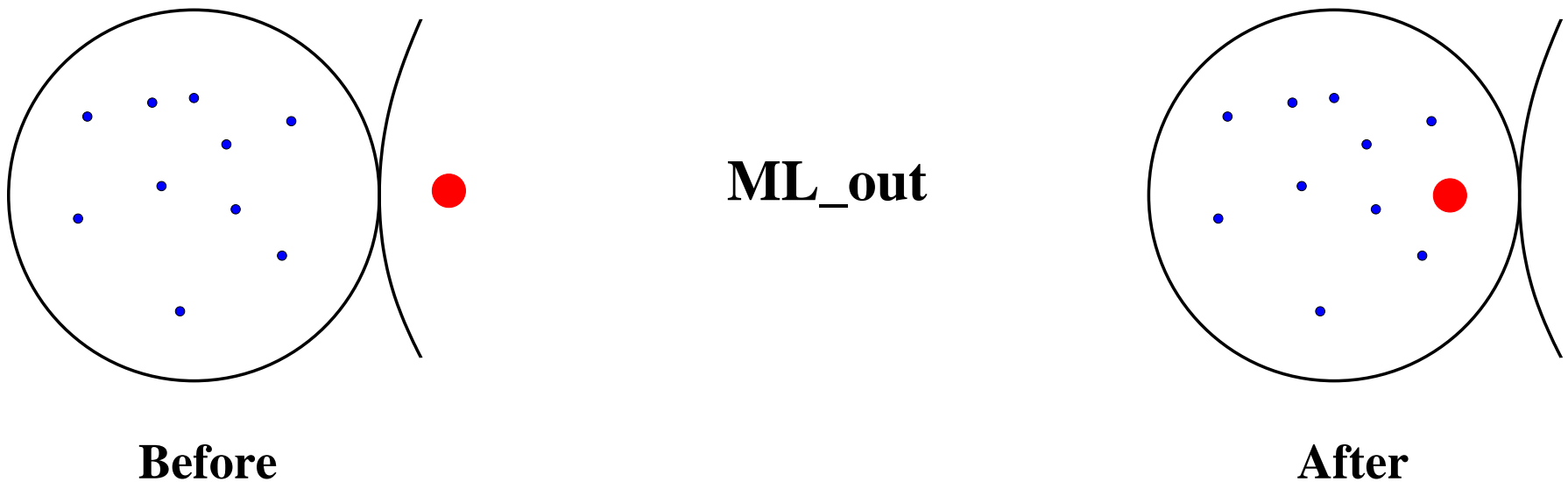
**inv0\_1:**  $n \in \mathbb{N}$

**inv0\_2:**  $n \leq d$

- $n$  is the **effective number of cars** on 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**, ... are chosen **systematically**
- The label **axm** or **inv** recalls the **purpose**:  
**axiom** of constants or **invariant** of variables
- The **0** as in **inv0\_1** stands for the initial model.
- Later we will have **inv1\_1** for an invariant of refinement **1**, etc.
- The **1** like in **inv0\_1** is a serial number
- Any convention is **valid** as long as it is **systematic**

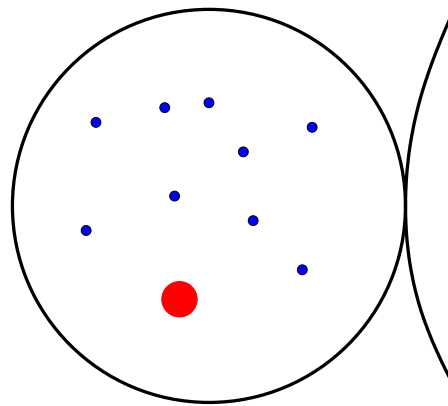
- This is the **first transition** (or event) that can be **observed**
- A car is leaving the mainland and entering the Island-Bridge



- The **number of cars** in the Island-Bridge is **incremented**

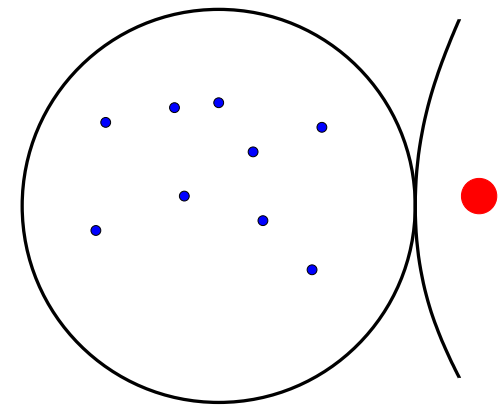


- We can also observe a **second transition** (or event)
- A car leaving the Island-Bridge and re-entering the mainland



**Before**

**ML\_in**



**After**

- The **number of cars** in the Island-Bridge is **decremented**

- 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)

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 have to perform a **proof** in order to **verify this consistency**.

- 
- An invariant is a **constraint** on the allowed values of the variables
  - An invariant **must hold on all reachable states** of a model
  - To verify that this holds we must show that
    1. the invariant holds for **initial states** (**later**), and
    2. the invariant is **preserved by all events** (**following slides**)
  - We will formalize these two statements as **proof obligations (POs)**
  - We need a **rigorous proof** showing that these POs indeed hold

- To each event can be associated a **before-after predicate**
- It describes the **relation** between the **values** of the variable(s) *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 **primed variable name**, 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.

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

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

- The after-value  $n'$  is defined as a **function** of the before-value  $n$
- This is because the corresponding events are **deterministic**
- In later lectures, we shall consider some **non-deterministic** events:

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

- Let us consider invariant **inv0\_1**

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

- And let us consider event ML\_out with before-after predicate

$$n' = n + 1$$

- **Preservation of inv0\_1** means that we have (just after ML\_out):

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



- Under **hypothesis**  $n \in \mathbb{N}$  the **conclusion**  $n + 1 \in \mathbb{N}$  holds
- This can be written as follows

$$n \in \mathbb{N} \vdash n + 1 \in \mathbb{N}$$

- This type of statement is called a **sequent** (**next slide**)
- Sequent above: **invariant preservation proof obligation for `inv0_1`**
- More **General form** of this PO will be introduced shortly

- A **sequent** is a formal statement of the following shape

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

- **H** denotes a **set of predicates**: the **hypotheses** (or **assumptions**)
- **G** denotes a predicate: the **goal** (or **conclusion**)
- The symbol "**⊢**", called the **turnstile**, stands for **provability**.  
It is read: "**Assumptions H yield conclusion G**"

- We collectively denote our set of constants by  $c$
- We denote our set of axioms by  $A(c)$ :  $A_1(c), A_2(c), \dots$
- We collectively denote our set of variables by  $v$
- We denote our set of invariants by  $I(c, v)$ :  $I_1(c, v), I_2(c, v), \dots$

- We are given an **event** with **before-after predicate**  $v' = E(c, v)$
- The following sequent expresses **preservation of invariant**  $I_i(c, v)$ :

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

- It says:  $I_i(c, E(c, v))$  provable 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
--	-----

- We assume that  $A(c)$  as well as  $I(c, v)$  hold just before the occurrence of the event represented by  $v' = E(c, v)$
- Just after the occurrence, invariant  $I_i(c, v)$  becomes  $I_i(c, v')$ , that is,  $I_i(c, E(c, v))$
- The predicate  $I_i(c, E(c, v))$  must then hold for  $I_i(c, v)$  to be an invariant

- The proof obligation

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

can be re-written vertically as follows:

$\begin{array}{l} \text{Axioms} \\ \text{Invariants} \\ \vdash \\ \text{Modified Invariant} \end{array}$	$\begin{array}{l} A(c) \\ I(c, v) \\ \vdash \\ I_i(c, E(c, v)) \end{array}$	INV
--	---	-----

- 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, we need to prove **four proof obligations**

**ML\_out**

$n := n + 1$

$(n' = n + 1)$

Axiom **axm0\_1**

Invariant **inv0\_1**

Invariant **inv0\_2**

⊢

Modified Invariant **inv0\_1**

$d \in \mathbb{N}$

$n \in \mathbb{N}$

$n \leq d$

⊢

$n + 1 \in \mathbb{N}$

- 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**

⊢

Modified Invariant **inv0\_2**

$d \in \mathbb{N}$

$n \in \mathbb{N}$

$n \leq d$

⊢

$n + 1 \leq d$

- This proof obligation is named: **ML\_out / inv0\_2 / INV**

**ML\_in**  
 $n := n - 1$

$$(n' = n - 1)$$

Axiom **axm0\_1**  
Invariant **inv0\_1**  
Invariant **inv0\_2**  
 $\vdash$   
Modified Invariant **inv0\_1**

$d \in \mathbb{N}$   
 $n \in \mathbb{N}$   
 $n \leq d$   
 $\vdash$   
 $n - 1 \in \mathbb{N}$

- 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**

⊢

Modified Invariant **inv0\_2**

$d \in \mathbb{N}$

$n \in \mathbb{N}$

$n \leq d$

⊢

$n - 1 \leq d$

- This proof obligation is named: **ML\_in / inv0\_2 / INV**

ML\_out / **inv0\_1** / INV

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

ML\_out / **inv0\_2** / INV

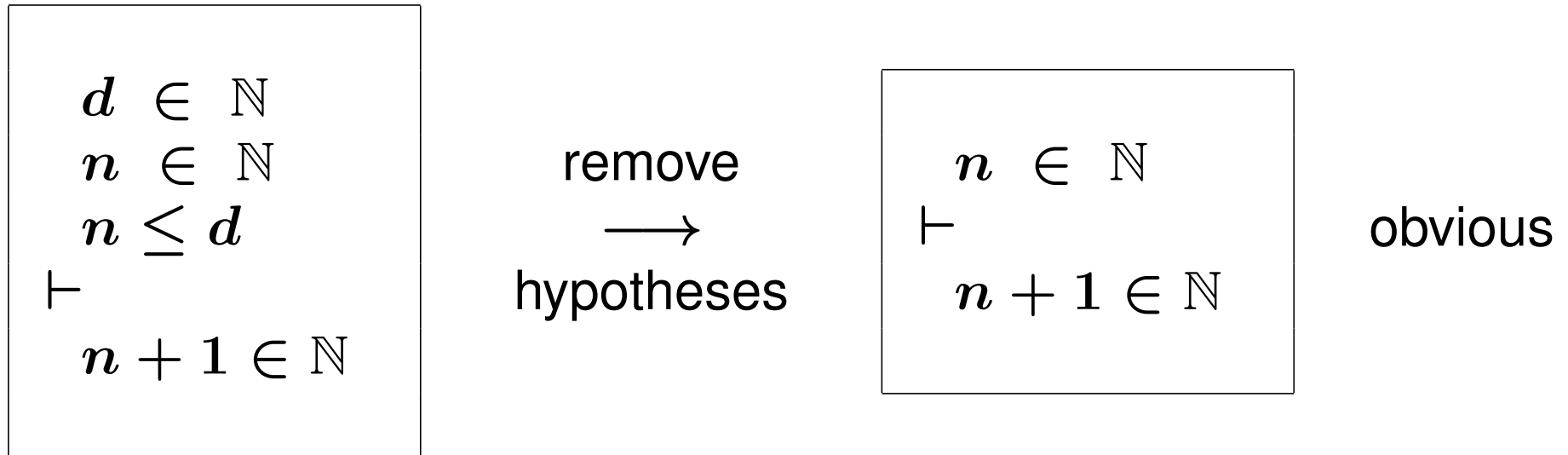
$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ \vdash \\ n + 1 \leq d \end{array}$$

ML\_in / **inv0\_1** / INV

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

ML\_in / **inv0\_2** / INV

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



- In the first step, we **remove some irrelevant hypotheses**
- In the second and final step, we **accept the sequent as it is**
- We have implicitly applied **inference rules**
- For **rigorous reasoning** we will make these rules **explicit**

$$\frac{\mathbf{H_1 \vdash G_1} \quad \dots \quad \mathbf{H_n \vdash G_n}}{\mathbf{H \vdash G}} \quad \mathbf{RULE\_NAME}$$

- Above horizontal line:  $n$  sequents called **antecedents** ( $n \geq 0$ )
- Below horizontal line: exactly one sequent called **consequent**
- To prove the consequent, **it is sufficient** to prove the antecedents
- A rule with no antecedent ( $n = 0$ ) is called an **axiom**

- The rule that removes hypotheses can be stated as follows:

$$\frac{H \vdash G}{H, H' \vdash G} \quad \text{MON}$$

- It expresses the **monotonicity** of the hypotheses

## - The Second Peano Axiom

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

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



- Axioms about **ordering relations** on the integers

$$\frac{n < m}{n + 1 \leq m} \quad \text{INC}$$

$$\frac{n \leq m}{n - 1 \leq m} \quad \text{DEC}$$

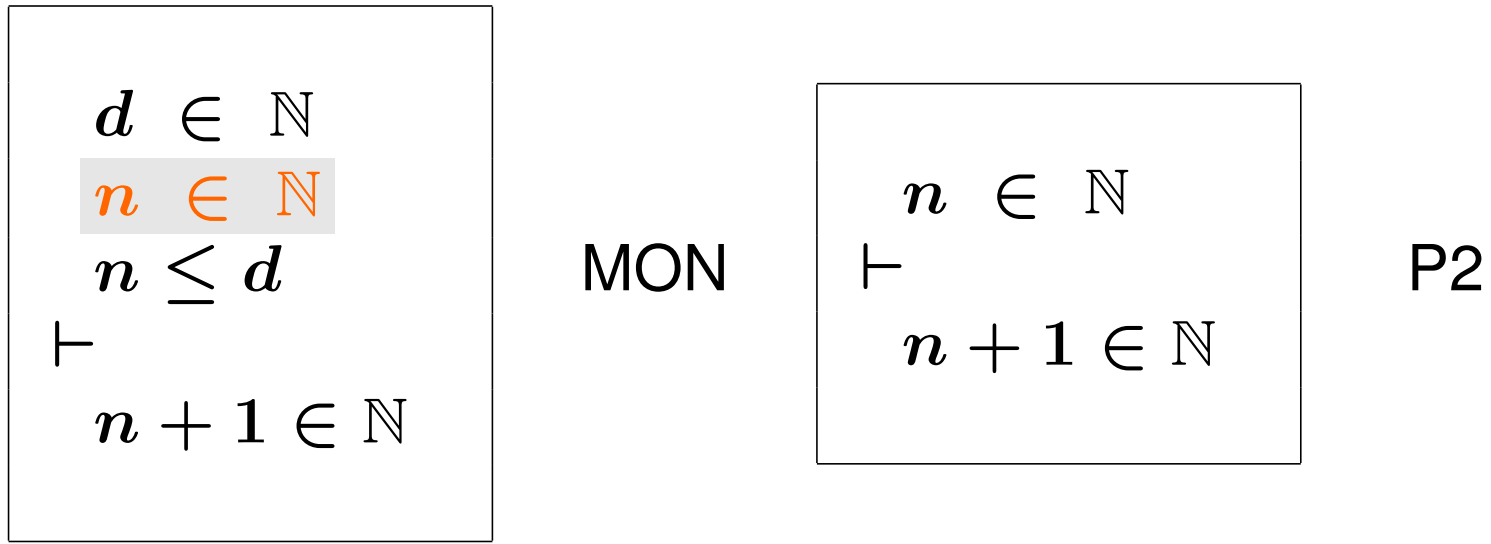
- Consider again the **2nd Peano axiom**:

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

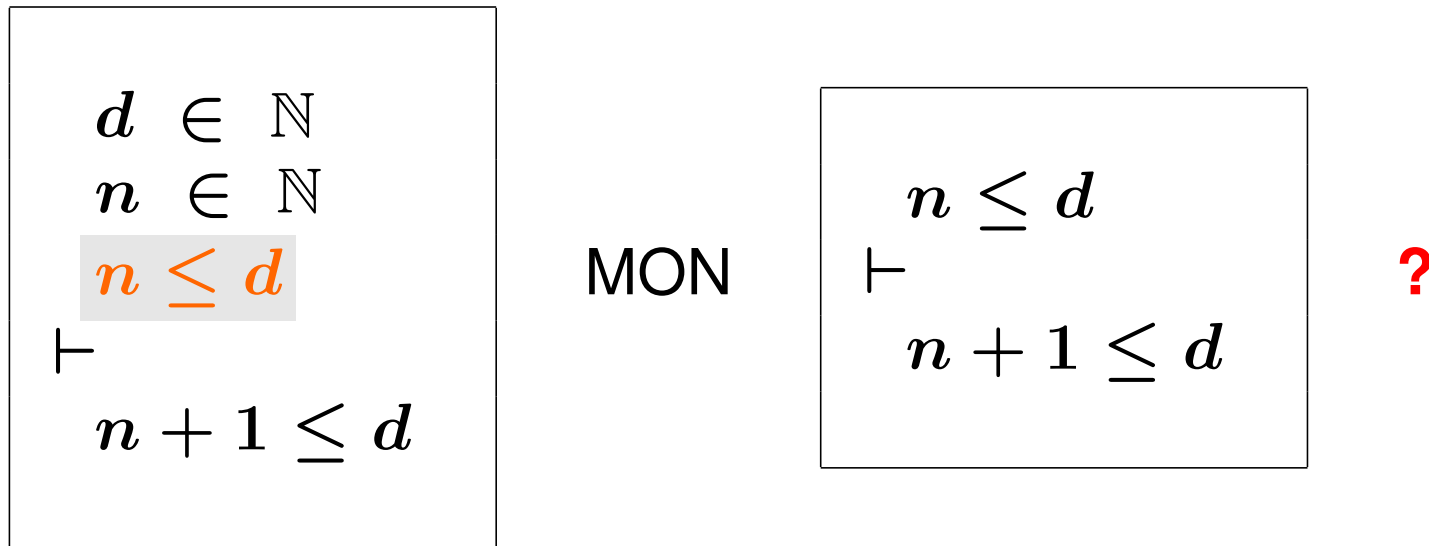
- It is a **rule schema** where **n** is called a **meta-variable**
- It can be applied to following sequent by **matching**  $a + b$  with **n**:

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

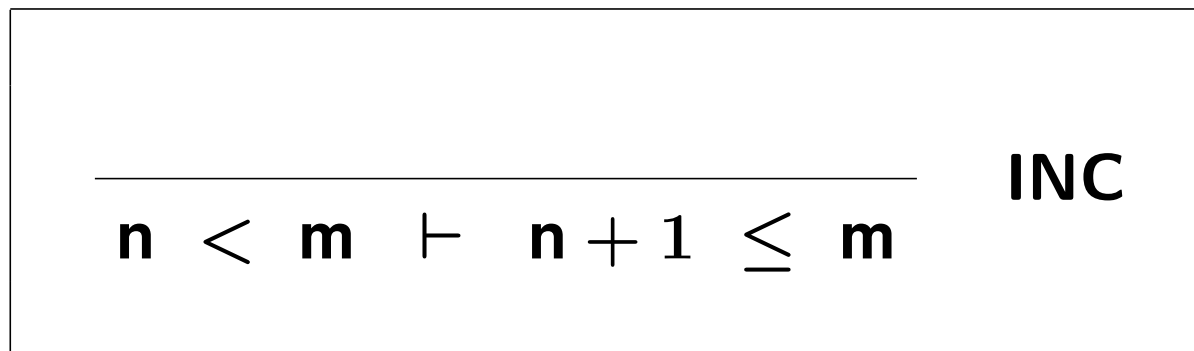
- 
- A **proof** is a **tree of sequents** with axioms at the leaves.
  - The rules applied to the **leaves are axioms**.
  - Each sequent is **labeled with** (name of) **proof rule** applied to it.
  - The sequent at the root of the tree is called the **root sequent**.
  - The **purpose** of a proof is to establish the **truth** of its root sequent.

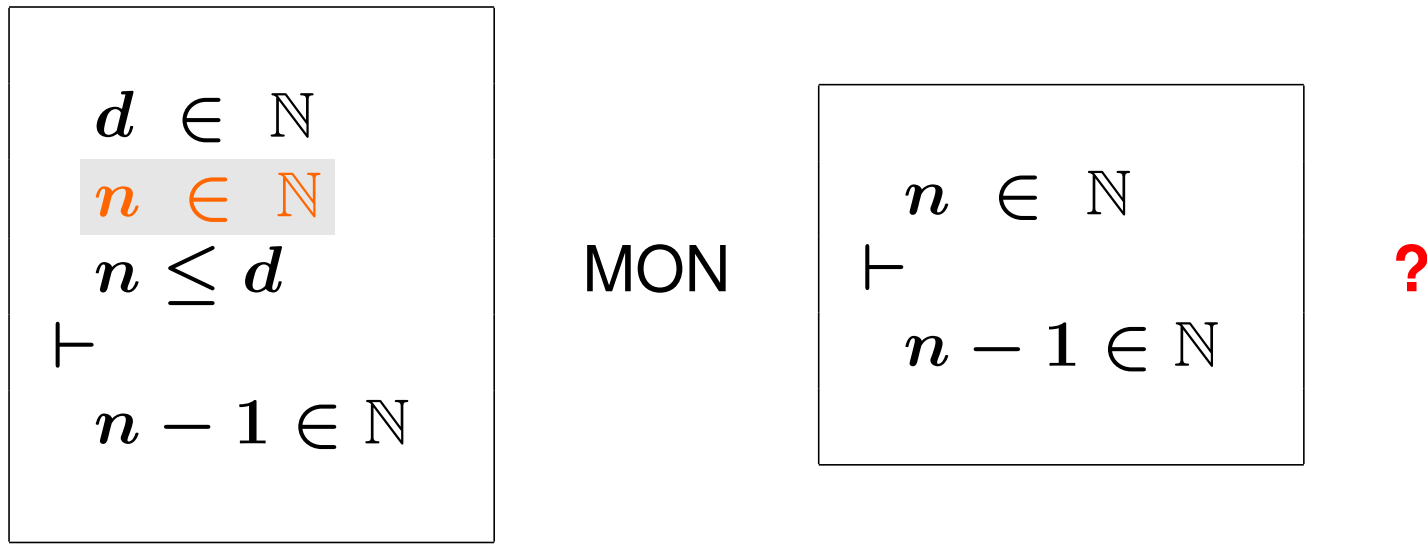


- Proof requires only application of two rules: **MON** and **P2**



- We put a **?** to indicate that **we have no rule to apply**
- The proof fails: we cannot conclude with rule **INC** ( **$n < d$  needed**)





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

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

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

MON

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

DEC

$$\frac{}{n \leq m \vdash n - 1 \leq m} \quad \text{DEC}$$

- We needed hypothesis  $n < d$  to prove  $\text{ML\_out} / \text{inv0\_2} / \text{INV}$
- We needed hypothesis  $0 < n$  to prove  $\text{ML\_in} / \text{inv0\_1} / \text{INV}$

**ML\_out**

$n := n + 1$

**ML\_in**

$n := n - 1$

- 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



ML\_out

**when**

$n < d$

**then**

$n := n + 1$

**end**

ML\_in

**when**

$0 < n$

**then**

$n := n - 1$

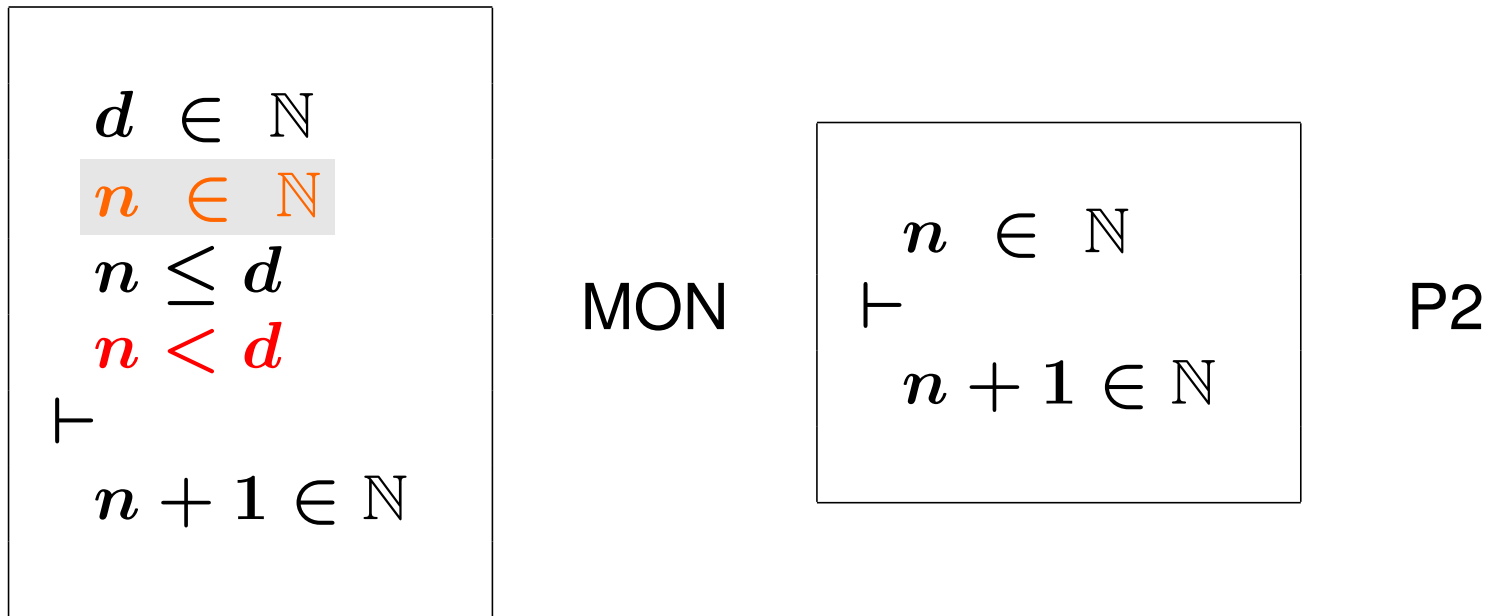
**end**

- We are adding **guards** to the events
- The guard is the **necessary condition** for an event to “occur”

- 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 PO as follows:

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

$A(c)$	INV
$I(c, v)$	
Guard of the event	
$\vdash$	
$I_i(c, E(c, v))$	



- Adding new assumptions to a sequent **does not affect its provability**

$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ \textcolor{red}{n} < \textcolor{red}{d} \\ \vdash \\ n + 1 \leq d \end{array}$	MON	$\begin{array}{l} n < d \\ \vdash \\ n + 1 \leq d \end{array}$	<b>INC</b>
--	-----	--	------------

