DÉVELOPPEMENT DE SYSTÈMES CRITIQUES AVEC LA MÉTHODE EVENT-B

MODÉLISATION, RAFFINEMENT ET PREUVE

3A cursus ingénieurs - Mention Sciences du Logiciel
CentraleSupelec - Université Paris-Saclay - 2024/2025

Idir AIT SADOUNE
idir.aitsadoune@centralesupelec.fr

OUTLINE

Back to the outline - Back to the begin

CentraleSupélec

OUTLINE

Back to the outline - Back to the begin

CentraleSupélec

THE RODIN PLATFORM

- The Rodin Platform is an Eclipse-based IDE for Event-B that provides effective support for refinement and mathematical proof.
- The platform is open source, contributes to the Eclipse framework and is further extendable with plugins.
- Rodin Platform and Plug-in Installation:
  - Requires Java 17
  - Download the Core: Rodin Platform file for your platform.
  - Install the Atelier B Provers plugin from the Atelier B Provers Update site.

CentraleSupélec

PURPOSE OF THIS LECTURE

- To present an **example of system development**
- Our approach $\rightarrow$ 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 **observer** *“zooms in”* to see more details

CentraleSupélec

PURPOSE OF THIS LECTURE

- 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**

WHAT YOU WILL LEARN

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

Remark

Theoretical background provided during development.

CentraleSupélec

OUTLINE

Back to the outline - Back to the begin

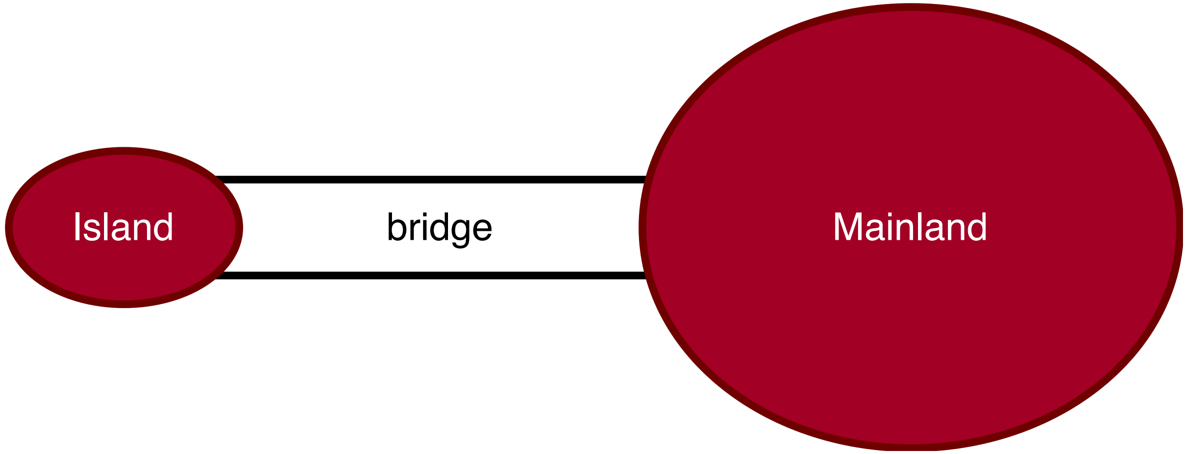
A REQUIREMENTS DOCUMENT

- The system we are going to build is a **piece of software** connected to some **equipment**.
- There are two kinds of requirements:
  - those concerned with the **equipment**, labeled **EQP**,
  - those concerned with the **function** of the system, labeled **FUN**.
- The function of this system is to **control cars** on a **narrow bridge**.
- This bridge is supposed to link the **mainland** to a small **island**.

CentraleSupélec

A REQUIREMENTS DOCUMENT

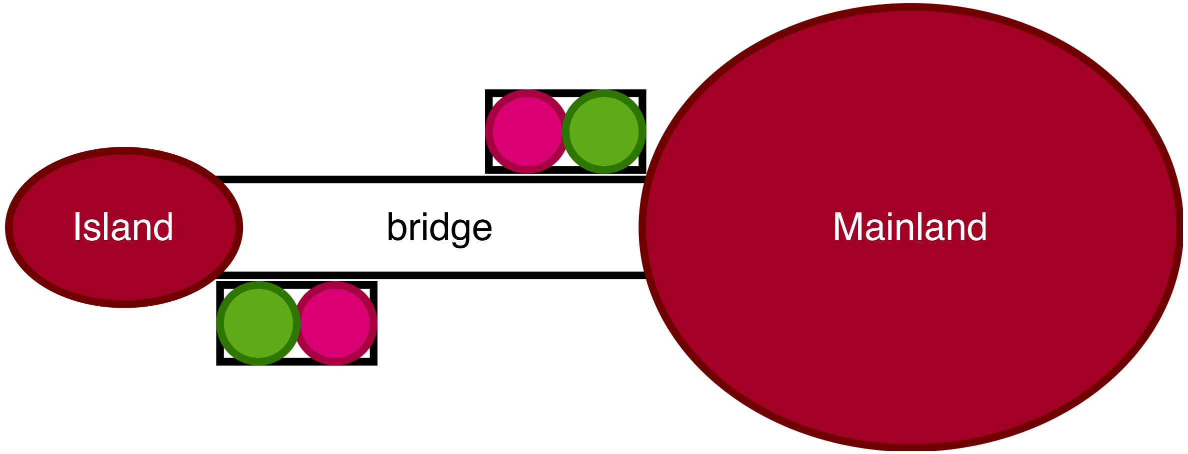
- **FUN-1** $\longrightarrow$ the system is controlling cars on a bridge between the mainland and an island.



CentraleSupélec

A REQUIREMENTS DOCUMENT

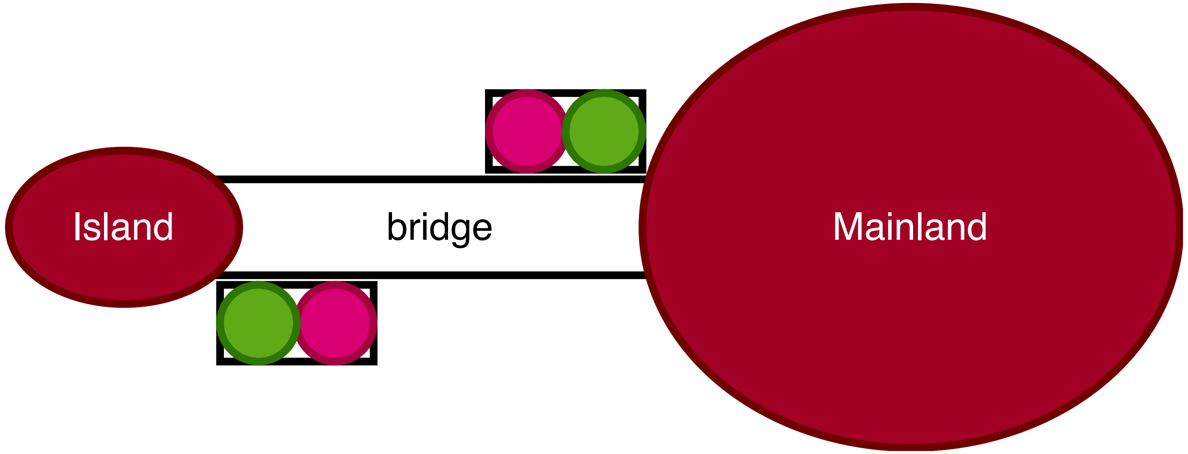
- **EQP-1** $\longrightarrow$ the system has two traffic lights with two colors: green and red, one of the traffic lights is situated on the mainland and the other one on the island Both are close to the bridge.



CentraleSupélec

A REQUIREMENTS DOCUMENT

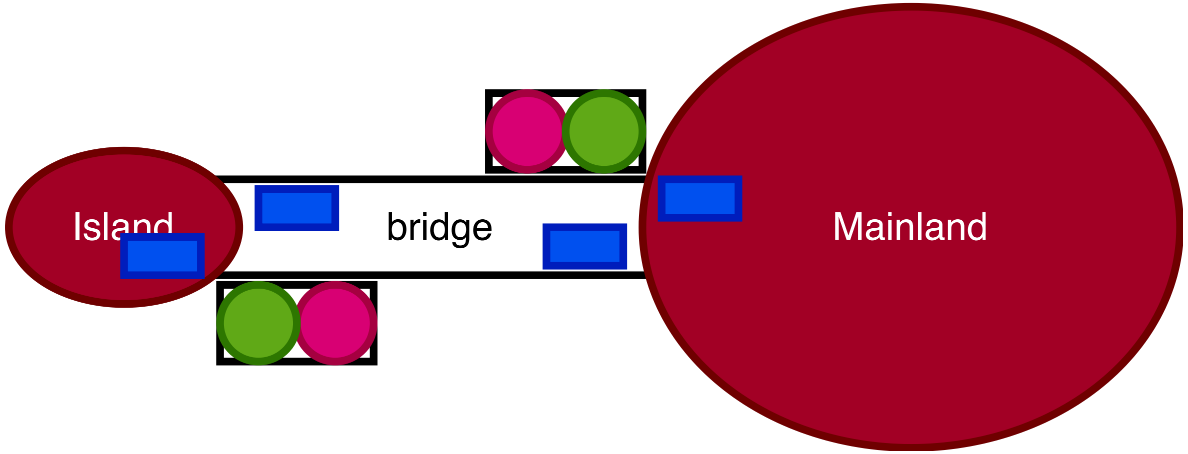
- EQP-2 $\longrightarrow$ the traffic lights control the entrance to the bridge at both ends of it.
- EQP-3 $\longrightarrow$ cars are not supposed to pass on a red traffic light, only on a green one.