- Now we can conclude the proof using rule **INC**

$\frac{n < m \quad \vdash \quad n + 1 \leq m}{\text{INC}}$
--

$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ \textcolor{red}{0} < \textcolor{red}{n} \\ \vdash \\ n - 1 \in \mathbb{N} \end{array}$	MON	$\begin{array}{l} 0 < n \\ \vdash \\ n - 1 \in \mathbb{N} \end{array}$	<b>P2'</b>
--	-----	--	------------

- Now we can conclude the proof using rule **P2'**

$\frac{}{0 < n \vdash n - 1 \in \mathbb{N}}$	<b>P2'</b>
--	------------

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

MON

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

DEC

- Again, the proof still works after the addition of a new assumption

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

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

$$n \leq d$$

$$n < d$$

$$\vdash$$

$$n + 1 \in \mathbb{N}$$

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

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

$$n \leq d$$

$$n < d$$

$$\vdash$$

$$n + 1 \leq d$$

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

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

$$n \leq d$$

$$0 < n$$

$$\vdash$$

$$n - 1 \in \mathbb{N}$$

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

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

$$n \leq d$$

$$0 < n$$

$$\vdash$$

$$n - 1 \leq d$$

- Our system must be **initialized** (with no car in the island-bridge)
- The initialization event is **never guarded**
- It does **not mention any variable** on the right hand side of  $:=$
- Its 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 PO is the following:

$\begin{array}{c} \text{Axioms} \\ \vdash \\ \text{Modified Invariant} \end{array}$	$\begin{array}{c} A(c) \\ \vdash \\ I_i(c, K(c)) \end{array}$	$\text{INV}$
---	---	--------------

**axm0\_1**  
 $\vdash$   
Modified **inv0\_1**

$$\begin{array}{l} d \in \mathbb{N} \\ \vdash \\ 0 \in \mathbb{N} \end{array}$$

**inv0\_1** / INV

**axm0\_1**  
 $\vdash$   
Modified **inv0\_2**

$$\begin{array}{l} d \in \mathbb{N} \\ \vdash \\ 0 \leq d \end{array}$$

**inv0\_2** / INV

## - First Peano Axiom

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

## - Third Peano Axiom (slightly modified)

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

$$\begin{array}{l} d \in \mathbb{N} \\ \vdash \\ 0 \in \mathbb{N} \end{array}$$

MON

$$\begin{array}{l} \vdash \\ 0 \in \mathbb{N} \end{array}$$

P1

$$\begin{array}{l} d \in \mathbb{N} \\ \vdash \\ 0 \leq d \end{array}$$

P3

- 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  $G_1(c, v), \dots, G_m(c, v)$  of the events
- We have to prove the following:

$\begin{array}{l} A(c) \\ I(c, v) \\ \vdash \\ G_1(c, v) \vee \dots \vee G_m(c, v) \end{array}$	$\text{DLF}$
---	--------------

**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 continue our proof by a case analysis:
  - case 1:  $n = d$
  - case 2:  $n < d$

- Proof by **case analysis**

$$\frac{H, P \vdash R \quad H, Q \vdash R}{H, P \vee Q \vdash R} \text{ OR\_L}$$

- Choice for proving a **disjunctive goal**

$$\frac{H \vdash P}{H \vdash P \vee Q} \text{ OR\_R1}$$

$$\frac{H \vdash Q}{H \vdash P \vee Q} \text{ OR\_R2}$$



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

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

$$n \leq d$$

 $\vdash$ 

$$n < d \vee 0 < n$$

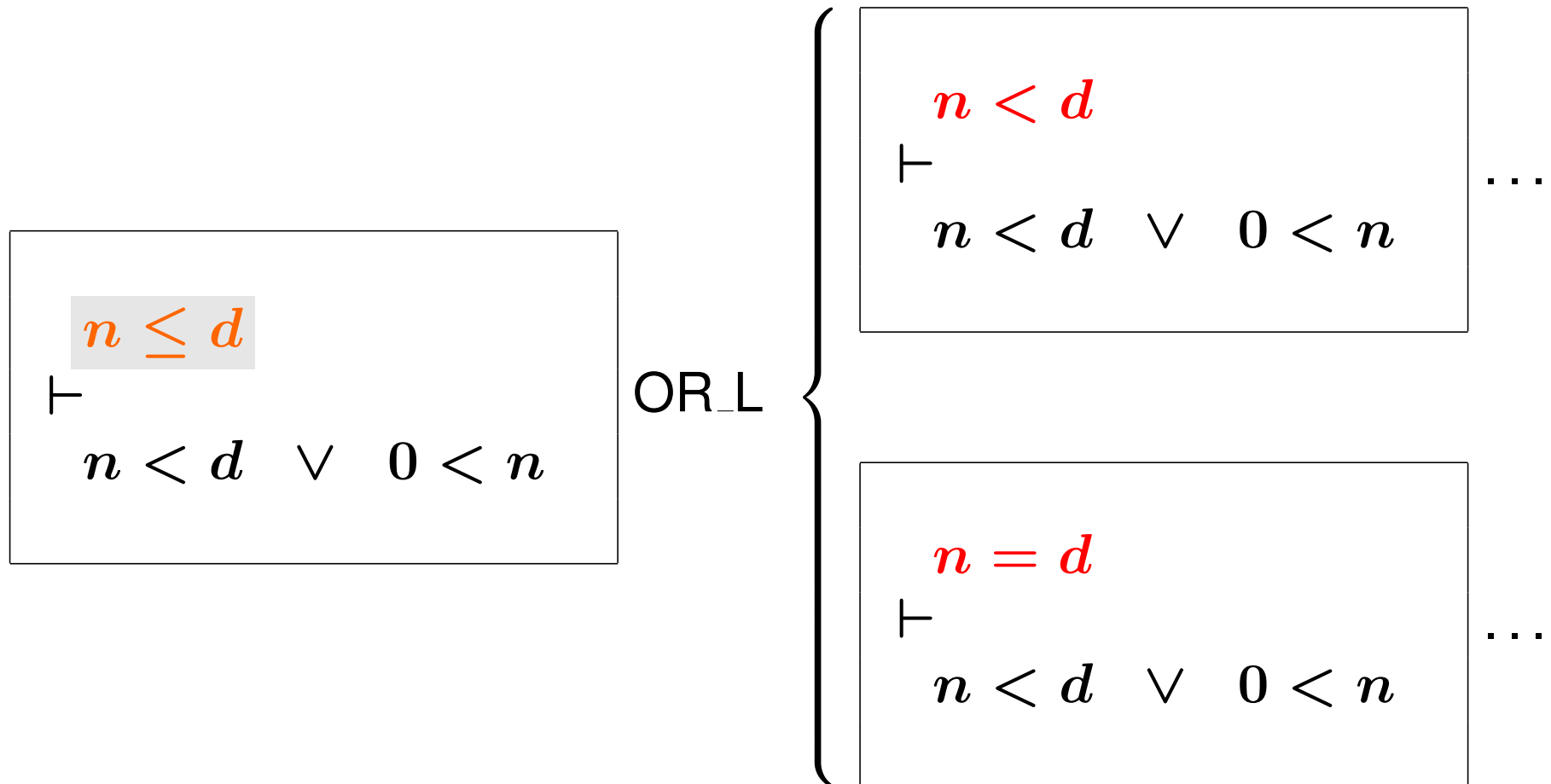
MON

$$n \leq d$$

 $\vdash$ 

$$n < d \vee 0 < n$$

...



$$\left\{ \begin{array}{l} \boxed{\begin{array}{l} n < d \\ \vdash \\ n < d \vee 0 < n \end{array}} \quad \text{OR\_R1} \quad \boxed{n < d \vdash n < d} \quad ? \\[1em] \boxed{\begin{array}{l} n = d \\ \vdash \\ n < d \vee 0 < n \end{array}} \quad ? \end{array} \right.$$

- The first ? seems to be obvious
- The second ? can be (partially) solved by applying the equality

- The **identity axiom** (conclusion holds by hypothesis)

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

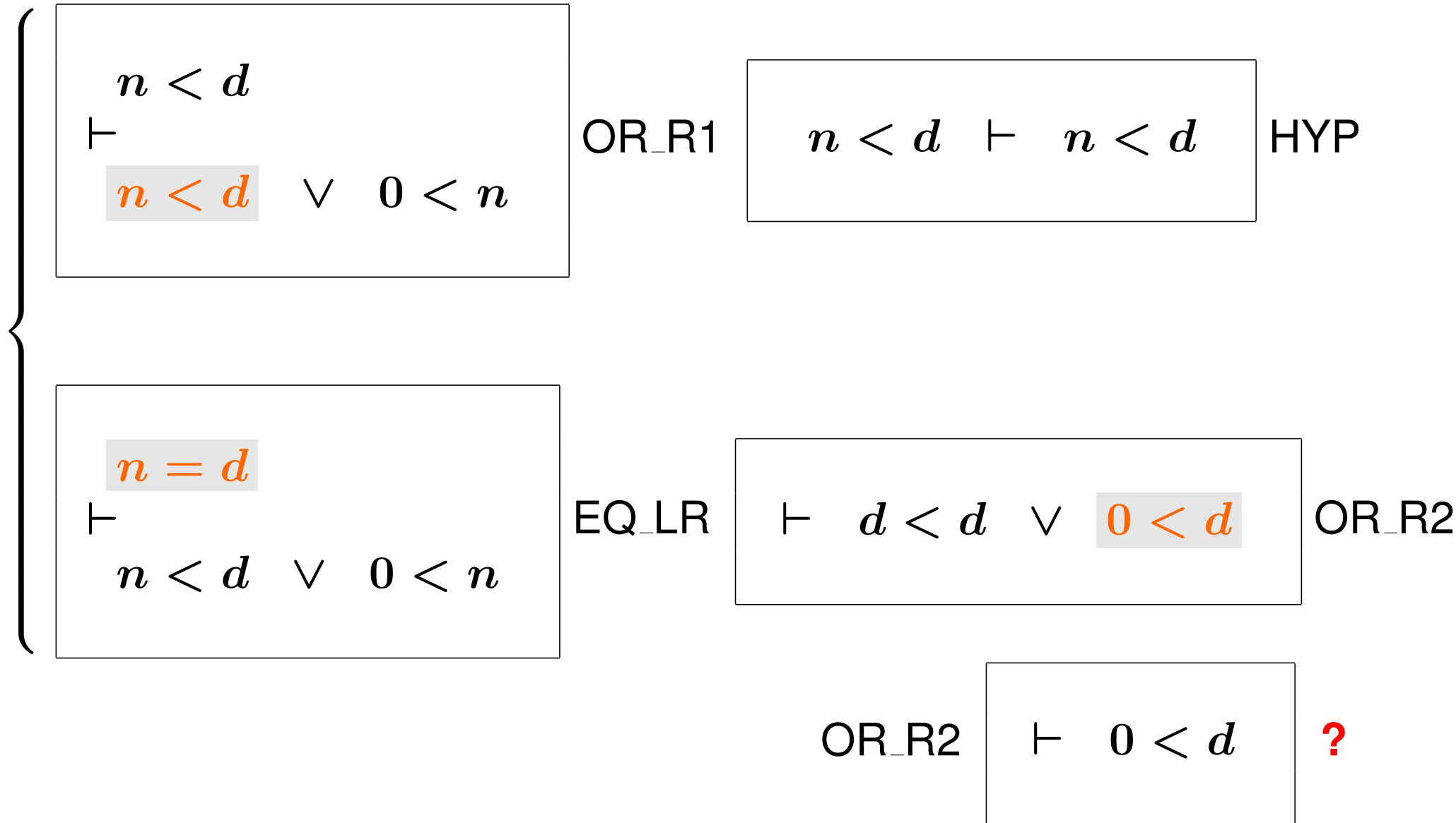
- **Rewriting an equality** (**EQ\_LR**) and **reflexivity of equality** (**EQL**)

$$\frac{H(F), E = F \vdash P(F)}{H(E), E = F \vdash P(E)} \text{EQ\_LR}$$

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

# Proof of Deadlock Freedom (end)

81



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

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

$$\text{axm0\_2: } 0 < d$$

- Thanks to the **proofs**, we discovered **3 errors**
- They were corrected by:
  - **adding guards** to both events
  - **adding an axiom**
- The **interaction of modeling and proving** is an essential element of Formal Methods with Proofs

- We have seen three kinds of proof obligations:
  - The **Invariant Establishment** PO: INV
  - The **Invariant Preservation** PO: INV
  - The **Deadlock Freedom** PO (optional): DLF



Axioms $\vdash$ Modified Invariant	INV
--	-----

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

Axiom Invariants $\vdash$ Disjunction of the guards	DLF
--	-----

**constant:**  $d$

**variable:**  $n$

**axm0\_1:**  $d \in \mathbb{N}$

**axm0\_2:**  $d > 0$

**inv0\_1:**  $n \in \mathbb{N}$

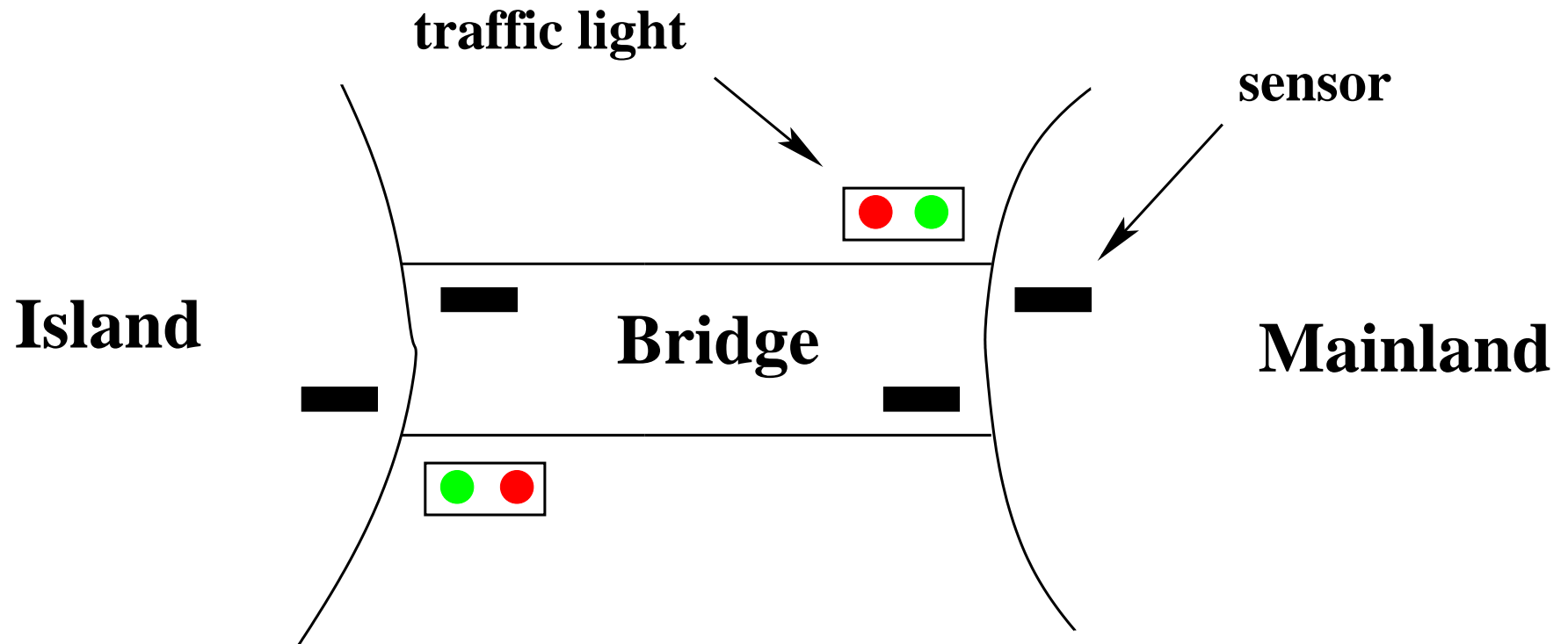
**inv0\_2:**  $n \leq d$

init  
 $n := 0$

ML\_out  
**when**  
     $n < d$   
**then**  
     $n := n + 1$   
**end**

ML\_in  
**when**  
     $0 < n$   
**then**  
     $n := n - 1$   
**end**

- **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)

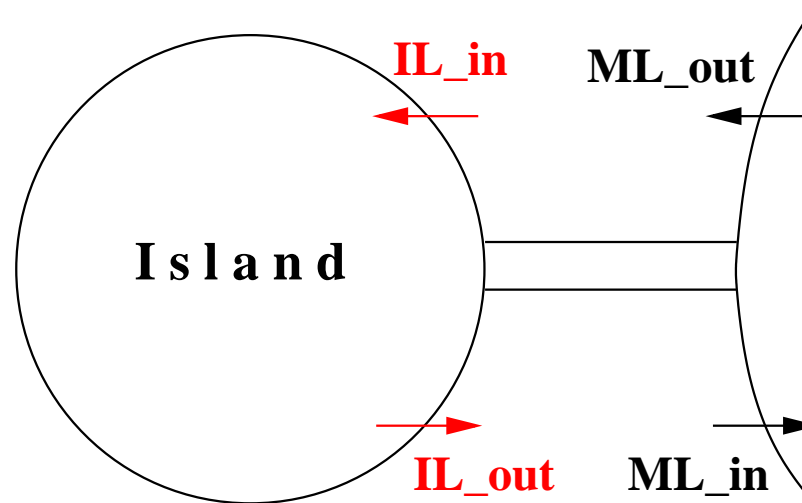
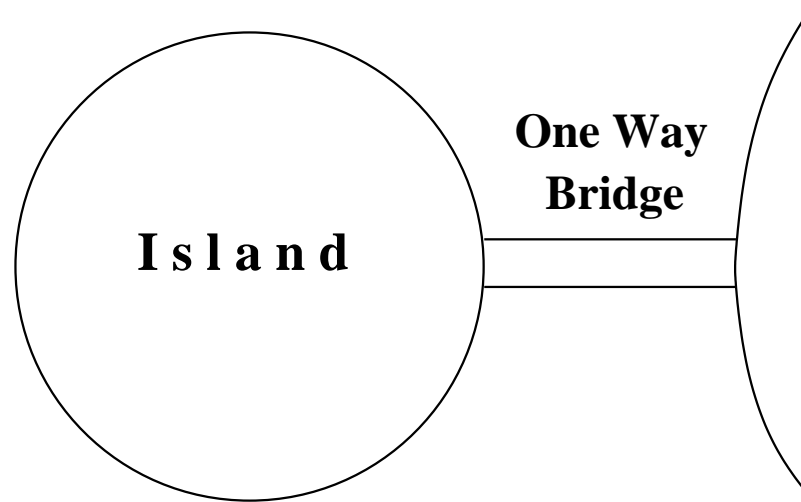


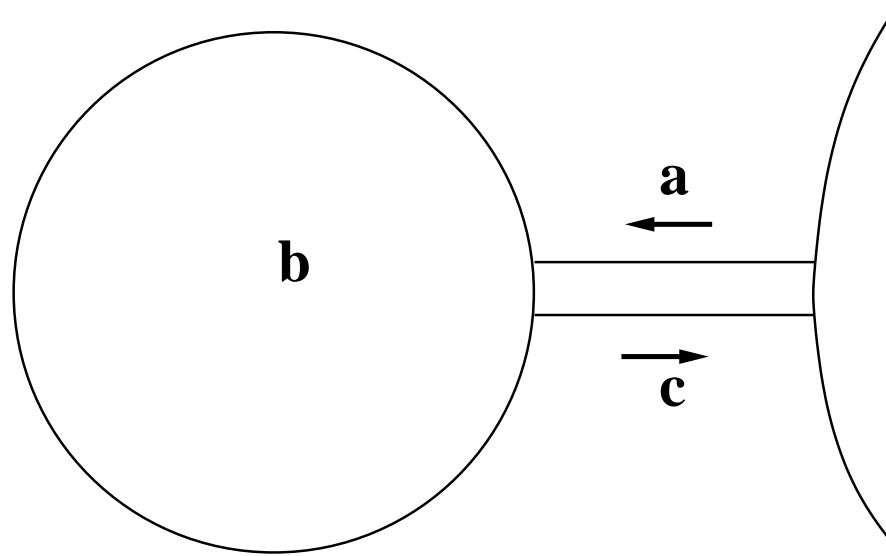
- We go down with our parachute
- Our view of the system gets more accurate
- We introduce the bridge and separate it from the island
- We refine the state and the events
- We also add two new events: IL\_in and IL\_out
- We are focusing on FUN-3: one-way bridge

# First Refinement: Introducing a one Way Bridge

---

90





- $a$  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  $n$

# Refining the State: Formalizing Variables $a$ , $b$ , and $c$ 92

---

- Variables  $a$ ,  $b$ , and  $c$  denote **natural numbers**

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

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

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



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

$$a + b + c = n$$

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

$$a = 0 \quad \vee \quad c = 0$$

**constants:**  $d$

**variables:**  $a, b, c$

**inv1\_1:**  $a \in \mathbb{N}$

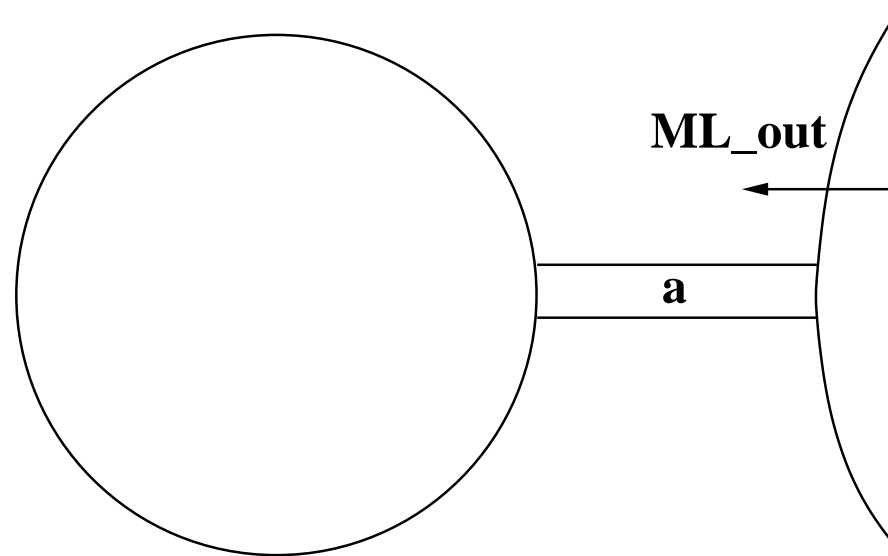
**inv1\_2:**  $b \in \mathbb{N}$

**inv1\_3:**  $c \in \mathbb{N}$

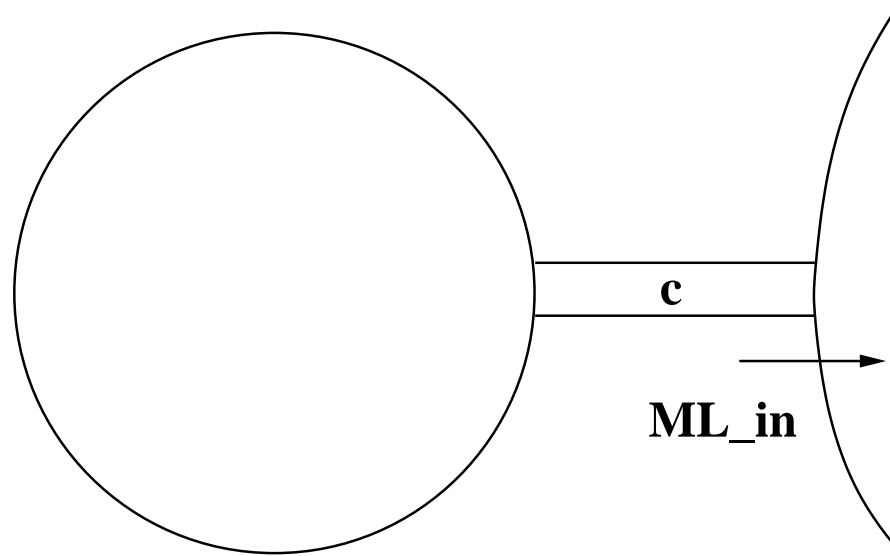
**inv1\_4:**  $a + b + c = n$

**inv1\_5:**  $a = 0 \vee c = 0$

- Invariants **inv1\_1** to **inv1\_5** are called the **concrete invariants**
- **inv1\_4** glues the abstract state,  $n$ , to the **concrete state**,  $a, b, c$



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



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

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 unmodified variables:

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

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

The concrete model behaves as specified by the abstract model  
(i.e., concrete model does not exhibit any new behaviors)

To show this we have to prove that

1. every concrete event is simulated by its abstract counterpart  
(event refinement: following slides)
2. to every concrete initial state corresponds an abstract one  
(initial state refinement: later)

We will make these two conditions more precise and formalize them as proof obligations.

```
(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** seem to be **compatible**

```
(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**



- The concrete guard is **stronger** than the abstract one
- Each concrete action is **compatible** with its abstract counterpart