CentraleSupélec

A REQUIREMENTS DOCUMENT

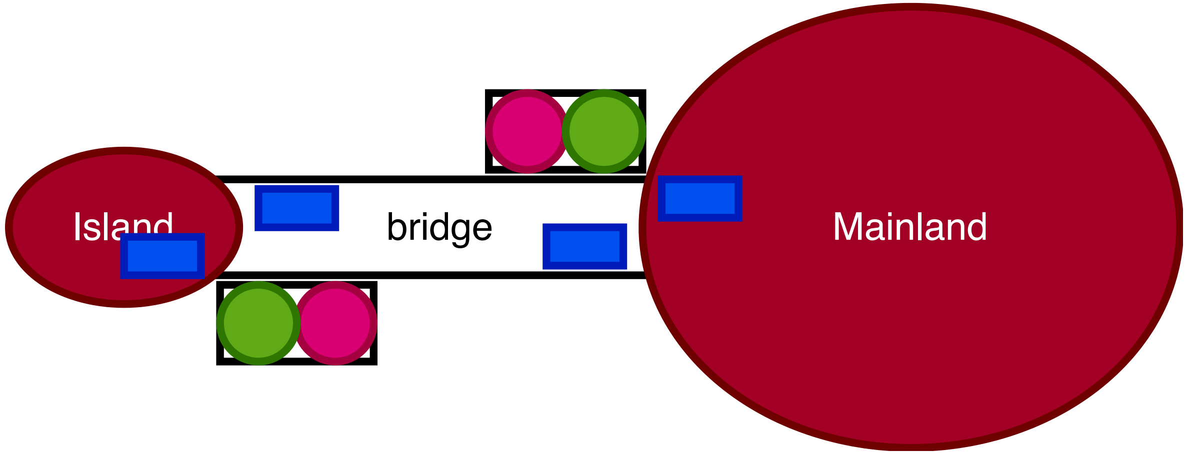
- **EQP-4** $\longrightarrow$ the system is equipped with four car sensors each with two states: on or off.
- **EQP-5** $\longrightarrow$ the sensors are used to detect the presence of cars entering or leaving the bridge.



CentraleSupélec

A REQUIREMENTS DOCUMENT

- **FUN-2** $\longrightarrow$ the number of cars on the bridge and the island is limited.
- **FUN-3** $\longrightarrow$ the bridge is one way or the other, not both at the same time.



CentraleSupélec

OUTLINE

Back to the outline - Back to the begin

CentraleSupélec

OUR REFINEMENT STRATEGY

- **Initial model** $\rightarrow$ Limiting the number of cars (**FUN-2**)
- **First refinement** $\rightarrow$ Introducing the one way bridge (**FUN-3**)
- **Second refinement** $\rightarrow$ Introducing the traffic lights (**EQP-1,2,3**)
- **Third refinement** $\rightarrow$ Introducing the sensors (**EQP-4,5**)