Constants  $c$  with axioms  $A(c)$

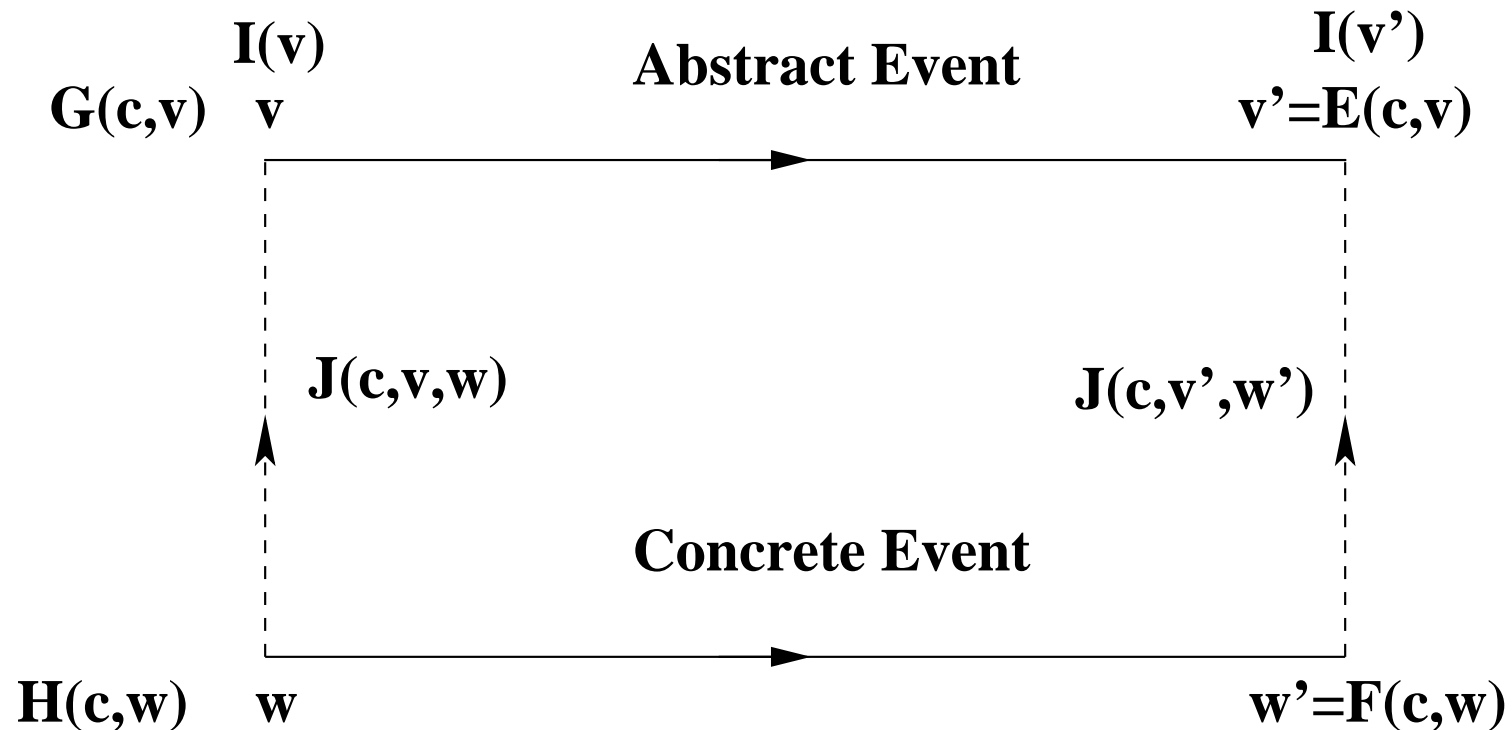
Abstract variables  $v$  with abstract invariant  $I(c, v)$

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

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

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

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



1. The concrete guard is **stronger** than the abstract one  
(**Guard Strengthening**, following slides)
2. Each concrete action is **simulated by** its abstract counterpart  
(**Concrete Invariant Preservation**, later)

<p>Axioms Abstract Invariant Concrete Invariant Concrete Guard <math>\vdash</math> Abstract Guard</p>	<p><math>A(c)</math> <math>I(c, v)</math> <math>J(c, v, w)</math> <math>H(c, w)</math> <math>\vdash</math> <math>G_i(c, v)</math></p>	<p>GRD</p>
---	---	------------

- 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

⊢

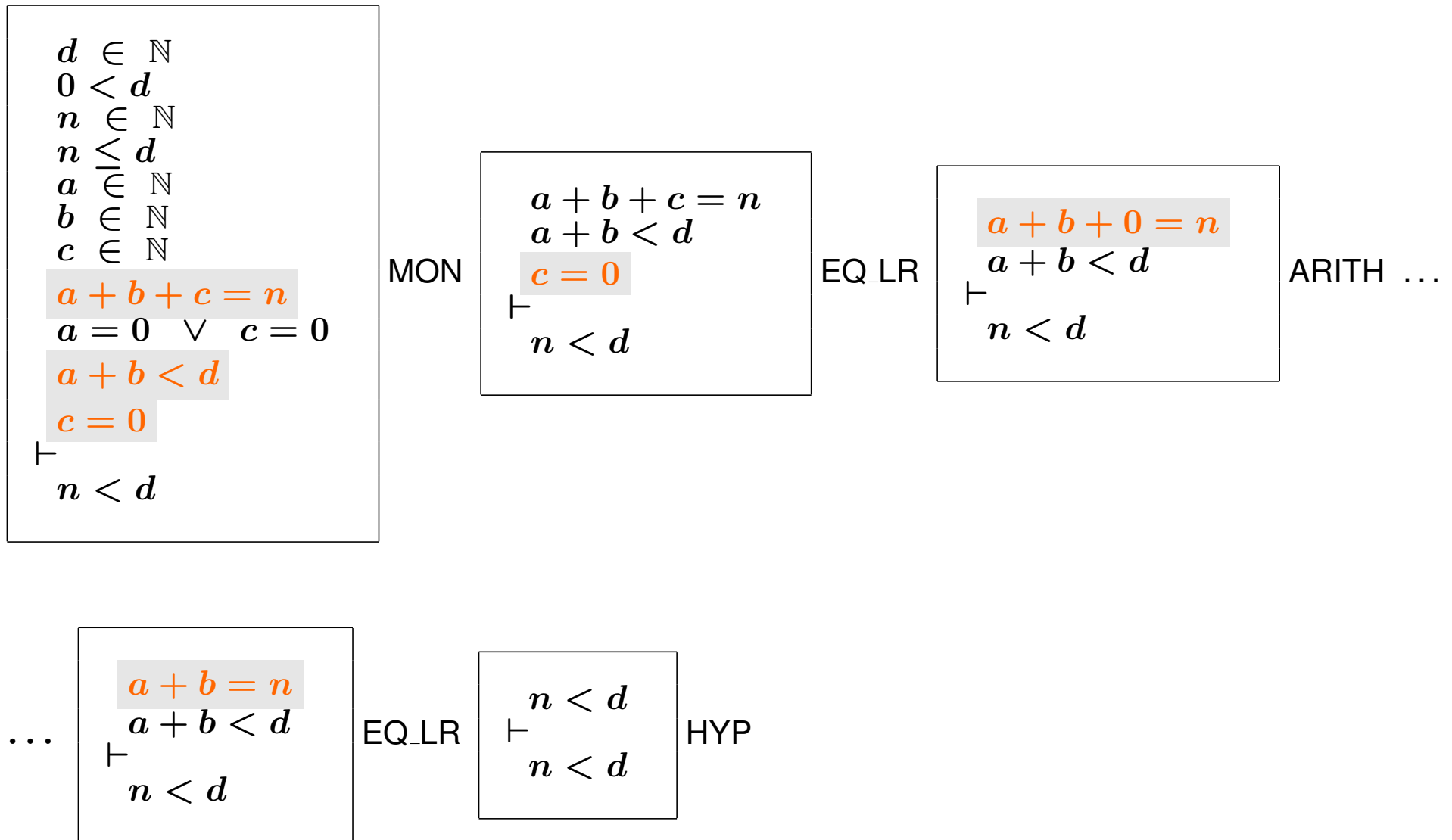
Abstract guard of ML\_out

$d \in \mathbb{N}$   
 $0 < d$   
 $n \in \mathbb{N}$   
 $n \leq d$   
 $a \in \mathbb{N}$   
 $b \in \mathbb{N}$   
 $c \in \mathbb{N}$   
 $a + b + c = n$   
 $a = 0 \vee c = 0$   
 $a + b < d$   
 $c = 0$   
⊢  
 $n < d$

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**



The "rule" name ARITH stands for **simple arithmetic simplifications**.

axm0\_1  
axm0\_2  
inv0\_1  
inv0\_2  
inv1\_1  
inv1\_2  
inv1\_3  
inv1\_4  
inv1\_5

Concrete guard of ML\_in

⊢

Abstract guard of ML\_in

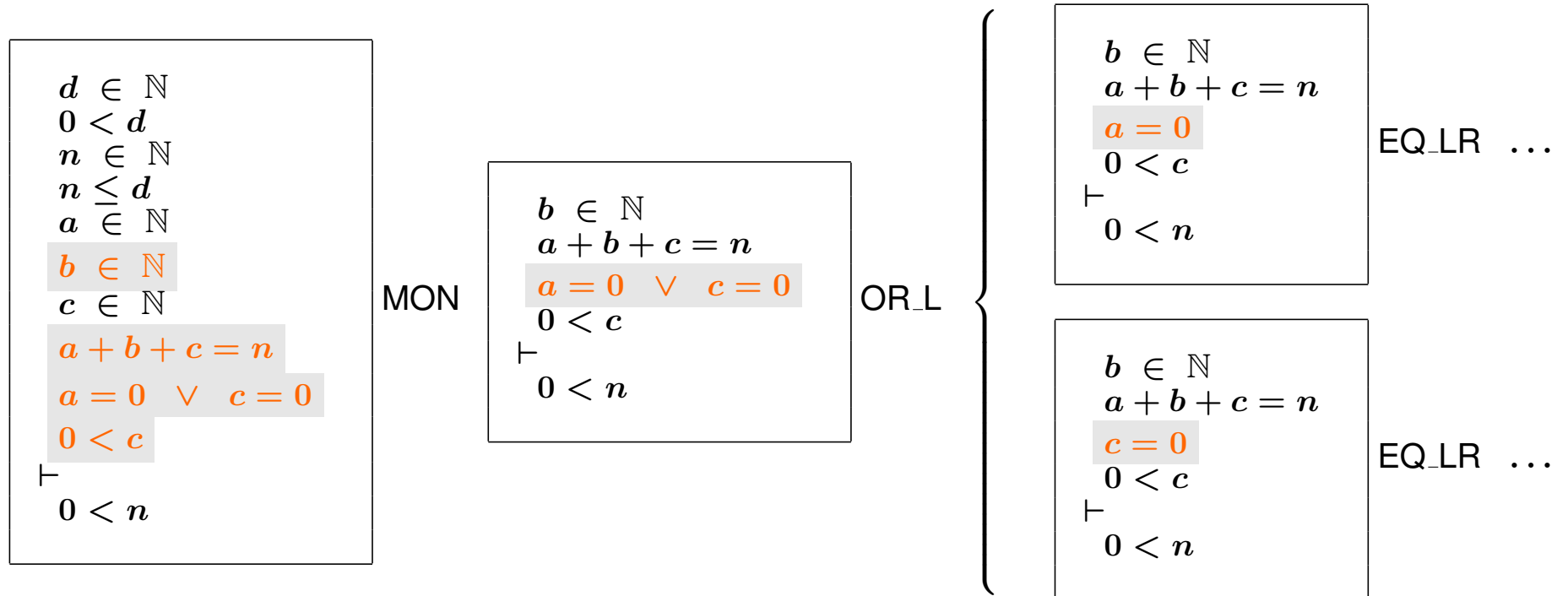
$d \in \mathbb{N}$   
 $0 < d$   
 $n \in \mathbb{N}$   
 $n \leq d$   
 $a \in \mathbb{N}$   
 $b \in \mathbb{N}$   
 $c \in \mathbb{N}$   
 $a + b + c = n$   
 $a = 0 \vee c = 0$   
 $0 < c$   
⊢  
 $0 < n$

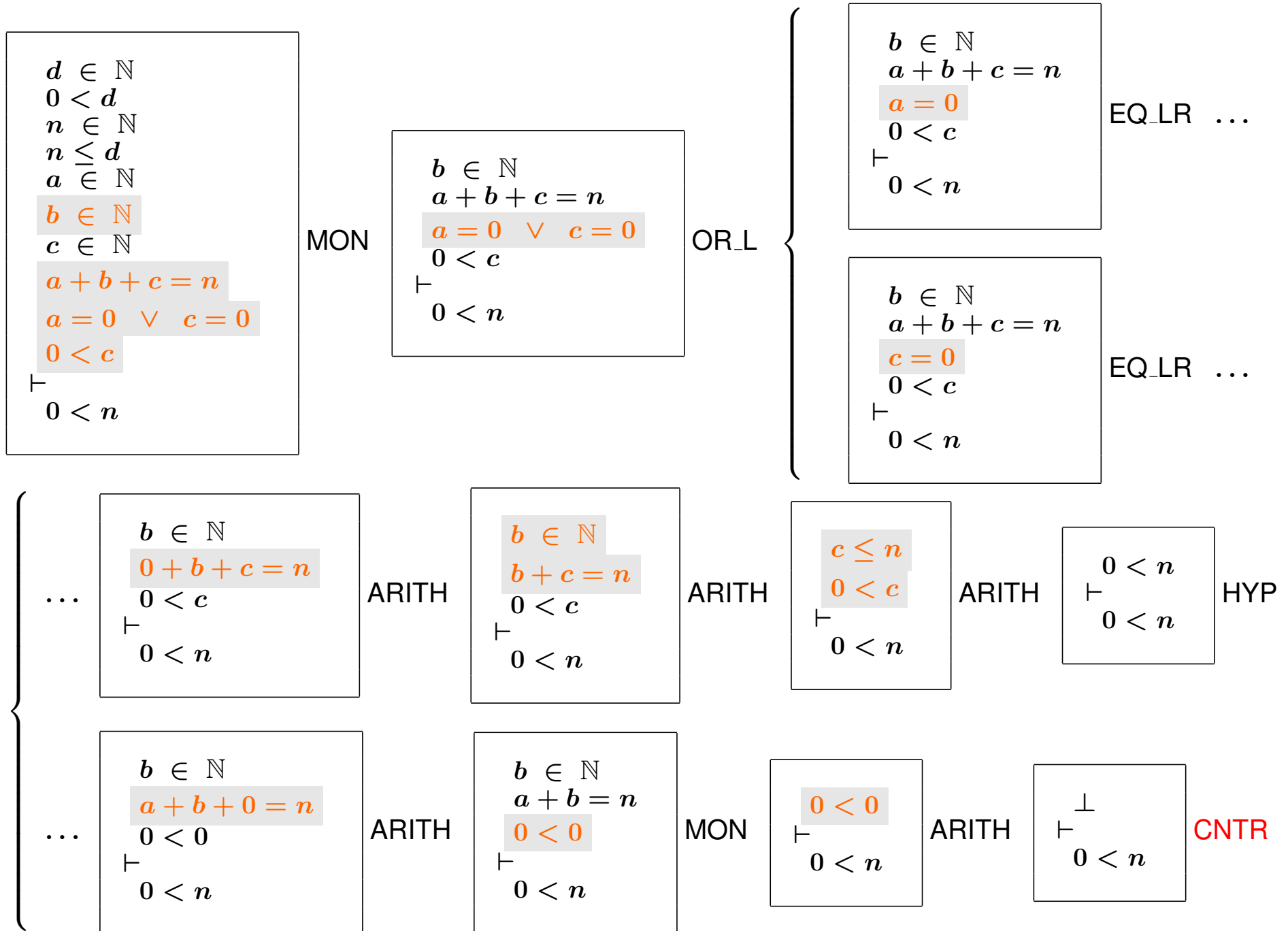
ML\_in / GRD

(abstract-)ML\_in  
**when**  
     $0 < n$   
**then**  
     $n := n - 1$   
**end**

(concrete-)ML\_in  
**when**  
     $0 < c$   
**then**  
     $c := c - 1$   
**end**

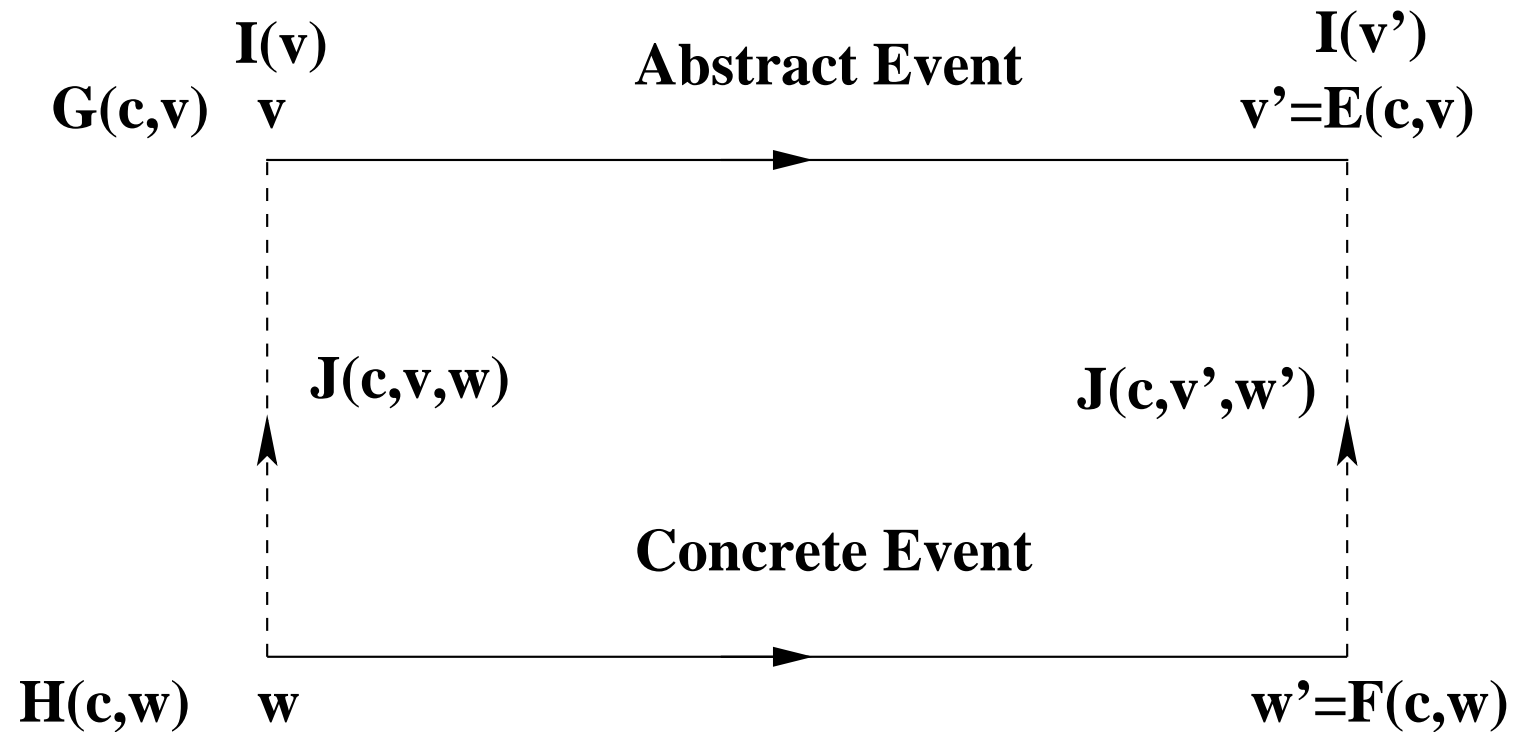






- In the previous proof, we have used an additional inference rule
- It says that a **false hypothesis entails any goal**

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



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
--	--	-----

- 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

axm0\_1  
axm0\_2  
inv0\_1  
inv0\_2  
inv1\_1  
inv1\_2  
inv1\_3  
inv1\_4  
inv1\_5

Concrete guards of ML\_out

⊢

Modified Invariant **inv1\_4**

$d \in \mathbb{N}$   
 $0 < d$   
 $n \in \mathbb{N}$   
 $n \leq d$   
 $a \in \mathbb{N}$   
 $b \in \mathbb{N}$   
 $c \in \mathbb{N}$   
 $a + b + c = n$   
 $a = 0 \vee c = 0$   
 $a + b < d$   
 $c = 0$

⊢

$a + 1 + b + c = n + 1$

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**

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

MON

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

ARITH ...

$$\begin{array}{l}
 \dots \\
 \textcolor{brown}{a + b + c = n} \\
 \vdash \\
 a + b + c + 1 = n + 1
 \end{array}$$

EQ\_LR

$$\vdash n + 1 = n + 1$$

EQL



axm0\_1  
axm0\_2  
inv0\_1  
inv0\_2  
inv1\_1  
inv1\_2  
inv1\_3  
inv1\_4  
inv1\_5

Concrete guards of ML\_out

⊢

Modified Invariant **inv1\_5**

$$\begin{aligned} d &\in \mathbb{N} \\ 0 &< d \\ n &\in \mathbb{N} \\ n &\leq d \\ a &\in \mathbb{N} \\ b &\in \mathbb{N} \\ c &\in \mathbb{N} \\ a + b + c &= n \\ \textcolor{red}{a} = 0 &\vee c = 0 \\ a + b &< d \\ c &= 0 \end{aligned}$$

⊢

$$\textcolor{red}{a} + \textcolor{red}{1} = 0 \vee c = 0$$

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**

$\textcolor{red}{a} := \textcolor{red}{a} + 1$

**end**

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

MON

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

OR\_R2

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

HYP

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\_4**

$d \in \mathbb{N}$   
 $0 < d$   
 $n \in \mathbb{N}$   
 $n \leq d$   
 $a \in \mathbb{N}$   
 $b \in \mathbb{N}$   
 $c \in \mathbb{N}$   
 $a + b + c = n$   
 $a = 0 \vee c = 0$   
 $0 < c$

⊢

$a + b + c - 1 = n - 1$

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**

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

MON

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

EQ\_LR

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

EQL

axm0\_1  
axm0\_2  
inv0\_1  
inv0\_2  
inv1\_1  
inv1\_2  
inv1\_3  
inv1\_4  
inv1\_5

Concrete guards of ML\_in

⊢

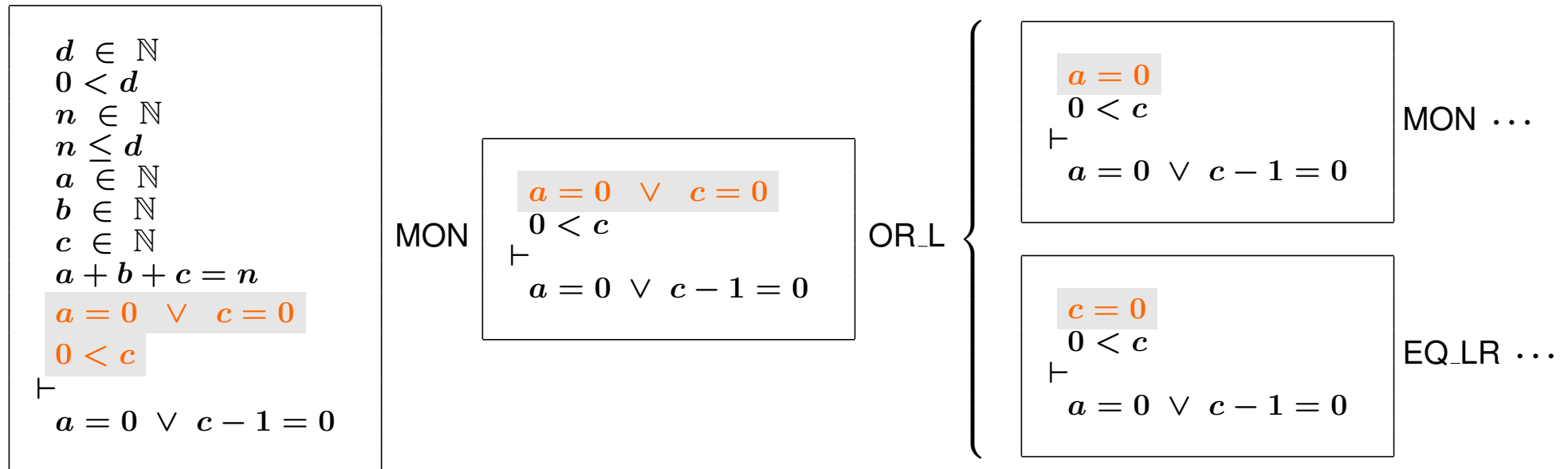
Modified Invariant **inv1\_5**

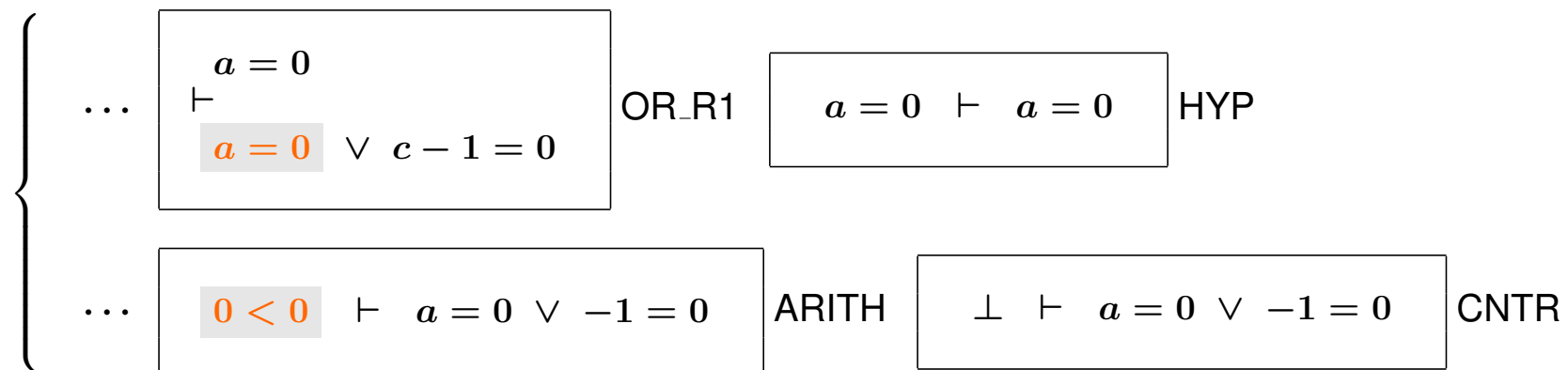
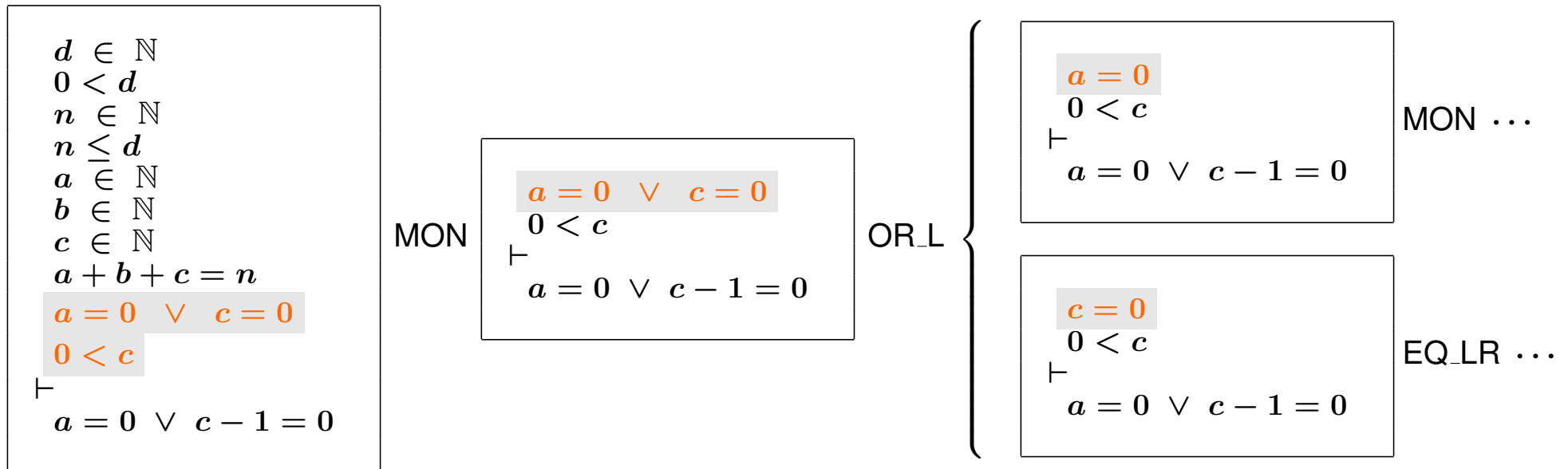
$$\begin{array}{l} d \in \mathbb{N} \\ 0 < d \\ n \in \mathbb{N} \\ n \leq d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ c \in \mathbb{N} \\ a + b + c = n \\ a = 0 \vee \textcolor{red}{c} = 0 \\ 0 < c \\ \vdash \\ a = 0 \vee \textcolor{red}{c} - \textcolor{red}{1} = 0 \end{array}$$