CentraleSupélec

OUTLINE

Back to the outline - Back to the begin

OUR REFINEMENT STRATEGY

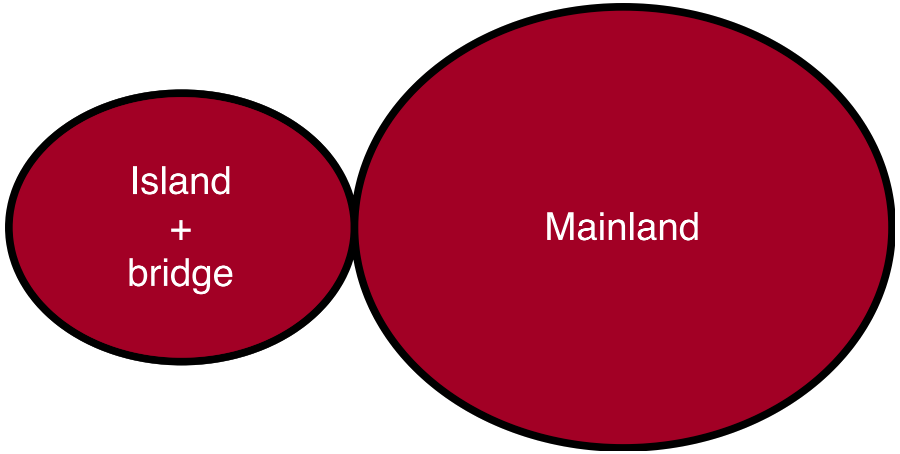
- **Initial model** $\rightarrow$ Limiting the number of cars (**FUN-2**)
- **First refinement** $\rightarrow$ Introducing the one way bridge (**FUN-3**)
- **Second refinement** $\rightarrow$ Introducing the traffic lights (**EQP-1,2,3**)
- **Third refinement** $\rightarrow$ Introducing the sensors (**EQP-4,5**)

CentraleSupélec

INITIAL MODEL

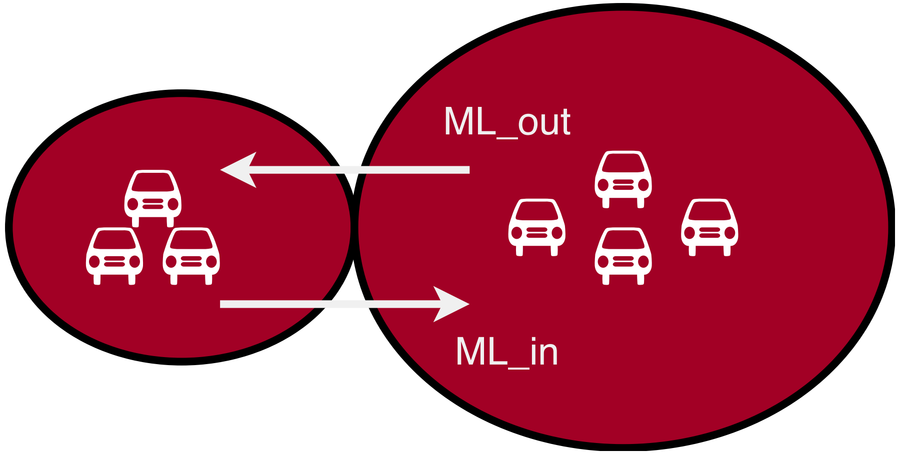
- It is very simple
- We completely ignore the equipment $\rightarrow$ traffic lights and sensors
- We do not even consider the bridge
- We are just interested in the pair “island-bridge”
- We are focusing FUN-2 $\rightarrow$ limited number of cars on island-bridge

A SITUATION AS SEEN FROM THE SKY



CentraleSupélec

TWO EVENTS THAT MAY BE OBSERVED



FORMALIZING THE STATE

- STATIC PART of the state $\rightarrow$ constant d with axiom axm0_1

```
CONSTANTS
   d
AXIOMS
   axm0_1:  d ∈ 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 $\mathbb{N} = \{0, 1, 2, \ldots\}$

CentraleSupélec

FORMALIZING THE STATE

- **DYNAMIC PART** of the state $\rightarrow$ **variable** n with **invariants** `inv0_1` and `inv0_2`

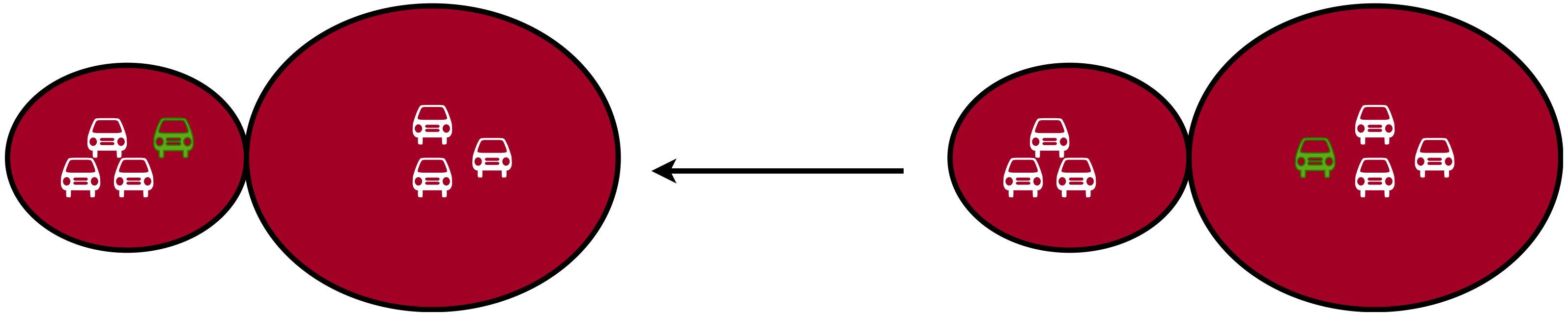
```
VARIABLES
  n
INVARIANTS
  inv0_1:  n ∈ ℕ
  inv0_2:  n ≤ d
```

- 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`) $\rightarrow$ this is **FUN 2**

CentraleSupélec

EVENT ML_out

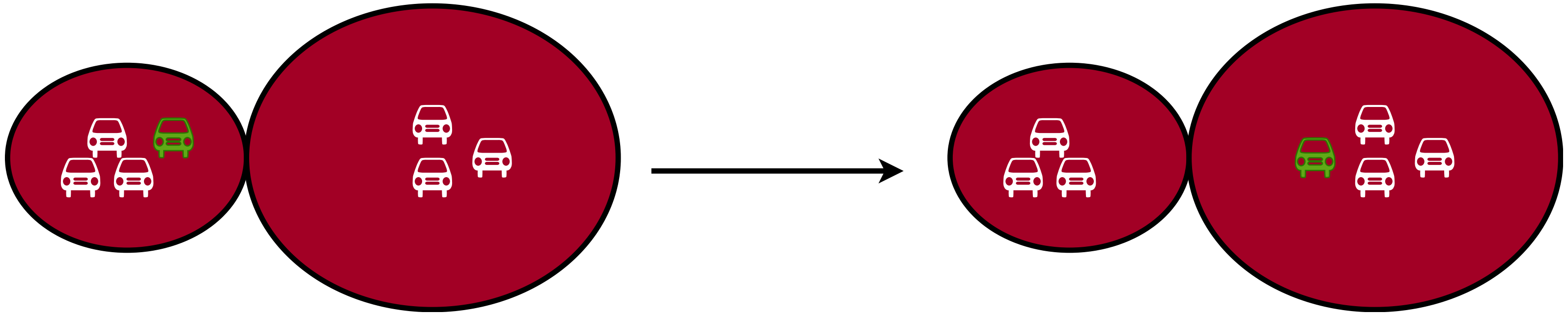
- This is the **first transition** (or event) that can be **observed**
- A car is leaving the mainland and entering the Island-Bridge



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

EVENT `ML_in`

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



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

FORMALIZING THE TWO EVENTS (APPROXIMATION)

- An event is denoted by its **name** and its **action** (an assignment)
- Event `ML_out` **increments** the number of cars

```
ML_out ≙
  then
    act0_1: n := n + 1
  end
```

- Event `ML_in` **decrements** the number of cars

```
ML_in ≙
  then
    act0_1: n := n − 1
  end
```

CentraleSupélec

WHY AN APPROXIMATION?

- These events are approximations for **two reasons**:
  1. They might be **insufficient** at this stage because **not consistent with the invariant**
  2. They might be **refined** (made more precise) later
- We have to perform a **proof** in order to **verify this consistency**.

CentraleSupélec

INVARIANTS

- 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**, and
  2. the invariant is **preserved by all events**
- We will formalize these two statements as **proof obligations (POs)**
- We need a **rigorous proof** showing that these POs indeed hold

CentraleSupélec

BEFORE-AFTER PREDICATES

- 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'

CentraleSupélec

BEFORE-AFTER PREDICATES

EXAMPLE

- The **events**