ML\_in / **inv1\_5** / INV

(abstract-)ML\_in  
**when**  
     $0 < n$   
**then**  
     $n := n - 1$   
**end**

(concrete-)ML\_in  
**when**  
     $0 < c$   
**then**  
     $\textcolor{red}{c} := \textcolor{red}{c} - 1$   
**end**





- Concrete initialization

init

$$a := 0$$
$$b := 0$$
$$c := 0$$

- Corresponding after predicate

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



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  $w' = L(c)$

<p>Axioms <math>\vdash</math> Modified concrete invariants</p>	<p><math>A(c)</math> <math>\vdash</math> <math>J_j(c, K(c), L(c))</math></p>	<p>INV</p>
--	--	------------

- 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

**axm0\_1**

**axm0\_2**

⊢

Modified concrete invariant **inv1\_4**  
 $(a + b + c = n)$

$d \in \mathbb{N}$

$d > 0$

⊢

$0 + 0 + 0 = 0$

**axm0\_1**

**axm0\_2**

⊢

Modified concrete invariant **inv1\_5**  
 $(a = 0 \vee c = 0)$

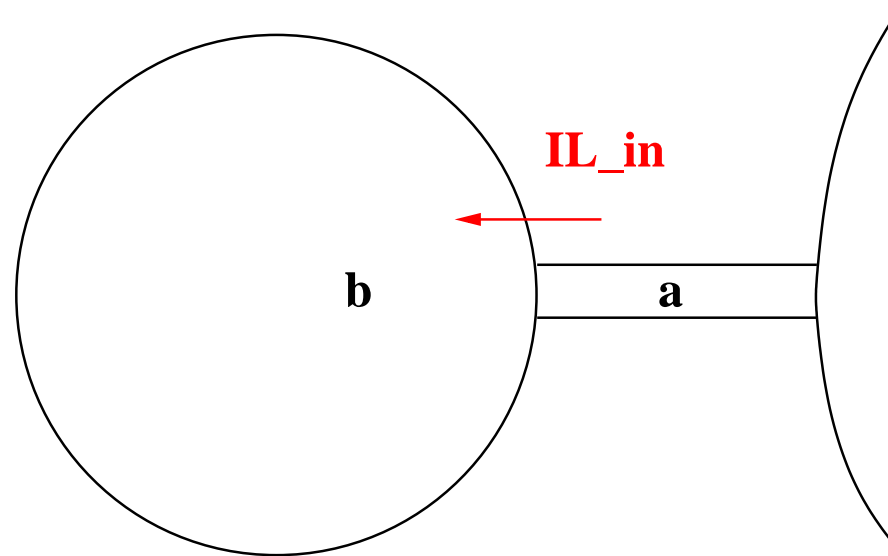
$d \in \mathbb{N}$

$d > 0$

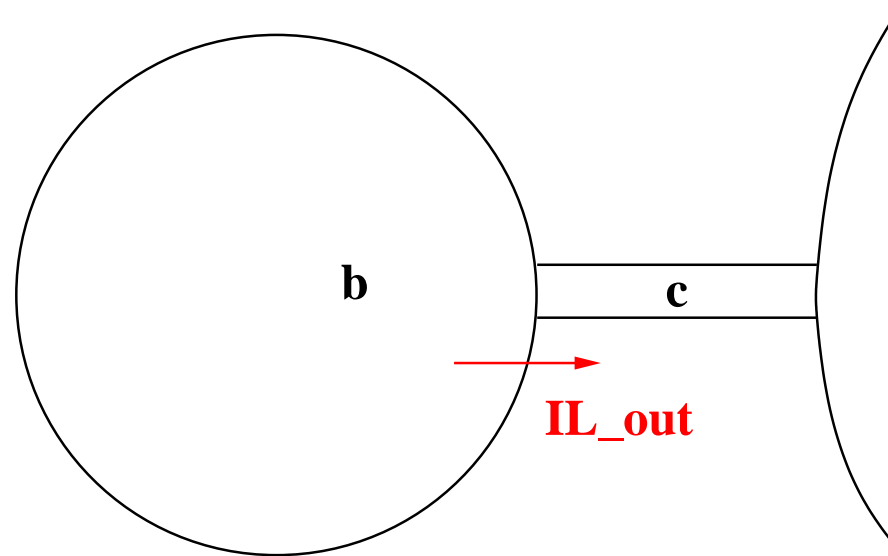
⊢

$0 = 0 \vee 0 = 0$

- 
- new events add transitions that have **no abstract counterpart**
  - can be seen as a kind of **internal steps** (w.r.t. abstract model)
  - can only be seen by an **observer** who is “**zooming in**”
  - **temporal refinement**: refined model has a finer time granularity



```
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
```

```
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 before-after predicate of **skip** in the **initial model**

$$n' = n$$

The before-after predicate of **skip** in the **first refinement**

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

The guard of the **skip** event is **true**.



- (1) A new event must **refine an implicit event**, made of a **skip action**
  - Guard strengthening is **trivial**
  - Need to prove **invariant refinement**
  
- (2) The new events **must not diverge**
  - To prove this we have to exhibit a **variant**
  - The variant yields a **natural number** (could be more complex)
  - Each new event must **decrease this variant**

- 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

axm0\_1  
axm0\_2  
inv0\_1  
inv0\_2  
inv1\_1  
inv1\_2  
inv1\_3  
inv1\_4  
inv1\_5

Concrete guards of IL\_in

⊢

Modified Invariant **inv1\_4**

$d \in \mathbb{N}$   
 $0 < d$   
 $n \in \mathbb{N}$   
 $n \leq d$   
 $a \in \mathbb{N}$   
 $b \in \mathbb{N}$   
 $c \in \mathbb{N}$   
 $a + b + c = n$   
 $a = 0 \vee c = 0$   
 $0 < a$

⊢

$a - 1 + b + 1 + c = n$

IL\_in / **inv1\_4** / INV

IL\_in

**when**

$0 < a$

**then**

$a := a - 1$

$b := b + 1$

**end**

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

$$0 < d$$

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

$$n \leq d$$

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

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

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

$$a + b + c = n$$

$$a = 0 \vee c = 0$$

$$0 < a$$

$$\vdash$$

$$a - 1 + b + 1 + c = n$$

MON

$$a + b + c = n$$

$$\vdash$$

$$a - 1 + b + 1 + c = n$$

ARITH

$$a + b + c = n$$

$$\vdash$$

$$a + b + c = n$$

HYP

axm0\_1  
axm0\_2  
inv0\_1  
inv0\_2  
inv1\_1  
inv1\_2  
inv1\_3  
inv1\_4  
inv1\_5

Concrete guards of IL\_in

⊢  
Modified Invariant **inv1\_5**

$d \in \mathbb{N}$   
 $0 < d$   
 $n \in \mathbb{N}$   
 $n \leq d$   
 $a \in \mathbb{N}$   
 $b \in \mathbb{N}$   
 $c \in \mathbb{N}$   
 $a + b + c = n$   
 $a = 0 \vee c = 0$   
 $0 < a$

⊢  
 $a - 1 = 0 \vee c = 0$

IL\_in / **inv1\_5** / INV

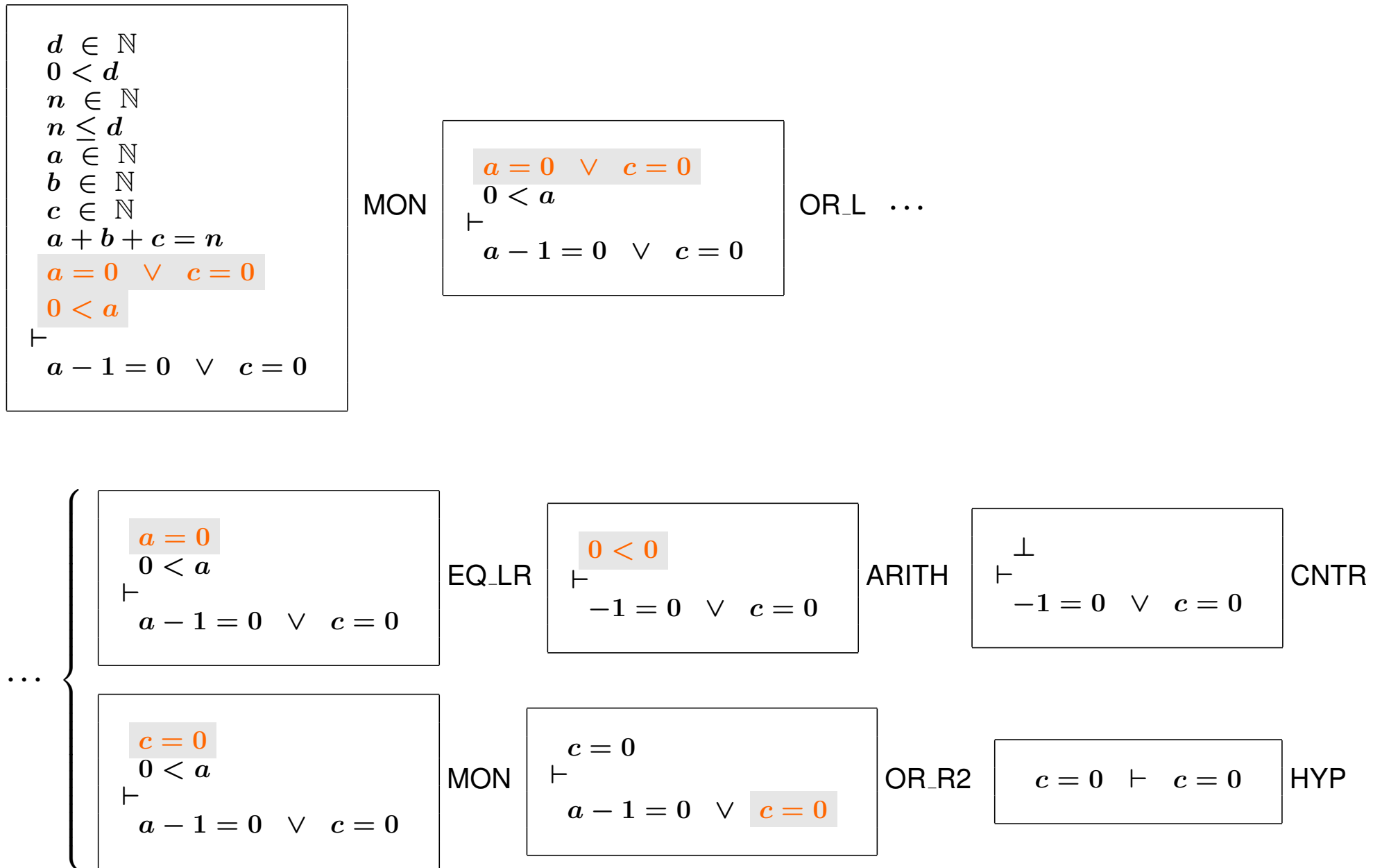
IL\_in  
**when**  
     $0 < a$   
**then**  
     $a := a - 1$   
     $b := b + 1$   
**end**

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

MON

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

OR\_L ...



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

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

Variant  $V(c, w)$

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



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  $V(c, w)$

<p>Axioms Abstract invariants Concrete invariants Concrete guard <math>\vdash</math> Modified Var. <math>&lt;</math> Var.</p>	<p><math>A(c)</math> <math>I(c, v)</math> <math>J(c, v, w)</math> <math>H(c, w)</math> <math>\vdash</math> <math>V(c, F(c, w)) &lt; V(c, w)</math></p>	<p>VAR</p>
---	--	------------

**variant\_1:**  $2 * a + b$

- **Weighted sum** of  $a$  and  $b$

- |   |               |
|---|---------------|
| –ML_out / GRD <b>done</b>                 | –IL_in / NAT  |
| –ML_in / GRD <b>done</b>                  | –IL_out / NAT |
| –ML_out / <b>inv1_4</b> / INV <b>done</b> | –IL_in / VAR  |
| –ML_out / <b>inv1_5</b> / INV <b>done</b> | –IL_out / VAR |
| –ML_in / <b>inv1_4</b> / INV <b>done</b>  |               |
| –ML_in / <b>inv1_5</b> / INV <b>done</b>  |               |
| – <b>inv1_4</b> / INV <b>done</b>         |               |
| – <b>inv1_5</b> / INV <b>done</b>         |               |
| –IL_in / <b>inv1_4</b> / INV <b>done</b>  |               |
| –IL_in / <b>inv1_5</b> / INV <b>done</b>  |               |
| –IL_out / <b>inv1_4</b> / INV <b>done</b> |               |
| –IL_out / <b>inv1_5</b> / INV <b>done</b> |               |

axm0\_1  
axm0\_2  
inv0\_1  
inv0\_2  
inv1\_1  
inv1\_2  
inv1\_3  
inv1\_4  
inv1\_5

Concrete guard of IL\_in

⊢

Modified variant < Variant

$d \in \mathbb{N}$   
 $0 < d$   
 $n \in \mathbb{N}$   
 $n \leq d$   
 $a \in \mathbb{N}$   
 $b \in \mathbb{N}$   
 $c \in \mathbb{N}$   
 $a + b + c = n$   
 $a = 0 \vee c = 0$   
 $0 < a$

⊢

$2 * (a - 1) + b + 1 < 2 * a + b$

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

⊢  
Modified variant < Variant

$d \in \mathbb{N}$   
 $0 < d$   
 $n \in \mathbb{N}$   
 $n \leq d$   
 $a \in \mathbb{N}$   
 $b \in \mathbb{N}$   
 $c \in \mathbb{N}$   
 $a + b + c = n$   
 $a = 0 \vee c = 0$   
 $0 < b$   
 $a = 0$   
⊢  
 $2 * a + b - 1 < 2 * a + b$

IL\_out / VAR

IL\_out  
**when**  
   $0 < b$   
   $a = 0$   
**then**  
   $b := b - 1$   
   $c := c + 1$   
**end**

---

There are **no new deadlocks in the concrete model**, that is, all deadlocks of the concrete model are already present in the abstract model.

Proof obligation requires that **whenever some abstract event is enabled then so is some concrete event**.

This proof obligation is **optional** (depending on system under study).

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

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

If some abstract guard is true then so is some concrete guard:

$\begin{array}{l} A(c) \\ I(c, v) \\ J(c, v, w) \\ \textcolor{red}{G_1(c, v)} \vee \dots \vee \textcolor{red}{G_m(c, v)} \\ \vdash \\ \textcolor{red}{H_1(c, w)} \vee \dots \vee \textcolor{red}{H_n(c, w)} \end{array}$	DLF
--	-----

axm0\_1  
axm0\_2

inv0\_1

inv0\_2

inv1\_1

inv1\_2

inv1\_3

inv1\_4

inv1\_5

Disjunction of abstract guards

⊢

Disjunction of concrete guards

$$\begin{aligned} & d \in \mathbb{N} \\ & 0 < d \\ & n \in \mathbb{N} \\ & n \leq d \\ & a \in \mathbb{N} \\ & b \in \mathbb{N} \\ & c \in \mathbb{N} \\ & a + b + c = n \\ & a = 0 \vee c = 0 \\ & 0 < n \vee n < d \end{aligned}$$

⊢

$$\begin{aligned} & (a + b < d \wedge c = 0) \vee \\ & c > 0 \vee a > 0 \\ & (b > 0 \wedge a = 0) \end{aligned}$$

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**



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

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

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

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

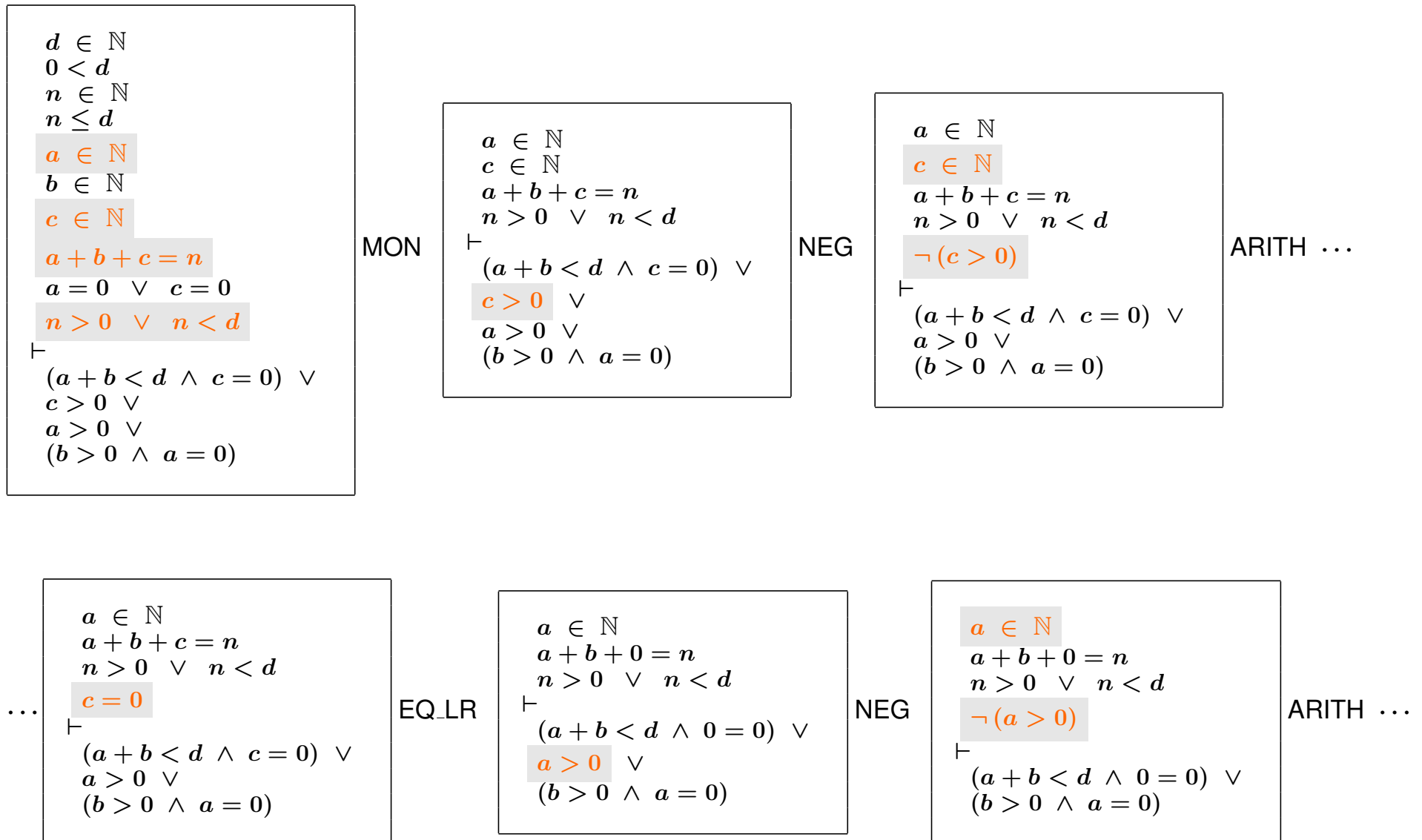
MON

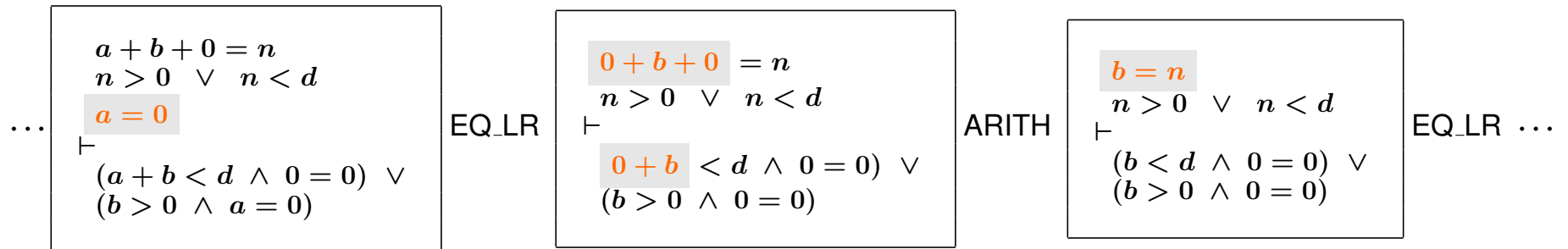
$$\begin{array}{l}
 a \in \mathbb{N} \\
 c \in \mathbb{N} \\
 a + b + c = n \\
 n > 0 \vee n < d \\
 \vdash \\
 (a + b < d \wedge c = 0) \vee \\
 c > 0 \vee \\
 a > 0 \vee \\
 (b > 0 \wedge a = 0)
 \end{array}$$

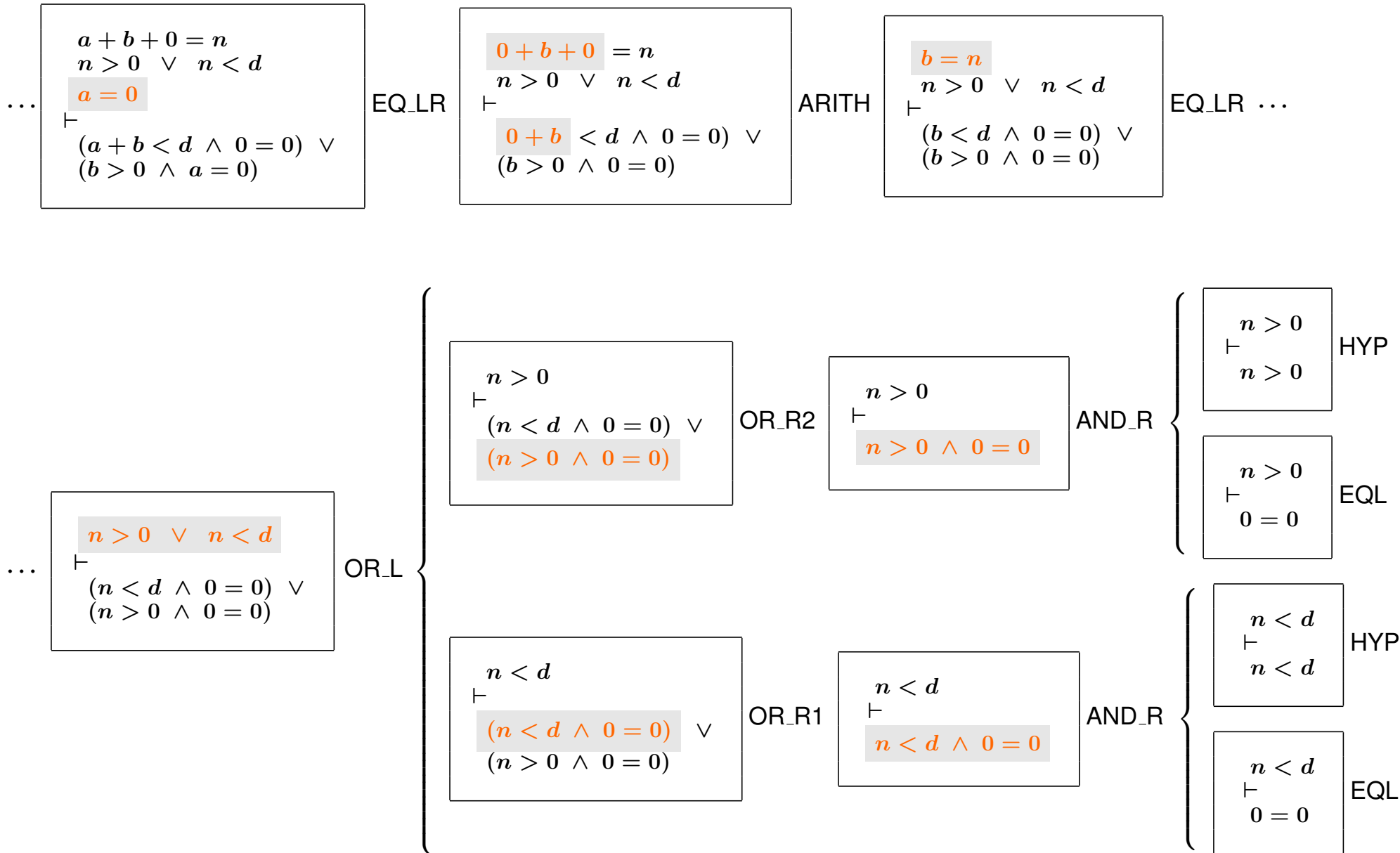
NEG

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

ARITH ...







- |   |                           |
|---|---------------------------|
| –ML_out / GRD <b>done</b>                 | –IL_in / NAT <b>done</b>  |
| –ML_in / GRD <b>done</b>                  | –IL_out / NAT <b>done</b> |
| –ML_out / <b>inv1_4</b> / INV <b>done</b> | –IL_in / VAR <b>done</b>  |
| –ML_out / <b>inv1_5</b> / INV <b>done</b> | –IL_out / VAR <b>done</b> |
| –ML_in / <b>inv1_4</b> / INV <b>done</b>  | –DLF <b>done</b>          |
| –ML_in / <b>inv1_5</b> / INV <b>done</b>  |                           |
| – <b>inv1_4</b> / INV <b>done</b>         |                           |
| – <b>inv1_5</b> / INV <b>done</b>         |                           |
| –IL_in / <b>inv1_4</b> / INV <b>done</b>  |                           |
| –IL_in / <b>inv1_5</b> / INV <b>done</b>  |                           |
| –IL_out / <b>inv1_4</b> / INV <b>done</b> |                           |
| –IL_out / <b>inv1_5</b> / INV <b>done</b> |                           |

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

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
---	-----



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
--	-----

**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:**  $a = 0 \vee c = 0$

**variant1:**  $2 * a + b$

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**

$b := b - 1$

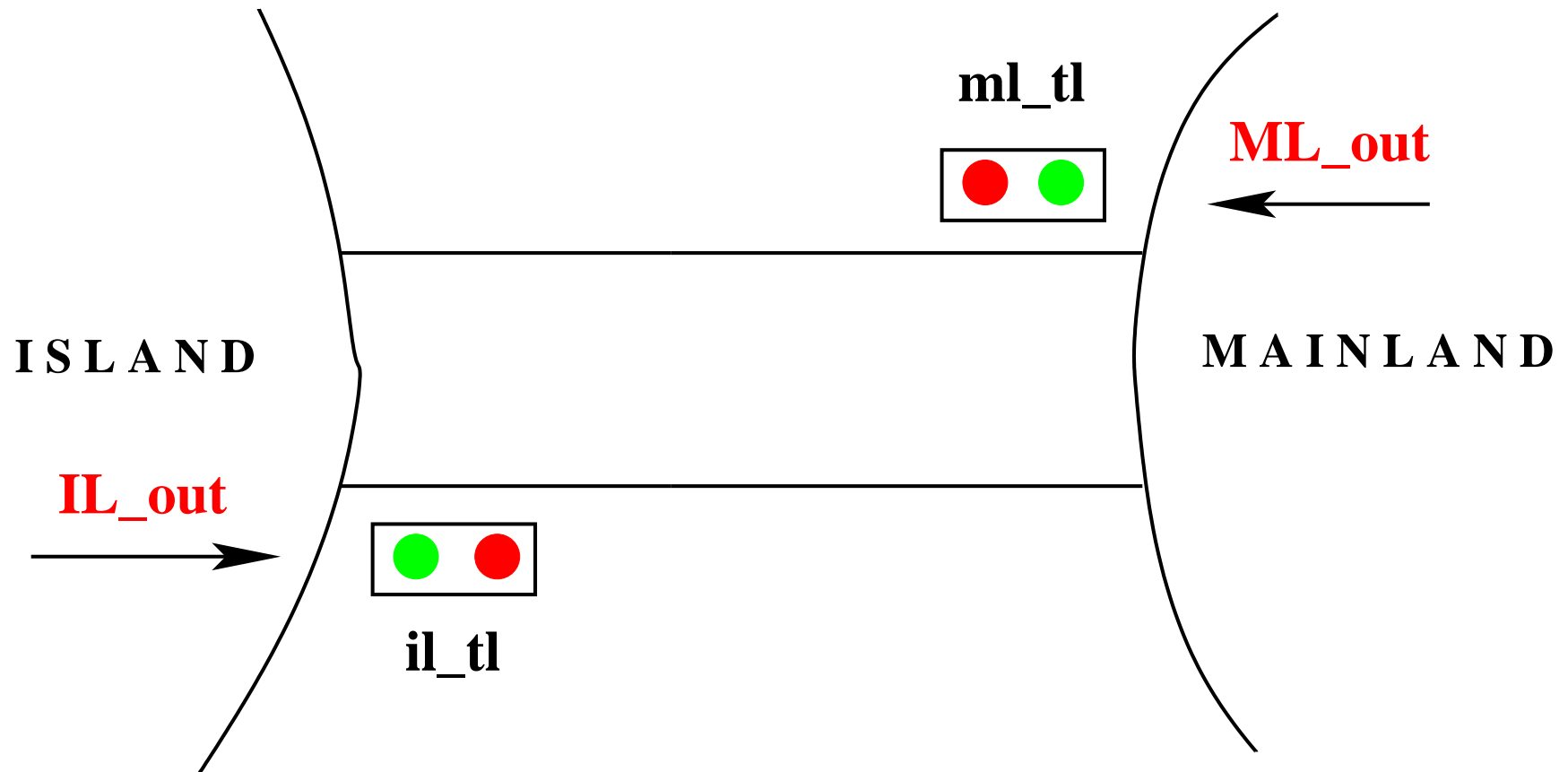
$c := c + 1$

**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

161



**set:** *COLOR*

**constants:** *red, green*

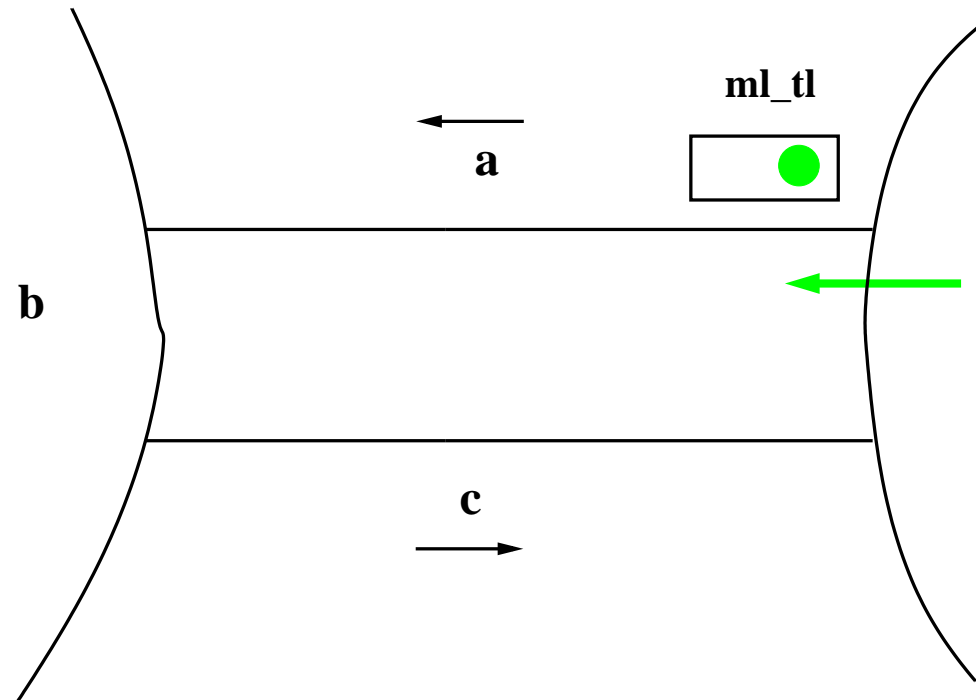
**axm2\_1:** *COLOR = {green, red}*

**axm2\_2:** *green  $\neq$  red*

$$il\_tl \in COLOR$$

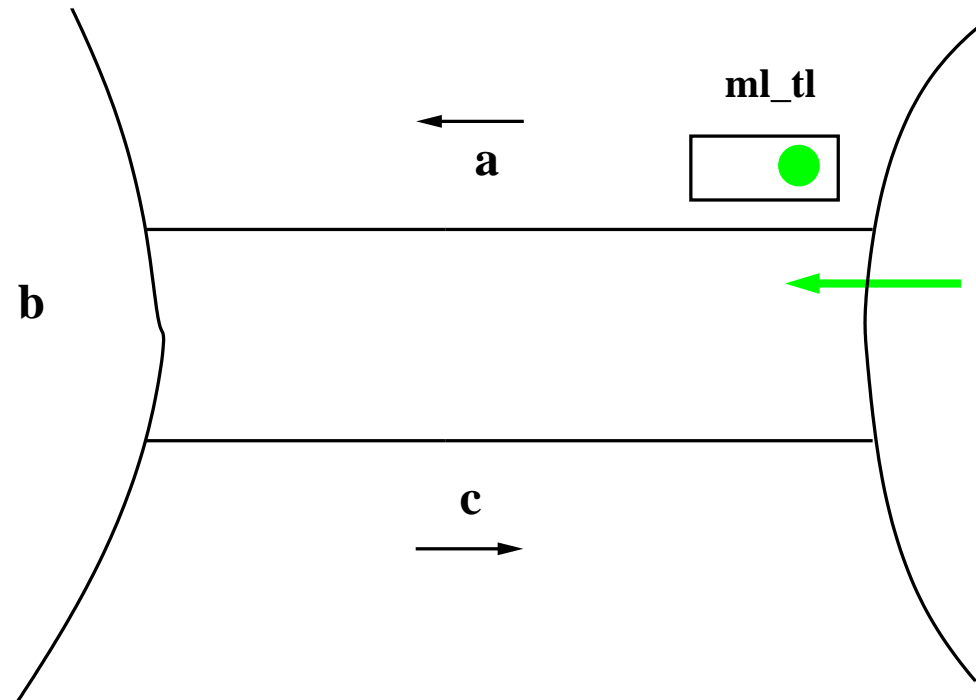
$$ml\_tl \in COLOR$$

Remark: Events **IL\_in** and **ML\_in** are **not modified** in this refinement



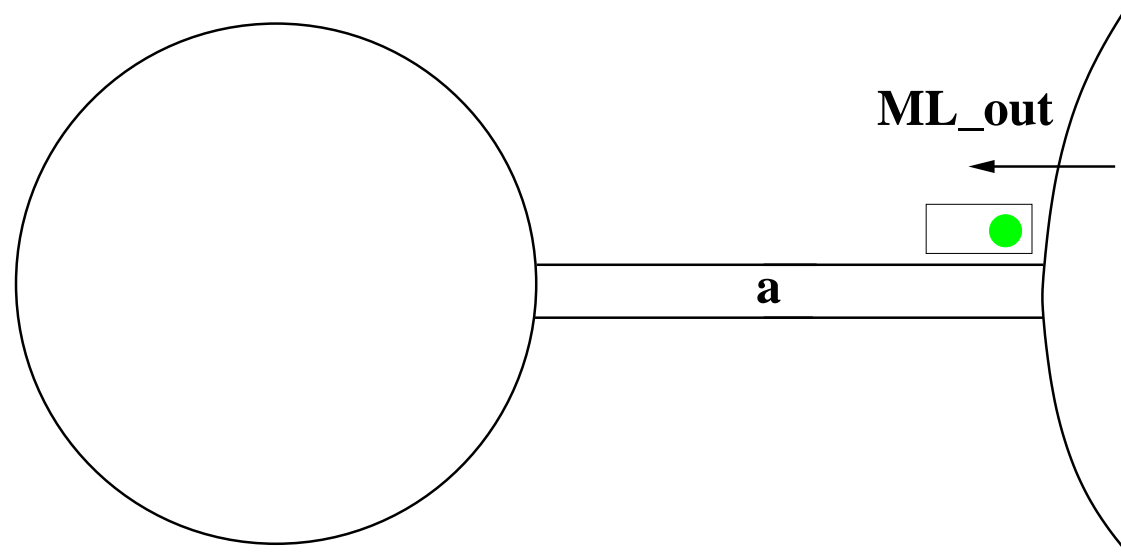
- A green **mainland traffic light** implies **safe access** to the bridge



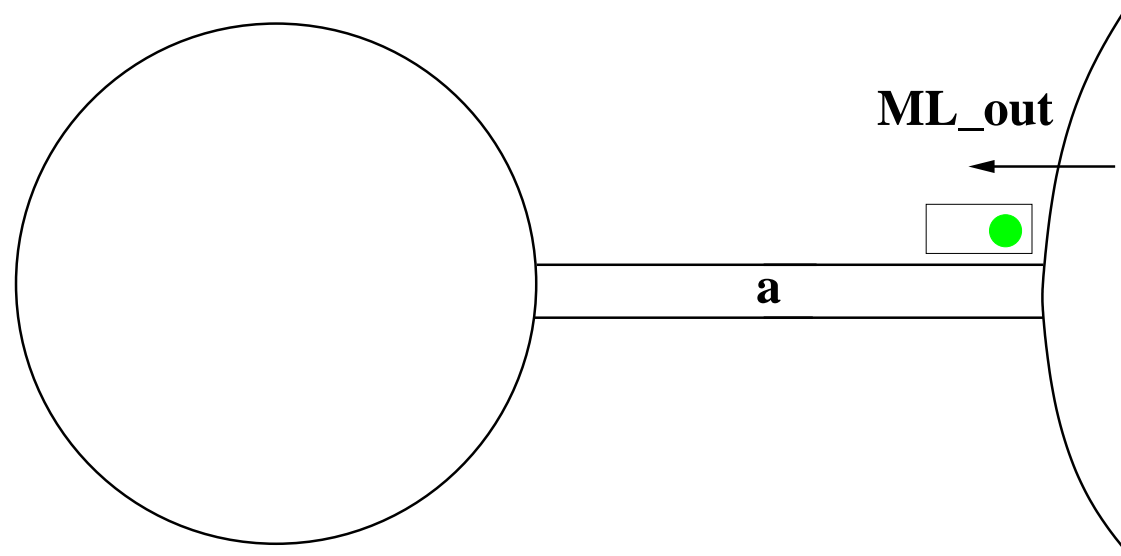


- A green **mainland traffic light** implies **safe access** to the bridge

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

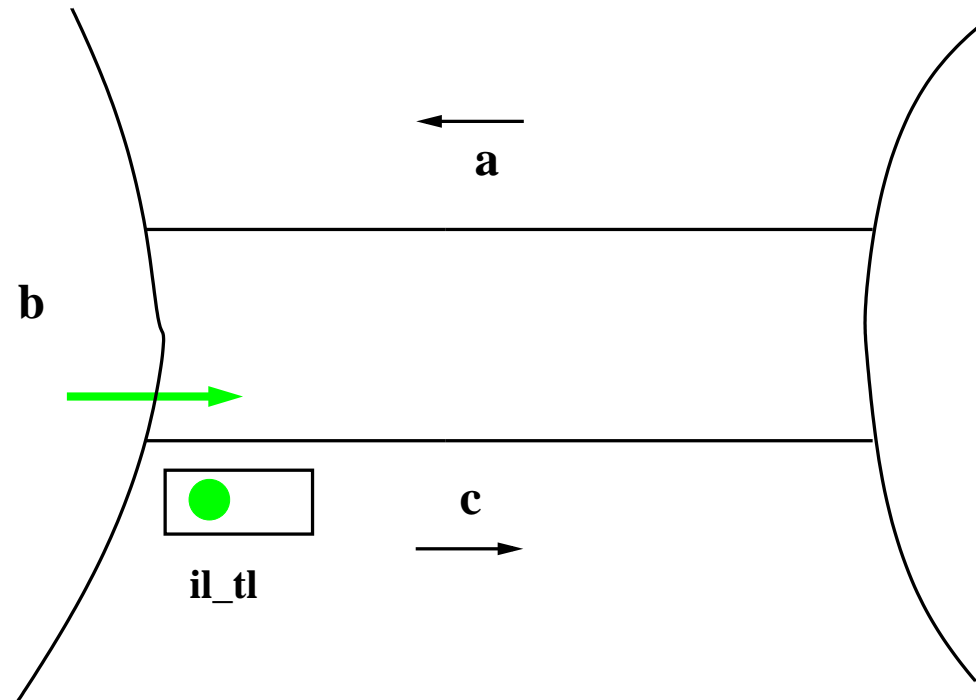


```
(abstract_)ML_out  
when  
   $c = 0$   
   $a + b < d$   
then  
   $a := a + 1$   
end
```

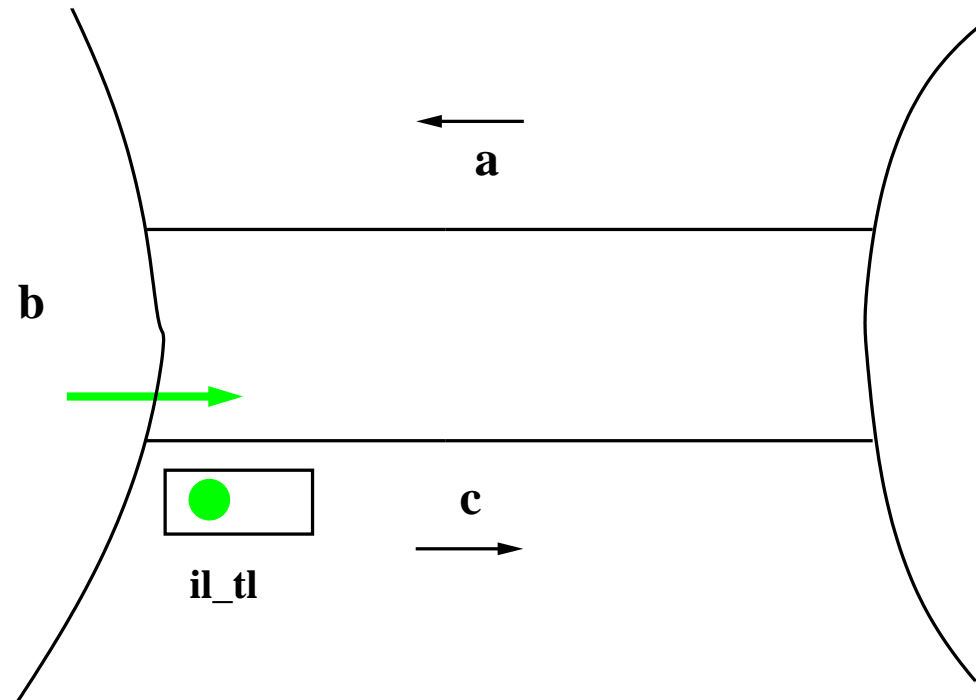


```
(abstract_)ML_out  
  when  
     $c = 0$   
     $a + b < d$   
  then  
     $a := a + 1$   
  end
```

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

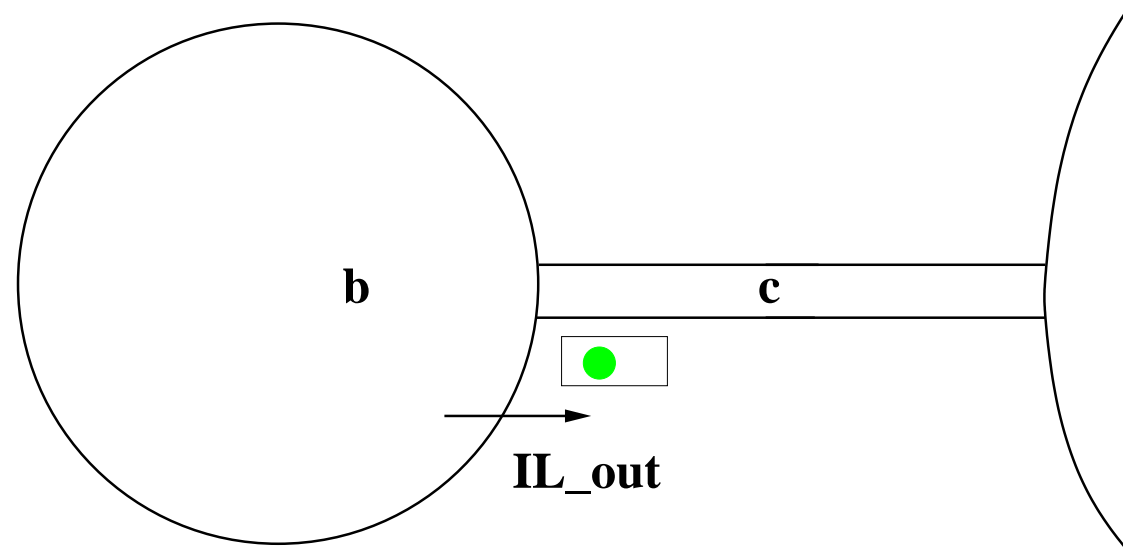


- A green **island traffic light** implies **safe access** to the bridge

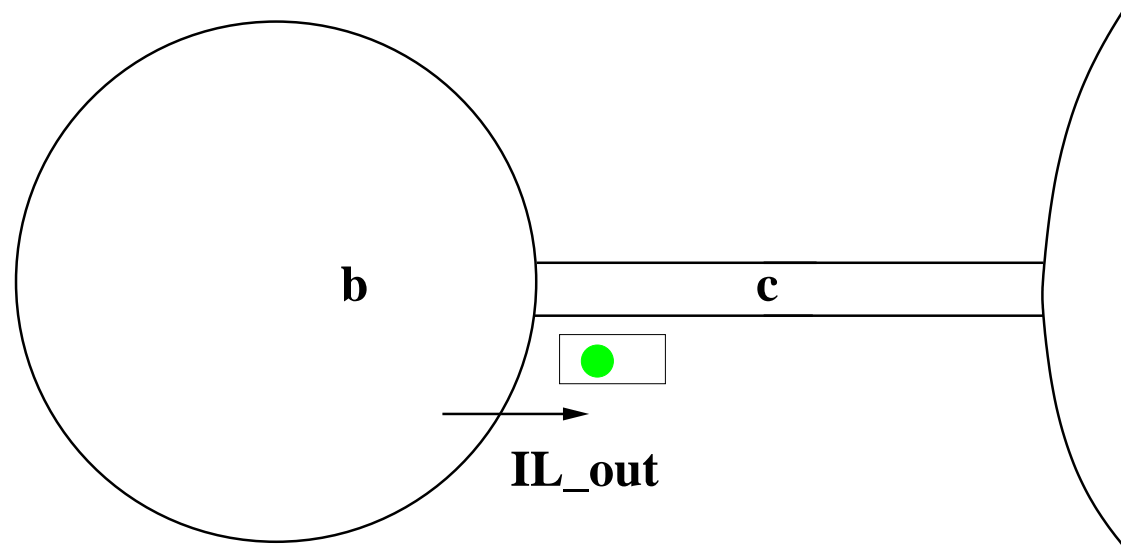


- A green **island traffic light** implies **safe access** to the bridge

$$il\_tl = \text{green} \Rightarrow a = 0 \wedge 0 < b$$



```
(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  
   $il\_tl = \text{green}$   
then  
   $b, c := b - 1, c + 1$   
end
```

ML\_tl\_green

**when**

$ml\_tl = \text{red}$

$c = 0$

$a + b < d$

**then**

$ml\_tl := \text{green}$

**end**

IL\_tl\_green

**when**

$il\_tl = \text{red}$

$a = 0$

$0 < b$

**then**

$il\_tl := \text{green}$

**end**

- Turning lights to **green** when **proper conditions hold**



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

**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:**  $il\_tl = \text{green} \Rightarrow 0 < b \wedge a = 0$

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

```
IL_out
  when
     $il\_tl = \text{green}$ 
  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

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

**inv2\_4:**  $il\_tl = \text{green} \Rightarrow 0 < b \wedge a = 0$

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

```
(abstract-)IL_out
when
   $a = 0$ 
   $0 < b$ 
then
   $b := b - 1$ 
   $c := c + 1$ 
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
```

```
(concrete-)ML_out
when
   $ml\_tl = \text{green}$ 
then
   $a := a + 1$ 
end
```

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

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

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

- SIM is completely trivial since the actions are the same

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

```
(abstract-)IL_out
when
   $a = 0$ 
   $0 < b$ 
then
   $b := b - 1$ 
   $c := c + 1$ 
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
```

```
(concrete-)ML_out
when
   $ml\_tl = \text{green}$ 
then
   $a := a + 1$ 
end
```

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

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

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

- GRD is also trivial

**inv2\_3:**  $ml\_tl = \text{green} \Rightarrow a + b < d \wedge c = 0$

**inv2\_4:**  $il\_tl = \text{green} \Rightarrow 0 < b \wedge a = 0$

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

```
(abstract-)IL_out  
when  
   $a = 0$   
   $0 < b$   
then  
   $b := b - 1$   
   $c := c + 1$   
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
```

```
(concrete-)ML_out  
when  
   $ml\_tl = \text{green}$   
then  
   $a := a + 1$   
end
```

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

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

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

- INV applied to ML\_in and IL\_in holds trivially

**inv2\_3:**  $ml\_tl = \text{green} \Rightarrow a + b < d \wedge c = 0$

**inv2\_4:**  $il\_tl = \text{green} \Rightarrow 0 < b \wedge a = 0$



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

```
(abstract-)IL_out  
when  
   $a = 0$   
   $0 < b$   
then  
   $b := b - 1$   
   $c := c + 1$   
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
```

```
(concrete-)ML_out  
when  
   $ml\_tl = \text{green}$   
then  
   $a := a + 1$   
end
```

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

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

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

- INV applied to ML\_out and IL\_out **raise some difficulties**

- ML\_out / **inv2\_4** / INV

- IL\_out / **inv2\_3** / INV

- ML\_out / **inv2\_3** / INV

- IL\_out / **inv2\_4** / INV

- Rules about **implication**

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

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

- Rules about **negation**

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

$$\frac{H, P \vdash Q \quad H, P \vdash \neg Q}{H \vdash \neg P} \quad \text{NOT\_R}$$

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**

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

ML\_out / **inv2\_4** / INV

ML\_out  
 when  
      $ml\_tl = \text{green}$   
 then  
      $\textcolor{red}{a} := \textcolor{red}{a} + 1$   
 end

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

MON

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

IMP\_R ...

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

MON

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

IMP\_R ...

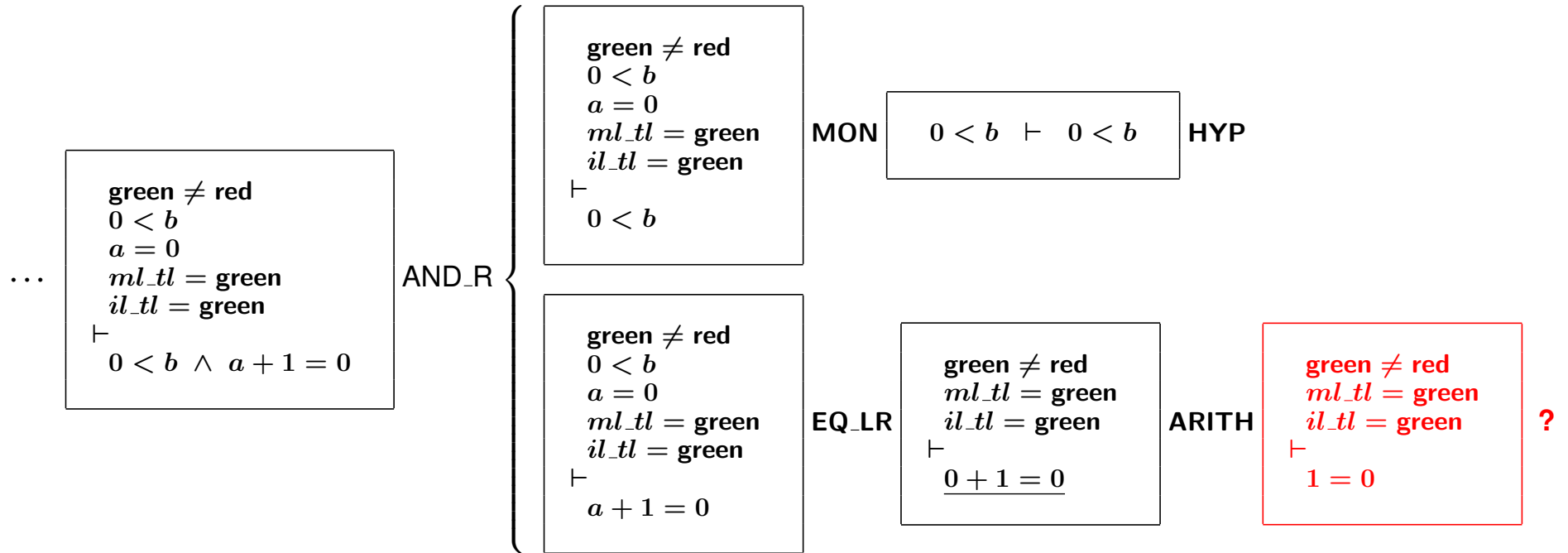
...