```
ML_out ≙
  then
    act0_1: n := n + 1
  end
```

```
ML_in ≙
  then
    act0_1: n := n − 1
  end
```

- The corresponding **before-after predicates**

$$n' = n + 1$$

$$n' = n - 1$$

These representations are equivalent.

CentraleSupélec

ABOUT THE SHAPE OF THE BEFORE-AFTER PREDICATES

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

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

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

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

CentraleSupélec

INTUITION
ABOUT INVARIANT PRESERVATION

- Let us consider invariant `inv0_1`

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

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

$$n' = n + 1$$

- Preservation of `inv0_1` means that we have (just after `ML_out`):

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

CentraleSupélec

BEING MORE PRECISE

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

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

- This type of statement is called a **sequent**
- Sequent above $\rightarrow$ invariant preservation proof obligation for `inv0_1`

CentraleSupélec

SEQUENTS

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

$$H \quad \vdash \quad G$$

- H denotes a set of predicates $\rightarrow$ the hypotheses (or assumptions)
- G denotes a predicate $\rightarrow$ the goal (or conclusion)
- The symbol $\vdash$, called the turnstyle, stands for provability.
  It is read $\rightarrow$ *Assumptions* H *yield conclusion* G

CentraleSupélec

PROOF OBLIGATION

INVARIANT PRESERVATION

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

CentraleSupélec

PROOF OBLIGATION

INVARIANT PRESERVATION

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

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

- It says $\rightarrow$ $I_i(c, E(c, v))$ provable under hypotheses $A(c)$ and $I(c, v)$
- We have given the name INV to this proof obligation

CentraleSupélec

EXPLANATION
OF THE PROOF OBLIGATION

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

- 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

CentraleSupélec

VERTICAL LAYOUT OF PROOF OBLIGATIONS

- The proof obligation

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

- can be re-written vertically as follows

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

CentraleSupélec

BACK TO OUR EXAMPLE

- We have two events

```
ML_out ≙
  then
    act0_1: n := n + 1
  end
```

```
ML_in ≙
  then
    act0_1: n := n − 1
  end
```

- ... and two invariants

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

inv0_2: $n \leq d$

- Thus, we need to prove four proof obligations

CentraleSupélec

PROOF OBLIGATION FOR ML_out AND inv0_1

```
ML_out ≙
  then
    act0_1: n := n + 1 // n' = n + 1
  end
```

Axioms axm0_1	$d \in \mathbb{N}$
Invariant inv0_1	$n \in \mathbb{N}$
Invariant inv0_2	$n \leq d$
$\vdash$	$\vdash$
Modified Invariant inv0_1	$n + 1 \in \mathbb{N}$

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

CentraleSupélec

PROOF OBLIGATION FOR ML_out AND inv0_2

ML_out $\widehat{=}$
  then
    act0_1: $n := n + 1$ // $n' = n + 1$
  end

Axioms axm0_1	$d \in \mathbb{N}$
Invariant inv0_1	$n \in \mathbb{N}$
Invariant inv0_2	$n \leq d$
$\vdash$	$\vdash$
Modified Invariant inv0_2	$n + 1 \leq d$

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

CentraleSupélec

PROOF OBLIGATION FOR `ML_in` AND `inv0_1`

```
ML_in ≙
  then
    act0_1: n := n − 1 // n′ = n − 1
  end
```

Axioms `axm0_1`	$d \in \mathbb{N}$
Invariant `inv0_1`	$n \in \mathbb{N}$
Invariant `inv0_2`	$n \leq d$
$\vdash$	$\vdash$
Modified Invariant `inv0_1`	$n - 1 \in \mathbb{N}$

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

CentraleSupélec

PROOF OBLIGATION FOR ML_in AND inv0_2

```
ML_in ≙
  then
    act0_1: n := n − 1 // n' = n − 1
  end
```

Axioms	axm0_1	$d \in \mathbb{N}$
Invariant	inv0_1	$n \in \mathbb{N}$
Invariant	inv0_2	$n \leq d$
$\vdash$		$\vdash$
Modified Invariant	inv0_2	$n - 1 \leq d$

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

CentraleSupélec

SUMMARY OF PROOF OBLIGATIONS

ML_out/inv0_1/INV

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

ML_out/inv0_2/INV

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

ML_in/inv0_1/INV

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

ML_in/inv0_2/INV

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

CentraleSupélec

INFORMAL PROOF OF ML_out/inv0_1/INV

$$\boxed{\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ \vdash \\ n+1 \in \mathbb{N} \end{array}} \quad \underset{\Longrightarrow}{\text{remove hypotheses}} \quad \boxed{\begin{array}{l} n \in \mathbb{N} \\ \vdash \\ n+1 \in \mathbb{N} \end{array}} \quad \underset{\surd}{\text{obvious}}$$

- 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

CentraleSupélec

INFERENCE RULES

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

- Above horizontal line $\longrightarrow$ n sequents called antecedents ($n \geq 0$)
- Below horizontal line $\longrightarrow$ 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**

CentraleSupélec

INFERENCE RULES

MONOTONICITY OF HYPOTHESES

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

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

- It expresses the monotonicity of the hypotheses

CentraleSupélec

SOME ARITHMETIC INFERENCE RULES

THE SECOND PEANO AXIOM

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

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

CentraleSupélec

MORE ARITHMETIC INFERENCE RULES

AXIOMS ABOUT ORDERING RELATIONS ON THE INTEGERS

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

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

APPLICATION OF INFERENCE RULES

- Consider again the 2^{nd} Peano axiom:

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

- 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:

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

CentraleSupélec

PROOFS

- 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.

CentraleSupélec

A FORMAL PROOF OF ML_out/inv0_1/INV

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

Proof requires only application of two rules $\longrightarrow$ MON and P2

CentraleSupélec

A FAILED PROOF ATTEMPT

ML_out/inv0_2/INV

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

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

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

CentraleSupélec

A FAILED PROOF ATTEMPT

`ML_in/inv0_1/INV`

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

- **The proof fails** $\rightarrow$ we cannot conclude with rule $\mathrm{P2'}$ ($0 < n$ needed)

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

A FORMAL PROOF OF ML_in/inv0_2/INV

$$\boxed{\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ \vdash \\ n - 1 \leq d \end{array}} \quad \overset{\text{MON}}{\Longrightarrow} \quad \boxed{\begin{array}{l} n \leq d \\ \vdash \\ n - 1 \leq d \end{array}} \quad \overset{\text{DEC}}{\surd}$$

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

CentraleSupélec

REASONS FOR PROOF FAILURE

- We needed hypothesis $n < d$ to prove `ML_out/inv0_2/INV`
- We needed hypothesis $0 < n$ to prove `ML_in/inv0_1/INV`