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

IMP\_L

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

AND\_L ...



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 IL\_out  
 $\vdash$   
 Modified inv2\_3

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

IL\_out / inv2\_3 / INV

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



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

MON

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

IMP\_R . .

$$\begin{aligned}
 & d \in \mathbb{N} \\
 & 0 < d \\
 & COLOR = \{\text{green}, \text{red}\} \\
 & \text{green} \neq \text{red} \\
 & n \in \mathbb{N} \\
 & n \leq d \\
 & a \in \mathbb{N} \\
 & b \in \mathbb{N} \\
 & c \in \mathbb{N} \\
 & a + b + c = n \\
 & a = 0 \vee c = 0 \\
 & ml\_tl \in COLOR \\
 & il\_tl \in COLOR \\
 & ml\_tl = \text{green} \Rightarrow a + \textcolor{red}{b} < d \wedge \textcolor{red}{c} = 0 \\
 & il\_tl = \text{green} \Rightarrow 0 < b \wedge a = 0 \\
 & \textcolor{blue}{il\_tl} = \text{green}
 \end{aligned}$$

$$\begin{aligned}
 & \vdash \\
 & ml\_tl = \text{green} \Rightarrow a + \textcolor{red}{b} - 1 < d \wedge \textcolor{red}{c} + 1 = 0
 \end{aligned}$$

MON

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

IMP\_R . .

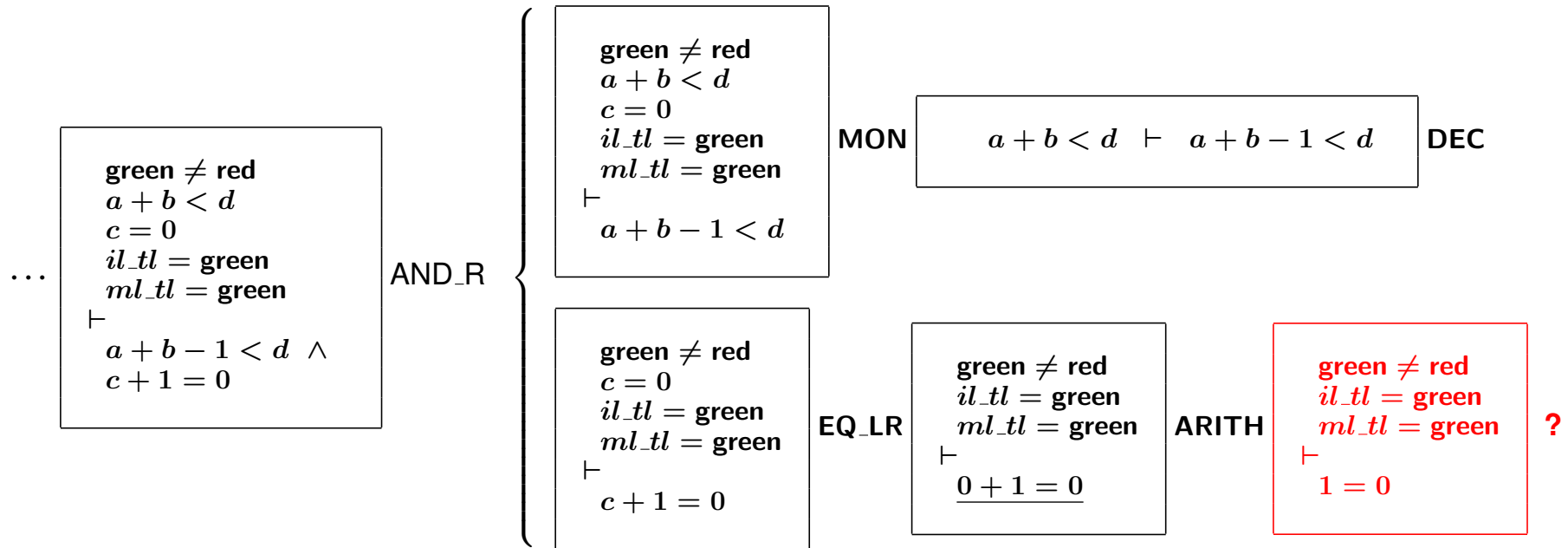
...

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

IMP\_L

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

AND\_L ...



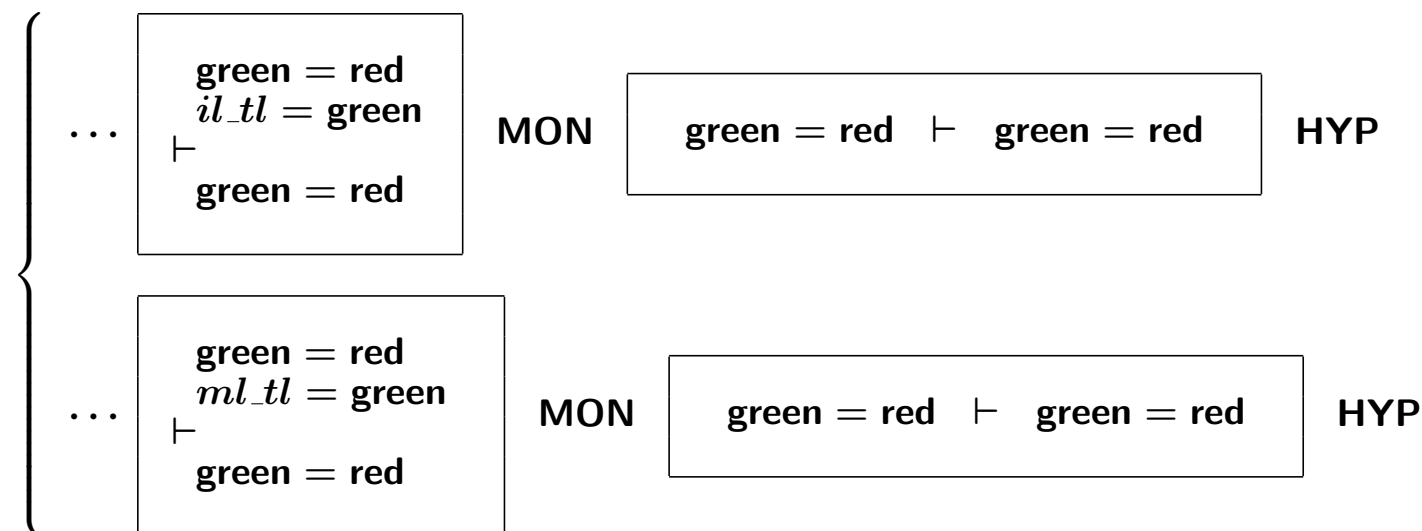
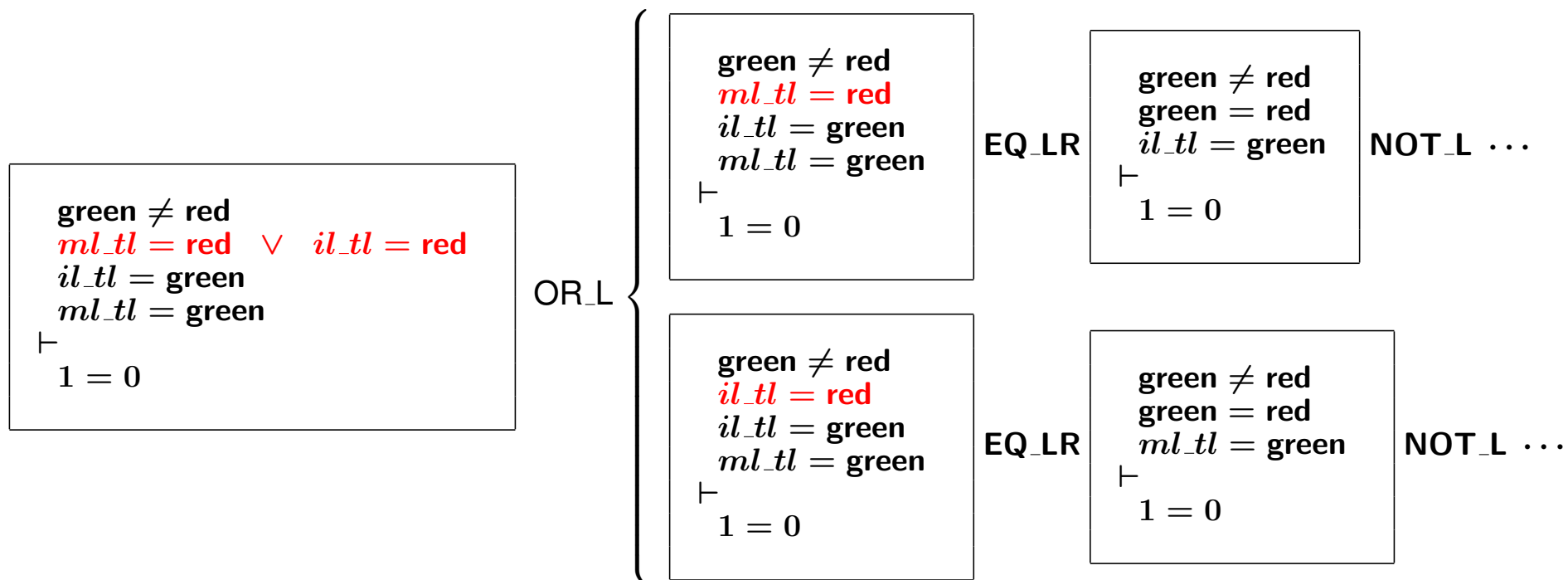
- In both cases, we were stopped by attempting to prove the following

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

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

$$\text{inv2\_5: } ml\_tl = \text{red} \vee il\_tl = \text{red}$$



**inv2\_5:**     $ml\_tl = \text{red} \vee il\_tl = \text{red}$

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

- 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

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 ML\_out  
 $\vdash$   
 Modified inv2\_3

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

ML\_out / inv2\_3 / INV

ML\_out  
 $\textcolor{blue}{when}$   
 $\quad \textcolor{blue}{ml\_tl} = \text{green}$   
 $\textcolor{blue}{then}$   
 $\quad \textcolor{red}{a} := \textcolor{red}{a} + \textcolor{red}{1}$   
 $\textcolor{blue}{end}$



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

MON

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

IMP\_R...

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

MON

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

IMP\_R ...

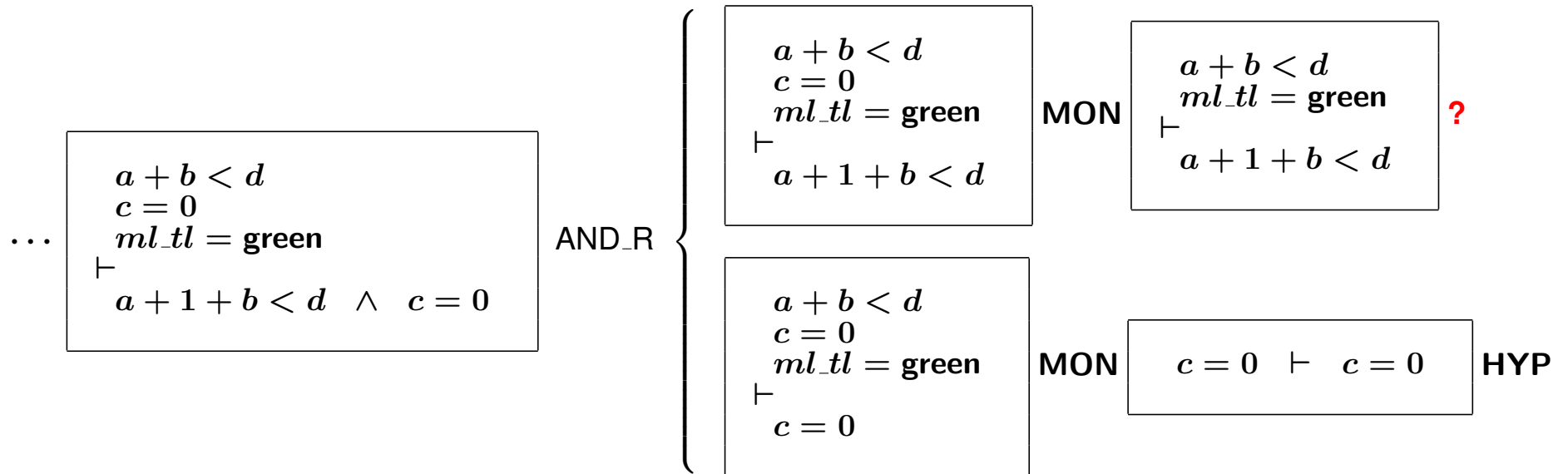
...

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

IMP\_L

$$\begin{aligned}
 & a + b < d \wedge c = 0 \\
 & ml\_tl = \text{green} \\
 & \vdash \\
 & a + 1 + b < d \wedge c = 0
 \end{aligned}$$

AND\_L ...



- This requires splitting the ML\_out in two separate events ML\_out\_1 and ML\_out\_2

ML\_out\_1

when

$ml\_tl = \text{green}$

$a + 1 + b < d$

then

$a := a + 1$

end

ML\_out\_2

when

$ml\_tl = \text{green}$

$a + 1 + b = d$

then

$a := a + 1$

$ml\_tl := \text{red}$

end

ML\_out\_1

**when**

$ml\_tl = \text{green}$

$a + 1 + b < d$

**then**

$a := a + 1$

**end**

ML\_out\_2

**when**

$ml\_tl = \text{green}$

$a + 1 + b = d$

**then**

$a := a + 1$

$ml\_tl := \text{red}$

**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  
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 ML\_out\_1

⊢  
Modified inv2\_3

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

ML\_out\_1 / inv2\_3 / INV

ML\_out\_1  
when  
     $ml\_tl = \text{green}$   
     $a + 1 + b < d$   
then  
     $a := a + 1$   
end

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

MON

$$\begin{aligned}
 & ml\_tl = \text{green} \Rightarrow a + b < d \wedge c = 0 \\
 & \textcolor{red}{a + 1 + b} < d \\
 & \vdash \\
 & ml\_tl = \text{green} \Rightarrow a + 1 + b < d \wedge c = 0
 \end{aligned}$$

IMP\_R...

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

MON

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

IMP\_R...

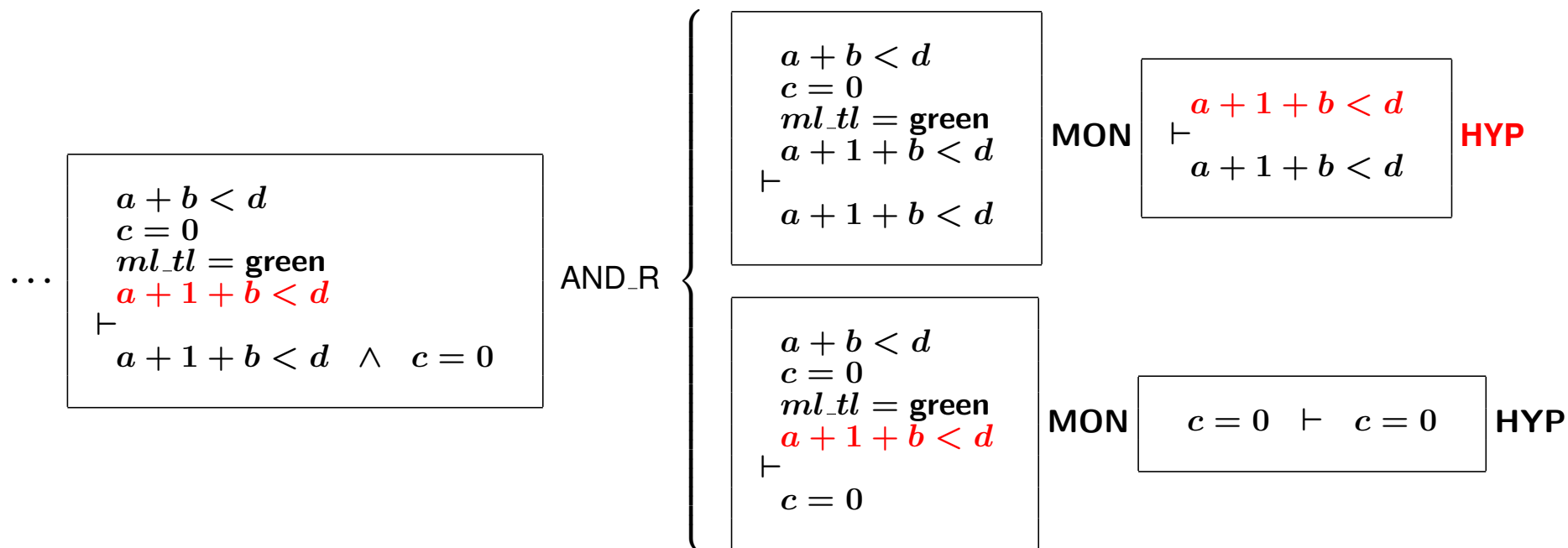
...

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

IMP\_L

$a + b < d \wedge c = 0$   
 $ml\_tl = \text{green}$   
 $a + 1 + b < d$   
 $\vdash$   
 $a + 1 + b < d \wedge c = 0$

AND\_L ...





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 ML\_out\_2

⊢  
Modified inv2\_3

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

ML\_out\_2 / inv2\_3 / INV

ML\_out\_2  
**when**  
     $ml\_tl = \text{green}$   
     $a + 1 + b = d$   
**then**  
     $a := a + 1$   
     $ml\_tl := \text{red}$   
**end**

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

MON

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

IMP\_R

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

MON

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

IMP\_R

...

$$\begin{aligned}
 & \text{green} \neq \text{red} \\
 & \text{red} = \text{green} \\
 & \vdash \\
 & a + 1 + b < d \wedge c = 0
 \end{aligned}$$

EQ\_LR

$$\begin{aligned}
 & \text{green} \neq \text{green} \\
 & \vdash \\
 & a + 1 + b < d \wedge c = 0
 \end{aligned}$$

NOT\_L

$$\begin{aligned}
 & \vdash \\
 & \text{green} = \text{green}
 \end{aligned}$$

EQL

- 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

$\text{axm0\_1}$   
 $\text{axm0\_2}$   
 $\text{axm2\_1}$   
 $\text{axm2\_2}$   
 $\text{inv0\_1}$   
 $\text{inv0\_2}$   
 $\text{inv1\_1}$   
 $\text{inv1\_2}$   
 $\text{inv1\_3}$   
 $\text{inv1\_4}$   
 $\text{inv1\_5}$   
 $\text{inv2\_1}$   
 $\text{inv2\_2}$   
 $\text{inv2\_3}$   
 $\text{inv2\_4}$   
 Guard of event  $\text{IL\_out}$   
 $\vdash$   
 Modified invariant  $\text{inv2\_4}$

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

$\text{IL\_out} / \text{inv2\_4} / \text{INV}$

$\text{IL\_out}$   
 $\text{when}$   
 $\quad \textcolor{blue}{\text{il\_tl}} = \text{green}$   
 $\text{then}$   
 $\quad \textcolor{red}{b} := \textcolor{red}{b} - 1$   
 $\quad \textcolor{red}{c} := \textcolor{red}{c} + 1$   
 $\text{end}$

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

MON

$$\begin{aligned}
 & il\_tl = \text{green} \Rightarrow 0 < \textcolor{red}{b} \wedge a = 0 \\
 & \textcolor{blue}{il\_tl} = \text{green} \\
 & \vdash \\
 & il\_tl = \text{green} \Rightarrow 0 < \textcolor{red}{b} - 1 \wedge a = 0
 \end{aligned}$$

IMP\_R

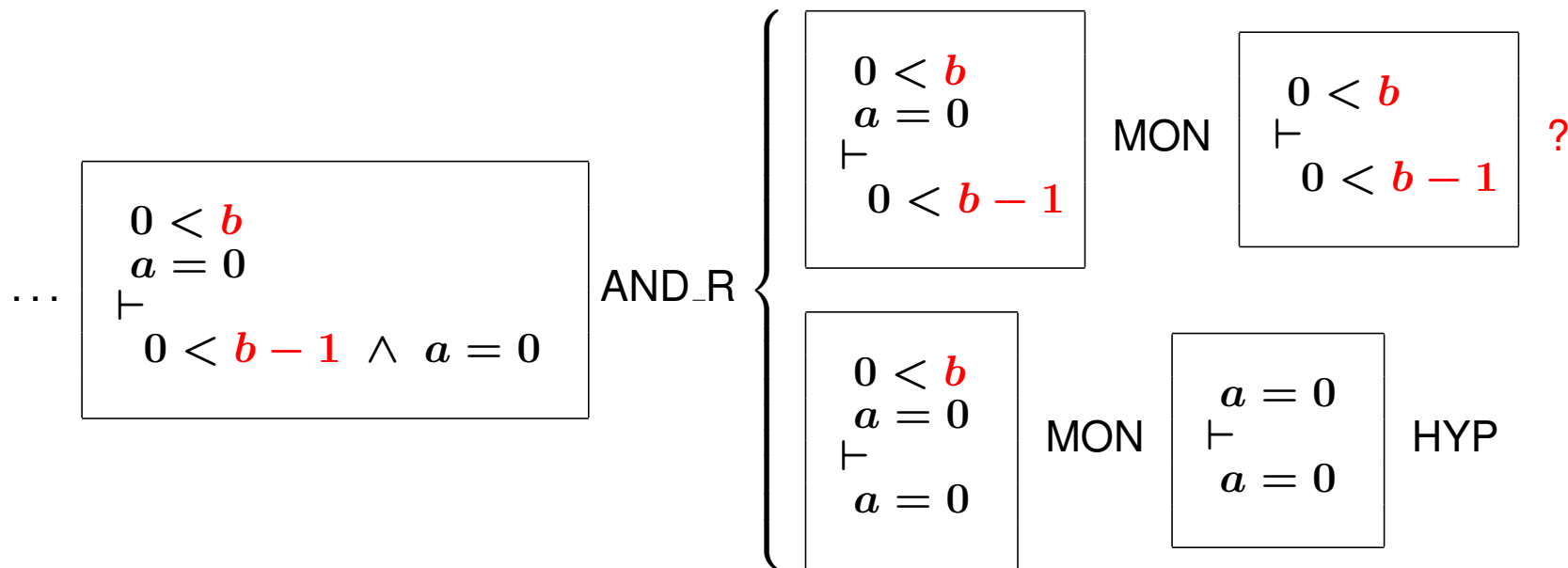
...

$$\begin{aligned}
 & il\_tl = \text{green} \Rightarrow 0 < \textcolor{red}{b} \wedge a = 0 \\
 & \textcolor{blue}{il\_tl} = \text{green} \\
 & \vdash \\
 & 0 < \textcolor{red}{b} - 1 \wedge a = 0
 \end{aligned}$$

IMP\_L

$$\begin{aligned}
 & 0 < \textcolor{red}{b} \wedge a = 0 \\
 & \vdash \\
 & 0 < \textcolor{red}{b} - 1 \wedge a = 0
 \end{aligned}$$

AND\_L



- This requires splitting the concrete IL\_out in **two separate events** IL\_out\_1 and IL\_out\_2

```

IL_out_1
  when
    il_tl = green
    b ≠ 1
  then
    b, c := b - 1, c + 1
  end
    
```

```

IL_out_2
  when
    il_tl = green
    b = 1
  then
    b, c := b - 1, c + 1
    il_tl := red
  end
    
```

```
IL_out_1
  when
    il_tl = green
    b ≠ 1
  then
    b, c := b - 1, c + 1
  end
```

```
IL_out_2
  when
    il_tl = green
    b = 1
  then
    b, c := b - 1, c + 1
    il_tl := 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

⊢

Modified invariant **inv2\_4**

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

IL\_out\_1 / **inv2\_4** / INV

IL\_out\_1  
  **when**  
     $\textcolor{blue}{il\_tl} = \textcolor{blue}{green}$   
     $\textcolor{blue}{b} \neq 1$   
  **then**  
     $\textcolor{red}{b}, \textcolor{red}{c} := \textcolor{red}{b} - 1, \textcolor{red}{c} + 1$   
  **end**

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

MON

$$\begin{aligned}
 & il\_tl = \text{green} \Rightarrow 0 < \textcolor{red}{b} \wedge a = 0 \\
 & \textcolor{blue}{il\_tl = green} \\
 & \textcolor{blue}{b \neq 1} \\
 & \vdash \\
 & il\_tl = \text{green} \Rightarrow 0 < \textcolor{red}{b - 1} \wedge \\
 & \quad a = 0
 \end{aligned}$$

IMP\_R

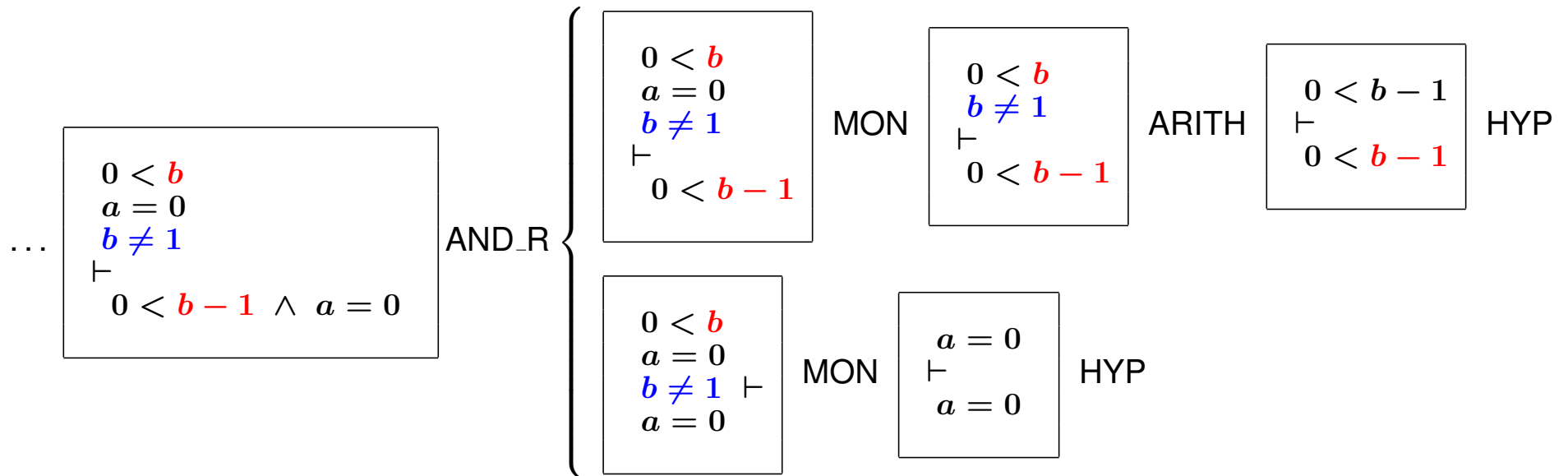
...

$$\begin{aligned}
 & il\_tl = \text{green} \Rightarrow 0 < \textcolor{red}{b} \wedge a = 0 \\
 & \textcolor{blue}{il\_tl = green} \\
 & \textcolor{blue}{b \neq 1} \\
 & \vdash \\
 & 0 < \textcolor{red}{b - 1} \wedge a = 0
 \end{aligned}$$

IMP\_L

$$\begin{aligned}
 & 0 < \textcolor{red}{b} \wedge a = 0 \\
 & \textcolor{blue}{b \neq 1} \\
 & \vdash \\
 & 0 < \textcolor{red}{b - 1} \wedge a = 0
 \end{aligned}$$

AND\_L



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\_2

⊢

Modified invariant **inv2\_4**

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

IL\_out\_1 / **inv2\_4** / INV

IL\_out\_2  
**when**  
     $\textcolor{blue}{il\_tl} = \text{green}$   
     $\textcolor{blue}{b} = 1$   
**then**  
     $\textcolor{red}{b}, \textcolor{red}{c}, \textcolor{red}{il\_tl} := \textcolor{red}{b} - 1, \textcolor{red}{c} + 1, \textcolor{red}{red}$   
**end**

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

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

MON IMP\_R

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

...

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

EQ\_LR NOT\_L

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

EQL

- 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

$$\text{inv2\_5: } ml\_tl = \text{red} \vee il\_tl = \text{red}$$

Unless we correct them as follows:

```
ML_tl_green
  when
    ml_tl = red
    a + b < d
    c = 0
  then
    ml_tl := green
    il_tl := red
  end
```

```
IL_tl_green
  when
    il_tl = red
    0 < b
    a = 0
  then
    il_tl := green
    ml_tl := red
  end
```

- Correct event refinement: OK
- Absence of divergence of new events: FAILURE
- Absence of deadlock: ?



ML\_tl\_green

**when**

$ml\_tl = \text{red}$

$a + b < d$

$c = 0$

**then**

$ml\_tl := \text{green}$

$il\_tl := \text{red}$

**end**

IL\_tl\_green

**when**

$il\_tl = \text{red}$

$0 < b$

$a = 0$

**then**

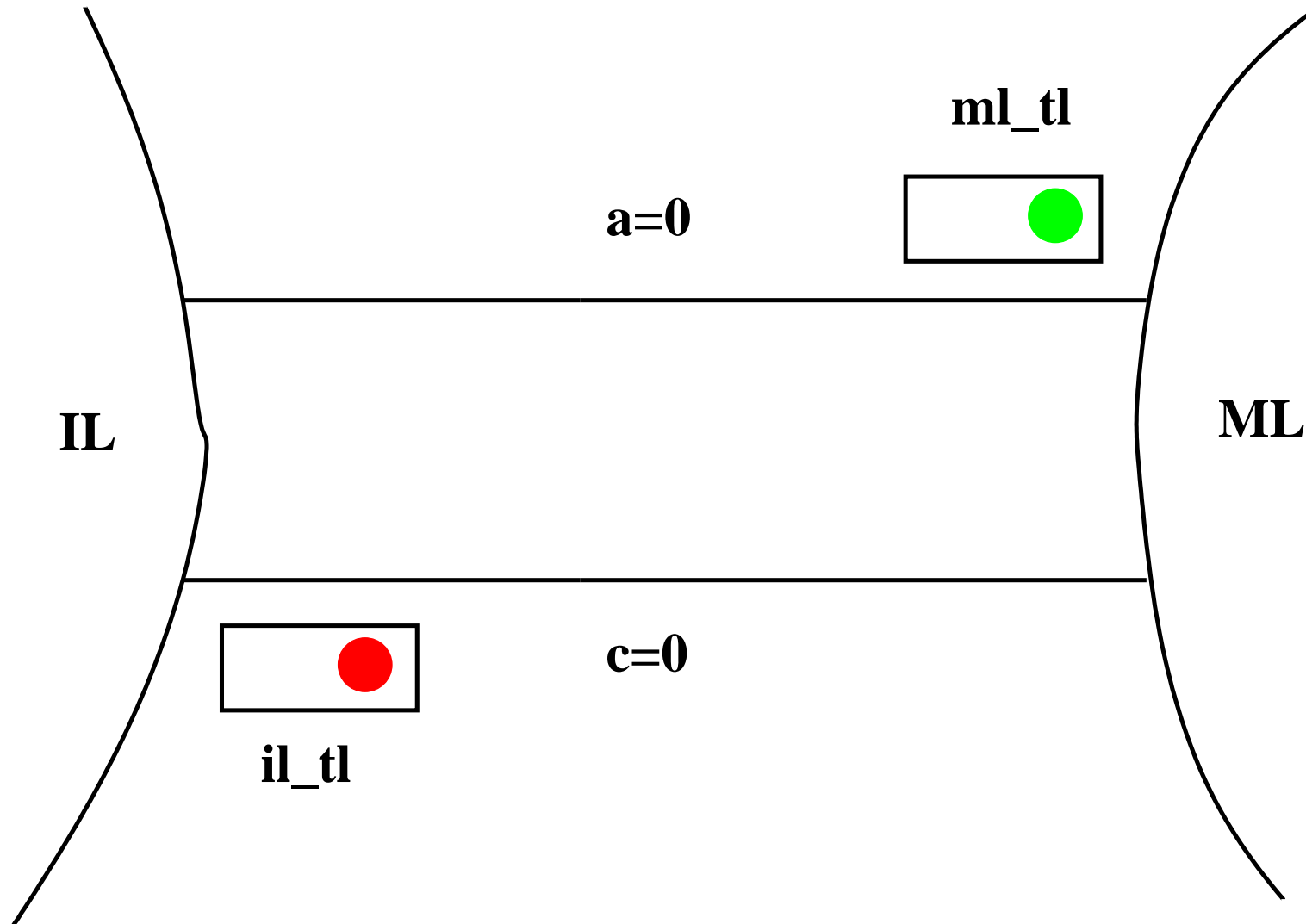
$il\_tl := \text{green}$

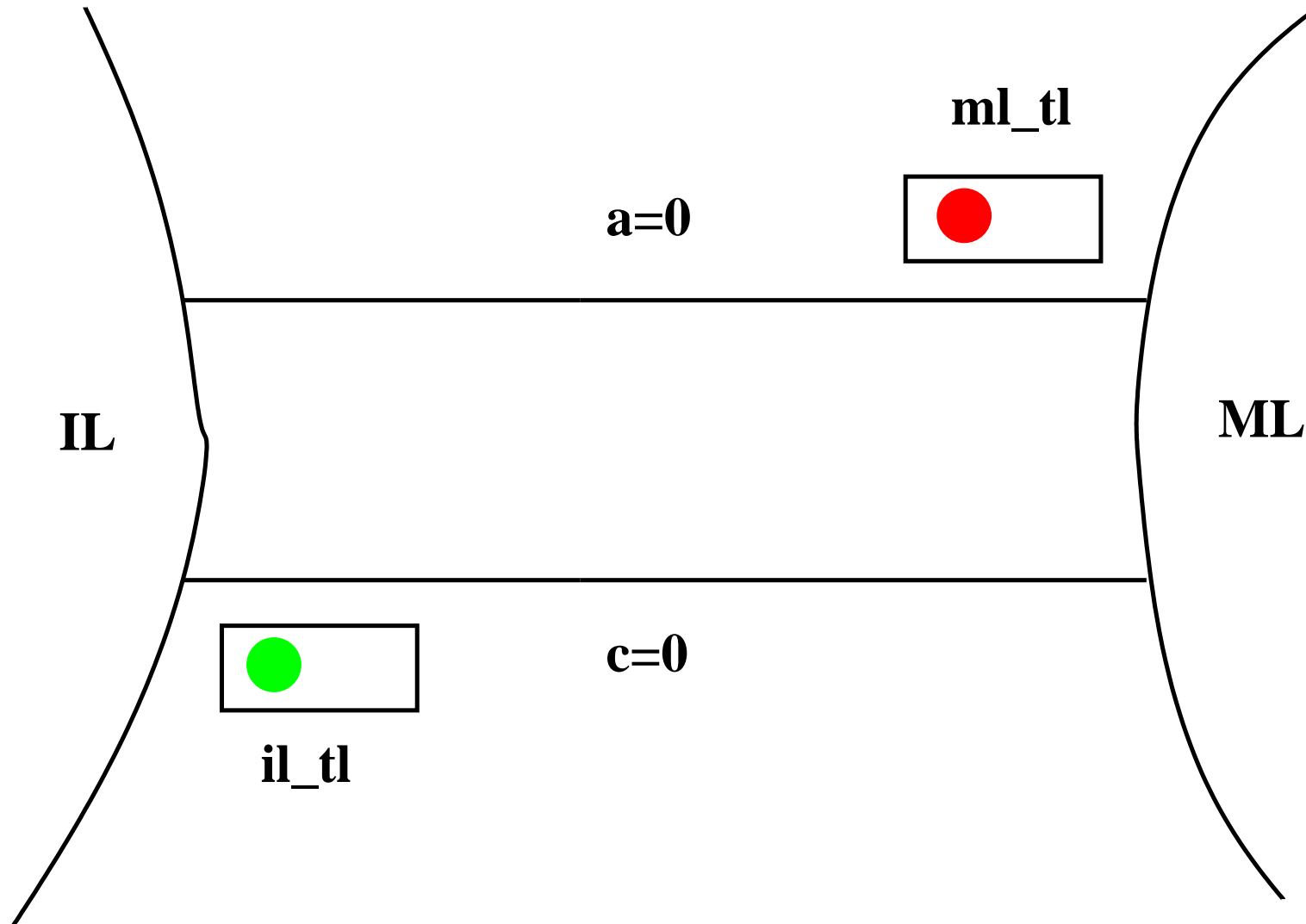
$ml\_tl := \text{red}$

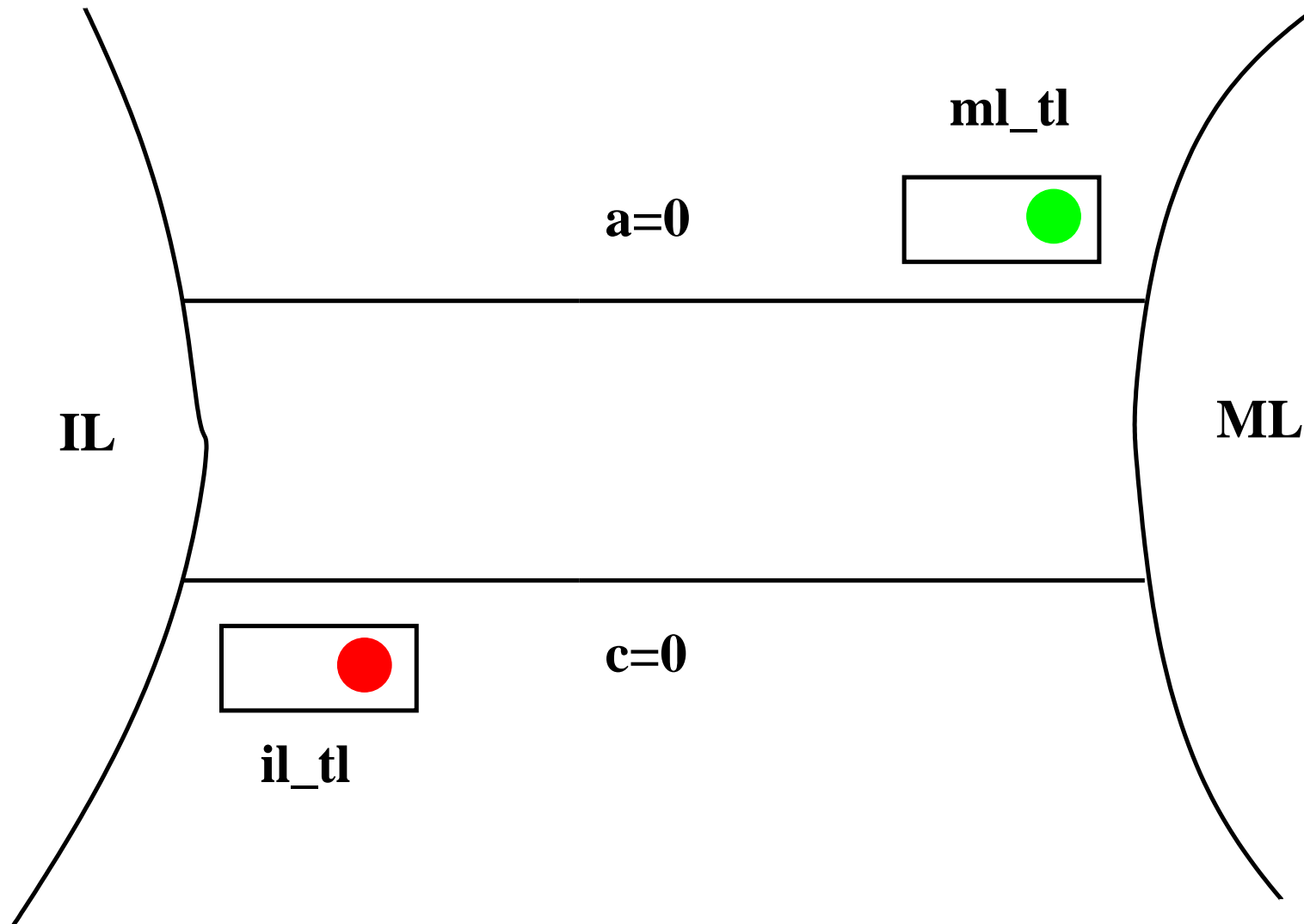
**end**

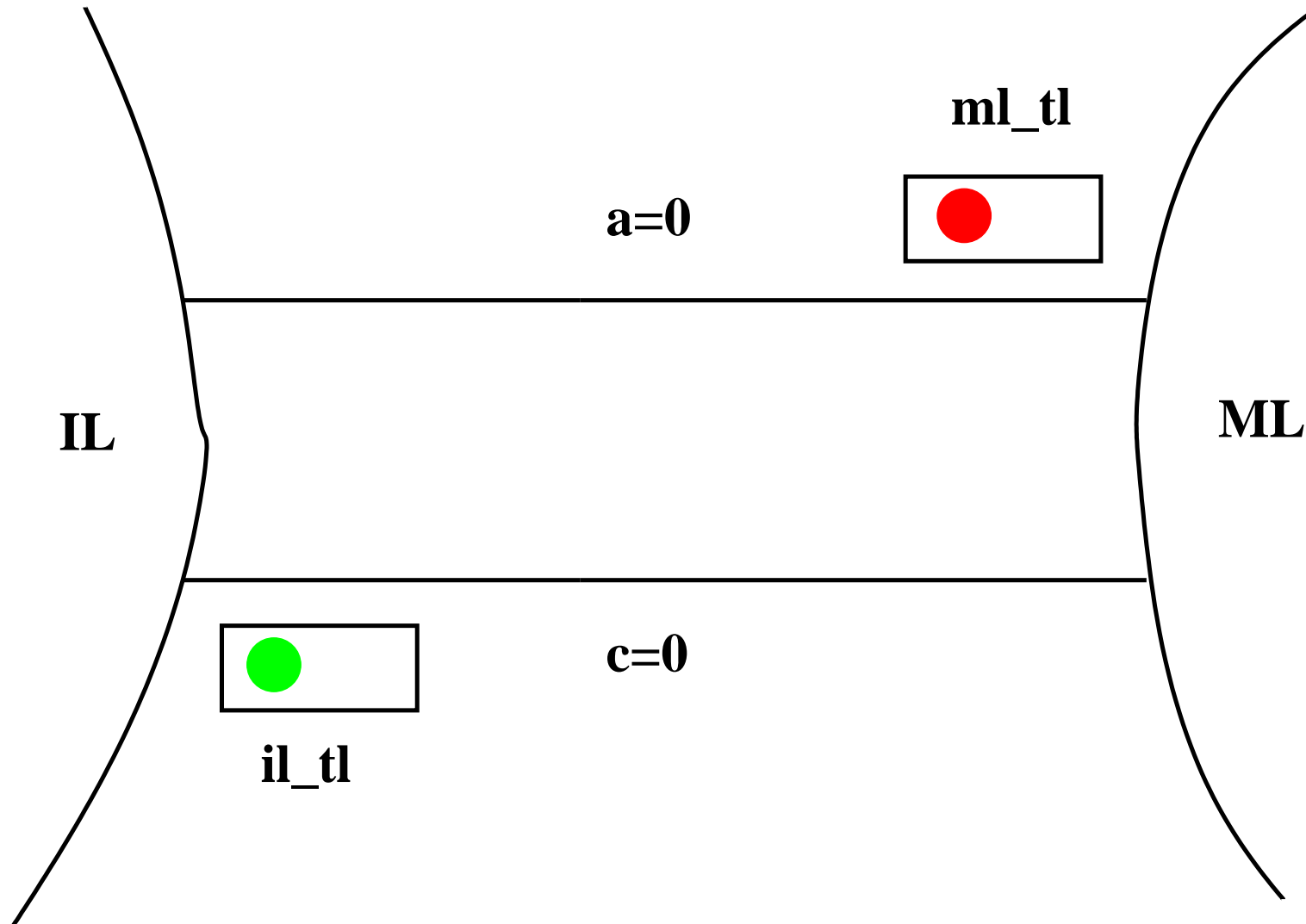
When  $a$  and  $c$  are both equal to 0 and  $b$  is positive, then both events are always alternatively enabled

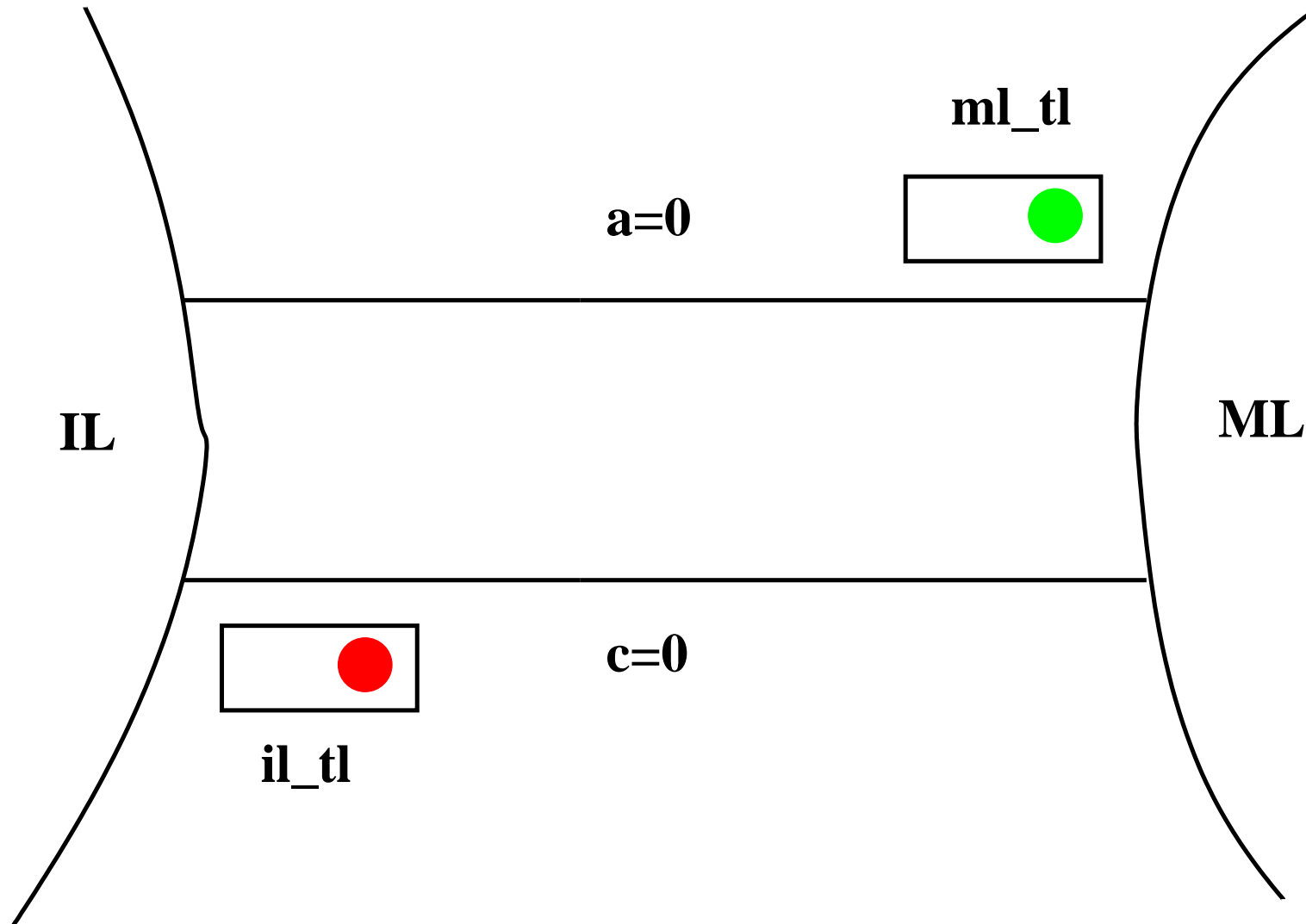
The lights can change colors very rapidly

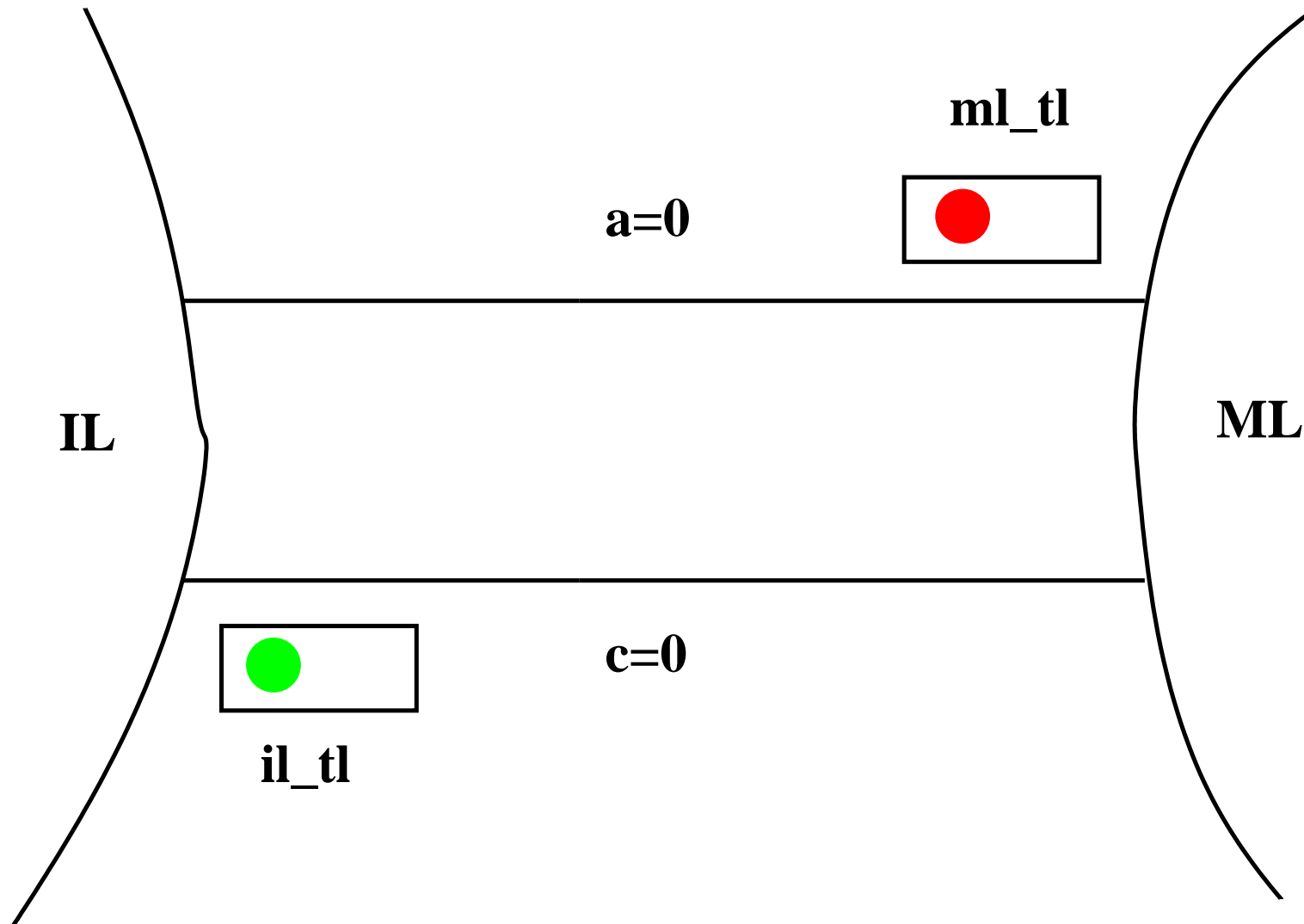












- Allowing each light to turn green only when at least one car has passed in the other direction
- For this, we introduce two additional variables:

**inv2\_6:**  $ml\_pass \in \{0, 1\}$

**inv2\_7:**  $il\_pass \in \{0, 1\}$



ML\_out\_1

**when**

$ml\_tl = \text{green}$

$a + 1 + b < d$

**then**

$a := a + 1$

$ml\_pass := 1$

**end**

ML\_out\_2

**when**

$ml\_tl = \text{green}$

$a + 1 + b = d$

**then**

$a := a + 1$

$ml\_tl := \text{red}$

$ml\_pass := 1$

**end**

```
IL_out_1
  when
    il_tl = green
    b ≠ 1
  then
    b := b - 1
    c := c + 1
    il_pass := 1
  end
```