```
ML_out ≙
  then
    act0_1: n := n + 1
  end
```

```
ML_in ≙
  then
    act0_1: n := n − 1
  end
```

- 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`

CentraleSupélec

IMPROVING THE EVENTS

INTRODUCING GUARDS

```
ML_out ≙
  when
    grd0_1: n < d
  then
    act0_1: n := n + 1
  end
```

```
ML_in ≙
  when
    grd0_1: 0 < n
  then
    act0_1: n := n − 1
  end
```

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

CentraleSupélec

PROOF OBLIGATION

GENERAL INVARIANT PRESERVATION

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

CentraleSupélec

A FORMAL PROOF OF ML_out/inv0_1/INV

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

Adding new assumptions to a sequent does not affect its provability

CentraleSupélec

A FORMAL PROOF OF ML_out/inv0_2/INV

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

- Now we can conclude the proof using rule INC

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

CentraleSupélec

A FORMAL PROOF OF ML_in/inv0_1/INV

$$\boxed{\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ {\color{red}0 < n} \\ \vdash \\ n - 1 \in \mathbb{N} \end{array}} \quad \overset{\texttt{MON}}{\Longrightarrow} \quad \boxed{\begin{array}{l} 0 < n \\ \vdash \\ n - 1 \in \mathbb{N} \end{array}} \quad \begin{array}{c} \texttt{P2'} \\ \surd \end{array}$$

- Now we can conclude the proof using rule $\mathrm{P2'}$

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

CentraleSupélec

A FORMAL PROOF OF `ML_in/inv0_2/INV`

$$\boxed{\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ \color{red}{0 < n} \\ \vdash \\ n - 1 \leq d \end{array}} \quad \underset{\Longrightarrow}{\text{MON}} \quad \boxed{\begin{array}{l} n \leq d \\ \vdash \\ n - 1 \leq d \end{array}} \quad \underset{\surd}{\text{DEC}}$$

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

CentraleSupélec

RE-PROVING THE EVENTS
NO PROOFS FAIL

ML_out/inv0_1/INV

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

ML_out/inv0_2/INV

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

ML_in/inv0_1/INV

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \leq d$
$0 < n$
$\vdash$
$n - 1 \in \mathbb{N}$

ML_in/inv0_2/INV

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \leq d$
$0 < n$
$\vdash$
$n - 1 \leq d$

CentraleSupélec

INITIALISATION

- Our system must be initialized (with no car in the island-bridge)
- The initialisation 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 $\widehat{=}$
  begin
    init0_1: $n := 0$
  end

After predicate $\Longrightarrow$ $n' = 0$

CentraleSupélec

PROOF OBLIGATION INVARIANT ESTABLISHMENT

- 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:

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

CentraleSupélec

APPLYING THE INVARIANT ESTABLISHMENT PO

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

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

CentraleSupélec

MORE ARITHMETIC INFERENCE RULES

- First Peano Axiom

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

- Third Peano Axiom (slightly modified)

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

CentraleSupélec

PROOFS OF INVARIANT ESTABLISHMENT

$$\boxed{\begin{array}{l} d \in \mathbb{N} \\ \vdash \\ 0 \in \mathbb{N} \end{array}} \quad \overset{\text{MON}}{\Longrightarrow} \quad \boxed{\begin{array}{l} \vdash \\ 0 \in \mathbb{N} \end{array}} \quad \begin{array}{c} \text{P1} \\ \surd \end{array}$$

$$\boxed{\begin{array}{l} d \in \mathbb{N} \\ \vdash \\ 0 \leq d \end{array}} \quad \begin{array}{c} \text{P3} \\ \surd \end{array}$$

A MISSING REQUIREMENT

- It is possible for the system to be blocked if both guards are false
- We do not want this to happen
- We figure out that one important requirement was missing
- **FUN-4** $\longrightarrow$ Once started, the system should work for ever (Deadlock Freedom)

PROOF OBLIGATION

THE THEOREM PO RULE

- Given c with axioms $A(c)$ and v with invariants $I(c, v)$
- Given the theorem $Th(c, v)$
- Given the guards $G_1(c, v), \ldots, G_m(c, v)$ of the events
- We have to prove the following:

$$\begin{array}{l} A(c) \\ I(c, v) \\ \vdash \\ Th(c, v) \end{array} \qquad \begin{array}{l} A(c) \\ I(c, v) \\ \vdash \\ G_1(c, v) \ \lor \ \ldots \ \lor \ G_m(c, v) \end{array}$$

CentraleSupélec

APPLYING THE DEADLOCK FREEDOM PO

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

- 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: $n < d$

CentraleSupélec

INFERENCE RULES FOR DISJUNCTION

- Proof by case analysis

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

- Choice for proving a disjunctive goal

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

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

CentraleSupélec

PROOF OF DEADLOCK FREEDOM

$$\boxed{\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ \vdash \\ n < d \ \lor \ 0 < n \end{array}} \quad \overset{\text{MON}}{\Longrightarrow} \quad \boxed{\begin{array}{l} n \leq d \\ \vdash \\ n < d \ \lor \ 0 < n \end{array}} \quad \overset{\text{OR\_L}}{\Longrightarrow} \quad \begin{array}{l} \boxed{\begin{array}{l} n < d \\ \vdash \\ n < d \ \lor \ 0 < n \end{array}} \\ \\ \boxed{\begin{array}{l} n = d \\ \vdash \\ n < d \ \lor \ 0 < n \end{array}} \end{array}$$

$$\boxed{\begin{array}{l} n < d \\ \vdash \\ n < d \ \lor \ 0 < n \end{array}} \quad \overset{\text{OR\_R1}}{\Longrightarrow} \quad \boxed{\begin{array}{l} n < d \\ \vdash \\ n < d \end{array}} \quad \overset{?}{\Longrightarrow}$$

seems to be obvious

$$\boxed{\begin{array}{l} n = d \\ \vdash \\ n < d \ \lor \ 0 < n \end{array}} \quad \overset{?}{\Longrightarrow}$$

can be (partially) solved
by applying the equality

MORE INFERENCE RULES

IDENTITY AND EQUALITY

- The identity axiom (conclusion holds by hypothesis)

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

- Rewriting an equality (**EQ_LR**) and reflexivity of equality (**EQL**)

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

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

CentraleSupélec

PROOF OF DEADLOCK FREEDOM

$$\boxed{\begin{array}{l} n < d \\ \vdash \\ n < d \ \lor \ 0 < n \end{array}} \quad \overset{\text{OR\_R1}}{\Longrightarrow} \quad \boxed{\begin{array}{l} n < d \\ \vdash \\ n < d \end{array}} \quad \overset{\text{HYP}}{\surd}$$

$$\boxed{\begin{array}{l} n = d \\ \vdash \\ n < d \ \lor \ 0 < n \end{array}} \quad \overset{\text{EQ\_LR}}{\Longrightarrow} \quad \boxed{\begin{array}{l} \vdash \\ d < d \ \lor \ 0 < d \end{array}} \quad \overset{\text{OR\_R2}}{\Longrightarrow} \quad \boxed{\begin{array}{l} \vdash \\ 0 < d \ ? \end{array}}$$

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

CentraleSupélec

ADDING THE FORGOTTEN AXIOM

- If $d = 0$, then no car can ever enter the Island-Bridge

```
CONSTANTS
   d
AXIOMS
   axm0_1: d ∈ ℕ
   axm0_2: 0 < d
```

CentraleSupélec

INITIAL MODEL

CONCLUSION

- 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

THANK YOU

PDF version of the slides

Back to the begin - Back to the outline

CentraleSupélec