```
IL_out_2
  when
    il_tl = green
    b = 1
  then
    b := b - 1
    c := c + 1
    il_tl := red
    il_pass := 1
  end
```

ML\_tl\_green

**when**

$ml\_tl = \text{red}$

$a + b < d$

$c = 0$

$il\_pass = 1$

**then**

$ml\_tl := \text{green}$

$il\_tl := \text{red}$

$ml\_pass := 0$

**end**

IL\_tl\_green

**when**

$il\_tl = \text{red}$

$0 < b$

$a = 0$

$ml\_pass = 1$

**then**

$il\_tl := \text{green}$

$ml\_tl := \text{red}$

$il\_pass := 0$

**end**

We exhibit the following variant

**variant\_2:**  $ml\_pass + il\_pass$

$$\begin{aligned} & ml\_tl = \text{red} \\ & a + b < d \\ & c = 0 \\ & il\_pass = 1 \\ \Rightarrow & \\ & il\_pass + 0 < \\ & ml\_pass + il\_pass \end{aligned}$$

$$\begin{aligned} & il\_tl = \text{red} \\ & b > 0 \\ & a = 0 \\ & ml\_pass = 1 \\ \Rightarrow & \\ & ml\_pass + 0 < \\ & ml\_pass + il\_pass \end{aligned}$$

This cannot be proved. This suggests the following invariants:

$$\text{inv2\_8: } ml\_tl = \text{red} \Rightarrow ml\_pass = 1$$

$$\text{inv2\_9: } il\_tl = \text{red} \Rightarrow il\_pass = 1$$

$$0 < d$$

$$ml\_tl \in \{\text{red}, \text{green}\}$$

$$il\_tl \in \{\text{red}, \text{green}\}$$

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

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

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

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

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

$$ml\_tl = \text{red} \Rightarrow ml\_pass = 1$$

$$il\_tl = \text{red} \Rightarrow il\_pass = 1$$

$\Rightarrow$

$$(ml\_tl = \text{red} \wedge a + b < d \wedge c = 0 \wedge il\_pass = 1) \vee$$

$$(il\_tl = \text{red} \wedge a = 0 \wedge b > 0 \wedge ml\_pass = 1) \vee$$

$$ml\_tl = \text{green} \vee il\_tl = \text{green} \vee a > 0 \vee c > 0$$

The previous statement reduces to the following, which is true

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

 $\leadsto$ 

$$\begin{array}{l} 0 < d \\ b \in \mathbb{N} \\ \Rightarrow \\ b < d \vee b > 0 \end{array}$$

- 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

237

---

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

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

**inv2\_1:**  $ml\_tl \in \{\text{red}, \text{green}\}$

**inv2\_2:**  $il\_tl \in \{\text{red}, \text{green}\}$

**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:**  $ml\_tl = \text{red} \vee il\_tl = \text{red}$

**inv2\_6:**  $ml\_pass \in \{0, 1\}$

**inv2\_7:**  $il\_pass \in \{0, 1\}$

**inv2\_8:**  $ml\_tl = \text{red} \Rightarrow ml\_pass = 1$

**inv2\_9:**  $il\_tl = \text{red} \Rightarrow il\_pass = 1$

**variant2:**  $ml\_pass + il\_pass$

ML\_out\_1

**when**

$ml\_tl = \text{green}$

$a + 1 + b < d$

**then**

$a := a + 1$

$ml\_pass := 1$

**end**

ML\_out\_2

**when**

$ml\_tl = \text{green}$

$a + 1 + b = d$

**then**

$a := a + 1$

$ml\_pass := 1$

$ml\_tl := \text{red}$

**end**

```
IL_out_1
  when
    il_tl = green
    b ≠ 1
  then
    b := b - 1
    c := c + 1
    il_pass := 1
  end
```

```
IL_out_2
  when
    il_tl = green
    b = 1
  then
    b := b - 1
    c := c + 1
    il_pass := 1
    il_tl := red
  end
```

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

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

- These events are identical to their abstract versions

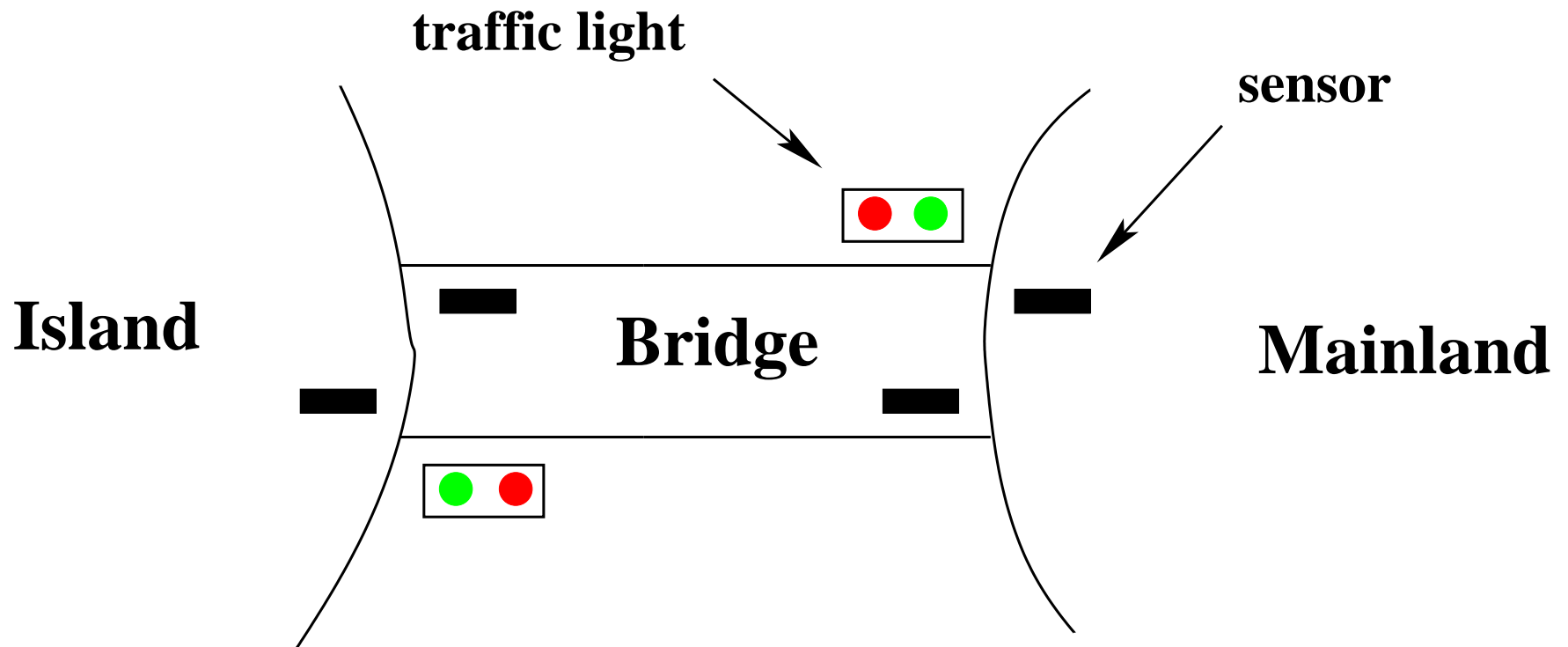
```
ML_in
  when
     $0 < c$ 
  then
     $c := c - 1$ 
  end
```

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

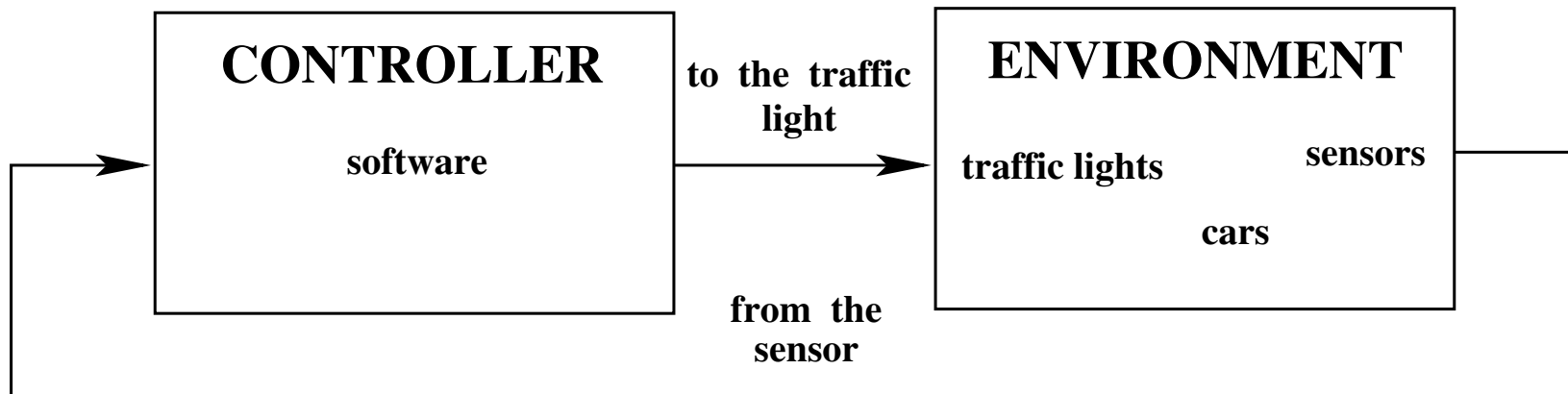
- **Initial model**: Limiting the number of cars (FUN\_2)
- **First refinement**: Introducing the one way bridge (FUN\_3)
- **Second refinement**: Introducing the traffic lights (EQP\_1,2,3)
- **Third refinement**: Introducing the sensors (EQP\_4,5)



Reminder of the **physical system**



- We want to **clearly identify** in our model:
  - The **controller**
  - The **environment**
  - The **communication channels** between the two



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

These **new variables** denote **physical objects**

Environment variables: *A*,

*B*,

*C*,

*ML\_OUT\_SR*,

*ML\_IN\_SR*,

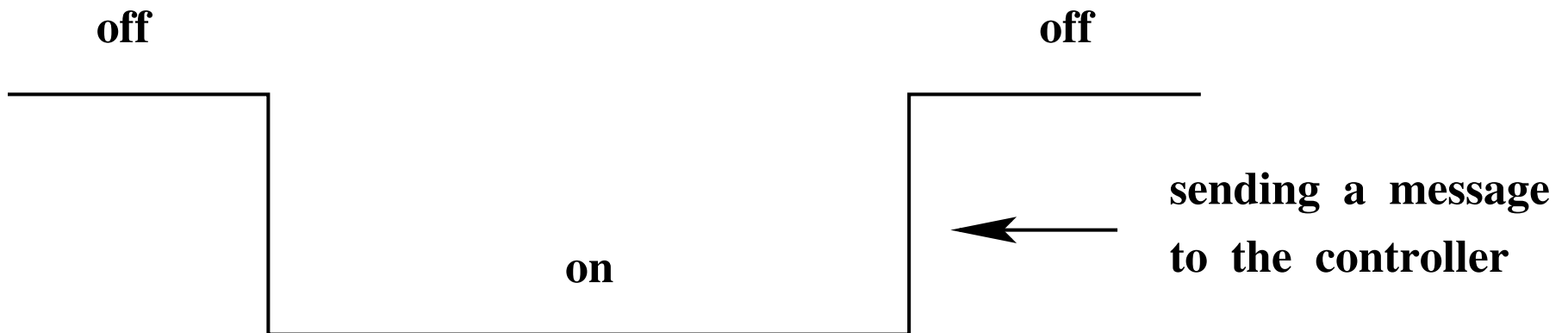
*IL\_OUT\_SR*,

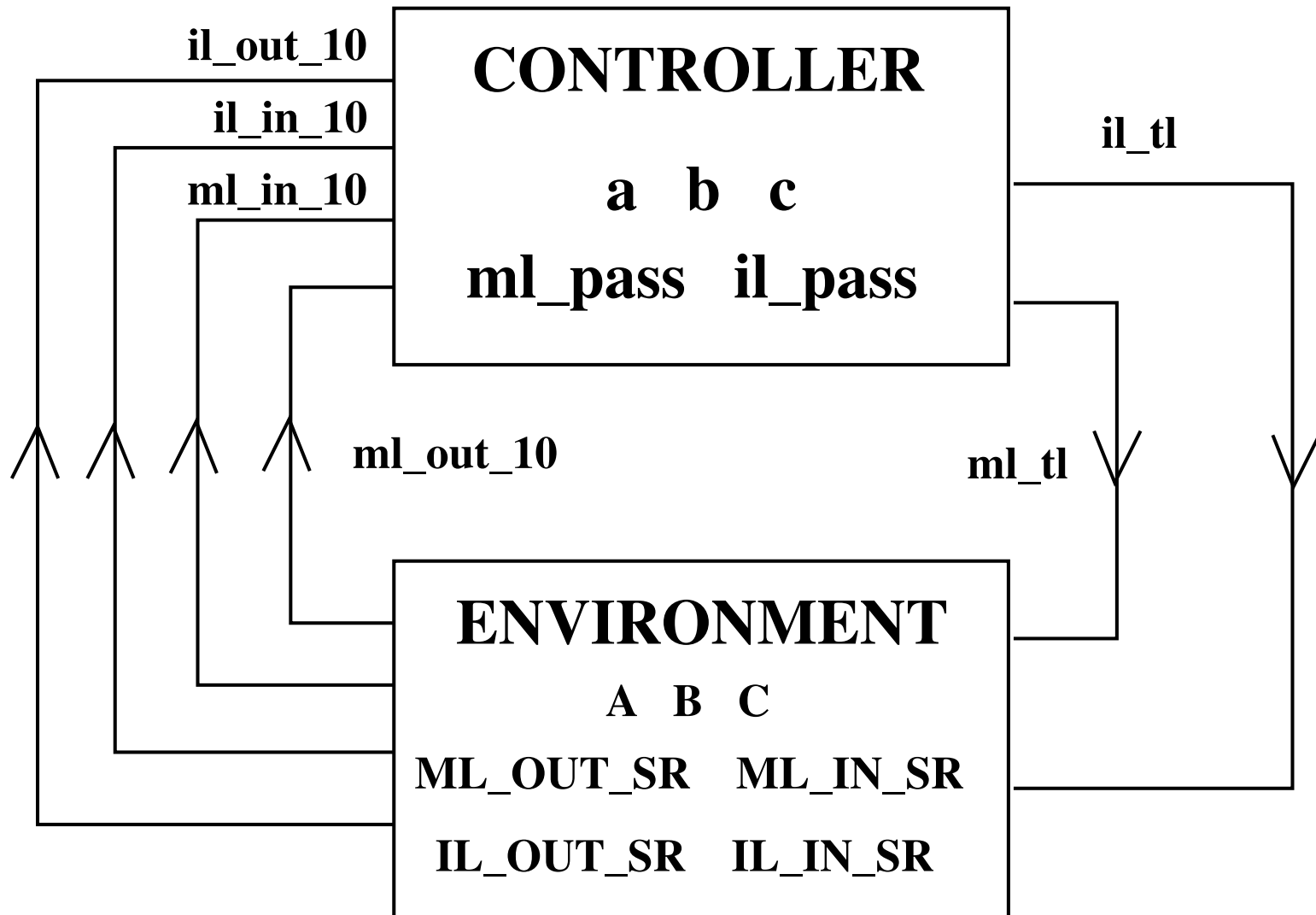
*IL\_IN\_SR*

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":





**carrier sets:**     $\dots, \textit{SENSOR}$

**constants:**     $\dots, \textit{on}, \textit{off}$

**axm3\_1:**     $\textit{SENSOR} = \{\textit{on}, \textit{off}\}$

**axm3\_2:**     $\textit{on} \neq \textit{off}$



inv3\_1 : *ML\_OUT\_SR* ∈ *SENSOR*

inv3\_2 : *ML\_IN\_SR* ∈ *SENSOR*

inv3\_3 : *IL\_OUT\_SR* ∈ *SENSOR*

inv3\_4 : *IL\_IN\_SR* ∈ *SENSOR*

inv3\_5 :  $A \in \mathbb{N}$

inv3\_6 :  $B \in \mathbb{N}$

inv3\_7 :  $C \in \mathbb{N}$

inv3\_8 :  $ml\_out\_10 \in \text{BOOL}$

inv3\_9 :  $ml\_in\_10 \in \text{BOOL}$

inv3\_10 :  $il\_out\_10 \in \text{BOOL}$

inv3\_11 :  $il\_in\_10 \in \text{BOOL}$

When sensors are on, there are cars on them

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

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

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

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

EQP-5

Drivers obey the traffic lights

$$\text{inv3\_15 : } ml\_out\_10 = \text{TRUE} \Rightarrow ml\_tl = green$$
$$\text{inv3\_16 : } il\_out\_10 = \text{TRUE} \Rightarrow il\_tl = green$$

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

inv3\_17 :  $IL\_IN\_SR = on \Rightarrow il\_in\_10 = \text{FALSE}$

inv3\_18 :  $IL\_OUT\_SR = on \Rightarrow il\_out\_10 = \text{FALSE}$

inv3\_19 :  $ML\_IN\_SR = on \Rightarrow ml\_in\_10 = \text{FALSE}$

inv3\_20 :  $ML\_OUT\_SR = on \Rightarrow ml\_out\_10 = \text{FALSE}$

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)

$$\text{inv3\_21 : } il\_in\_10 = \text{TRUE} \wedge ml\_out\_10 = \text{TRUE} \Rightarrow A = a$$

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

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

$$\text{inv3\_24 : } il\_in\_10 = \text{FALSE} \wedge ml\_out\_10 = \text{FALSE} \Rightarrow A = a$$

## Linking the physical and logical cars (2)

$$\text{inv3\_25 : } il\_in\_10 = \text{TRUE} \wedge il\_out\_10 = \text{TRUE} \Rightarrow B = b$$

$$\text{inv3\_26 : } il\_in\_10 = \text{TRUE} \wedge il\_out\_10 = \text{FALSE} \Rightarrow B = b + 1$$

$$\text{inv3\_27 : } il\_in\_10 = \text{FALSE} \wedge il\_out\_10 = \text{TRUE} \Rightarrow B = b - 1$$

$$\text{inv3\_28 : } il\_in\_10 = \text{FALSE} \wedge il\_out\_10 = \text{FALSE} \Rightarrow B = b$$

$$\text{inv3\_29 : } il\_out\_10 = \text{TRUE} \wedge ml\_out\_10 = \text{TRUE} \Rightarrow C = c$$

$$\text{inv3\_30 : } il\_out\_10 = \text{TRUE} \wedge ml\_out\_10 = \text{FALSE} \Rightarrow C = c + 1$$

$$\text{inv3\_31 : } il\_out\_10 = \text{FALSE} \wedge ml\_out\_10 = \text{TRUE} \Rightarrow C = c - 1$$

$$\text{inv3\_32 : } il\_out\_10 = \text{FALSE} \wedge ml\_out\_10 = \text{FALSE} \Rightarrow C = c$$

The basic properties hold for the physical cars

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

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

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



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

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

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

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

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

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

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

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

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

```
IL_in
  when
     $\frac{il\_in\_10 = \text{TRUE}}{0 < a}$ 
  then
     $a := a - 1$ 
     $b := b + 1$ 
     $\frac{il\_in\_10 := \text{FALSE}}$ 
  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
```

```
ML_tl_green
when
  ml_tl = red
  a + b < d
  c = 0
  il_pass = 1
  il_out_10 = FALSE
then
  ml_tl := green
  il_tl := red
  ml_pass := FALSE
end
```

```
IL_tl_green
when
  il_tl = red
  a = 0
  ml_pass = 1
  ml_out_10 = FALSE
then
  il_tl := green
  ml_tl := red
  il_pass := 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
```

```
ML_out_arr
  when
    ML_OUT_SR = off
    ml_out_10 = FALSE
  then
    ML_OUT_SR := on
  end
```

```
ML_in_arr
  when
    ML_IN_SR = off
    ml_in_10 = FALSE
    C > 0
  then
    ML_IN_SR := on
  end
```

```
IL_in_arr
  when
    IL_IN_SR = off
    il_in_10 = FALSE
    A > 0
  then
    IL_IN_SR := on
  end
```

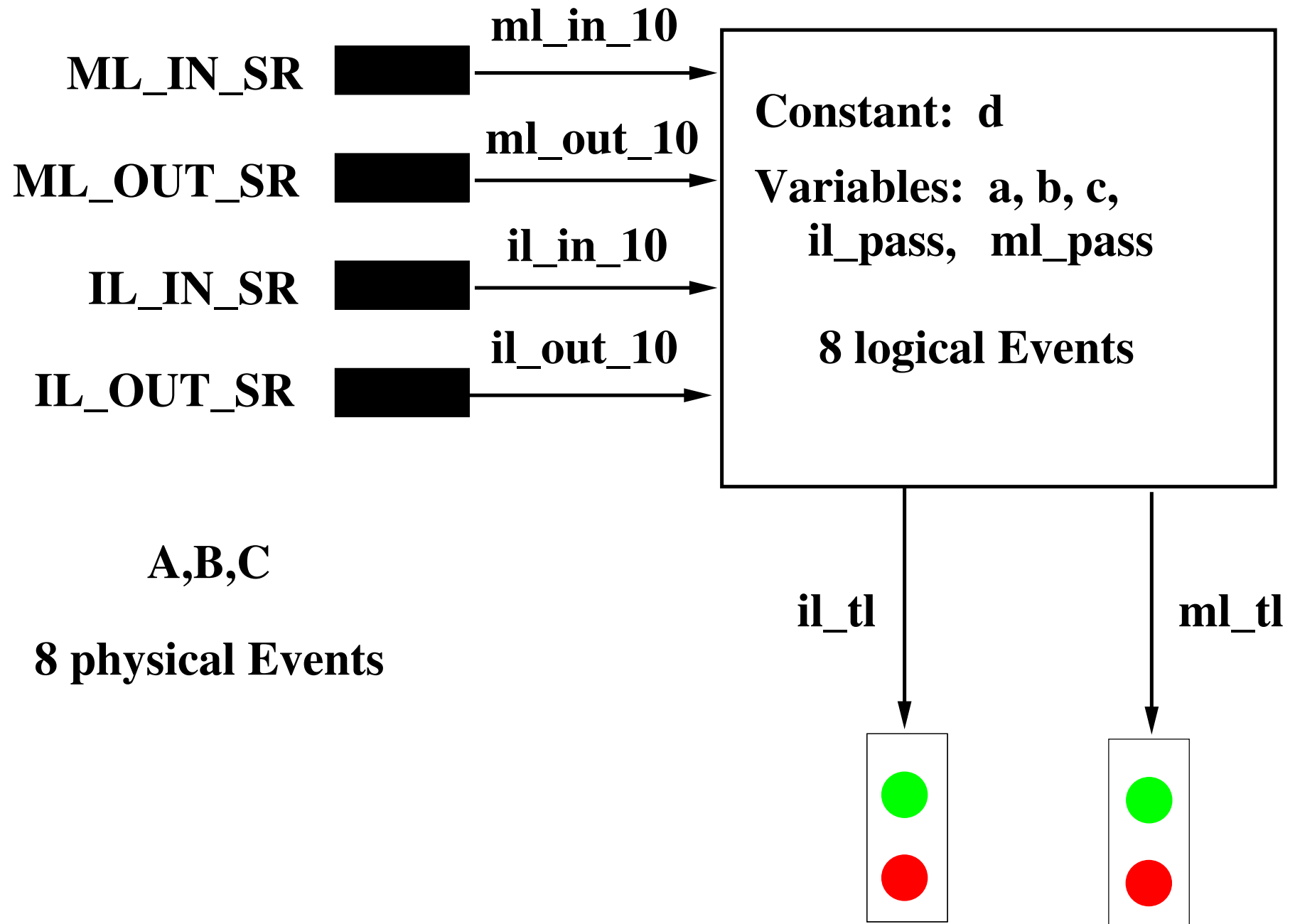
```
IL_out_arr
  when
    IL_OUT_SR = off
    il_out_10 = FALSE
    B > 0
  then
    IL_OUT_SR := on
  end
```

```
ML_out_dep
when
  ML_OUT_SR = on
  ml_tl = green
then
  ML_OUT_SR := off
  ml_out_10 := TRUE
  A := A + 1
end
```

```
ML_in_dep
when
  ML_IN_SR = on
then
  ML_IN_SR := off
  ml_in_10 := TRUE
  C := C - 1
end
```

```
IL_in_dep
when
  IL_IN_SR = on
then
  IL_IN_SR := off
  il_in_10 := 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 := TRUE
  B := B - 1
  C := C + 1
end
```



- 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?



- **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

- This development required 252 proofs
  - Initial model: 6 (0)
  - 1st refinement: 27 (0)
  - 2nd refinement: 81 (1)
  - 3rd refinement: 138 (3)
- All proved automatically (except 4) by the Rodin Platform

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

---

$\mathbb{N}$	set of Natural Numbers: $\{0, 1, 2, 3, \dots\}$
$\mathbb{Z}$	set of Integers: $\{0, 1, -1, 2, -2, \dots\}$
$\{a, b, \dots\}$	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

- For the init event in the initial model

$\begin{array}{l} \text{Axioms of the constants} \\ \Rightarrow \\ \text{Modified Invariants} \end{array}$	$\text{INV}$
--	--------------

- For other events in the initial model

<p>Axioms of the constants Invariants Guard of the event <math>\Rightarrow</math> Modified Invariants</p>	<p>INV</p>
---	------------



- This rule is not mandatory

<p>Axiom of the constant Invariants</p> <p><math>\Rightarrow</math></p> <p>Disjunction of the guards</p>	<p>DLF</p>
--	------------

- For old events only

<p>Axioms of the constants Abstract invariants Concrete invariants Concrete guards <math>\Rightarrow</math> Abstract guards</p>	<p>GRD</p>
---	------------

- For init event only

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

- For all events (except init)
- New events refine an implicit non-guarded event with skip action

<p>Axioms of the constants Abstract invariant Concrete invariant Concrete guard <math>\Rightarrow</math> Modified concrete invariant</p>	<p>INV</p>
--	------------

# Refinement Rules (4): Non-divergence of New Events

281

---

- 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

282

---

- For new events only

Axioms

Abstract invariants

Concrete invariants

Concrete guard

⊢

Modified Var.  $<$  Var.

# Refinement Rules (6): Relative Deadlock Freeness

283

---

- 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)

<p>Axioms of constants Abstract invariants Concrete invariants Concrete guards <math>\Rightarrow</math> Same actions on common variables</p>	<p>SIM</p>
--	------------