Laboratoire Méthodes Formelles

CentraleSupélec | université PARIS-SACLAY

DÉVELOPPEMENT DE SYSTÈMES CRITIQUES AVEC LA MÉTHODE EVENT-B

LE CONCEPT DE LA MODÉLISATION PAR RAFFINEMENT

🎓 3A cursus ingénieurs - Mention Sciences du Logiciel
🏛 CentraleSupelec - Université Paris-Saclay - 2025/2026

Idir AIT SADOUNE
idir.aitsadoune@centralesupelec.fr

PLAN

Retour au plan - Retour à l'accueil

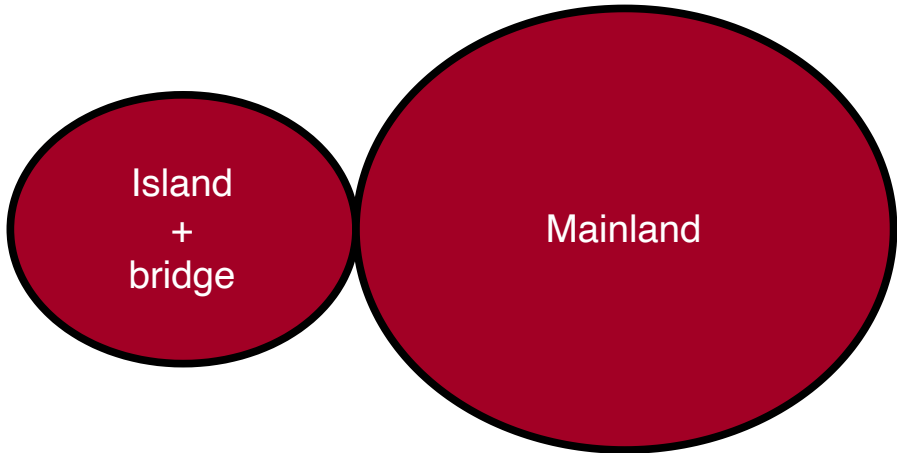
CentraleSupélec
université PARIS-SACLAY

PLAN

Retour au plan - Retour à l'accueil

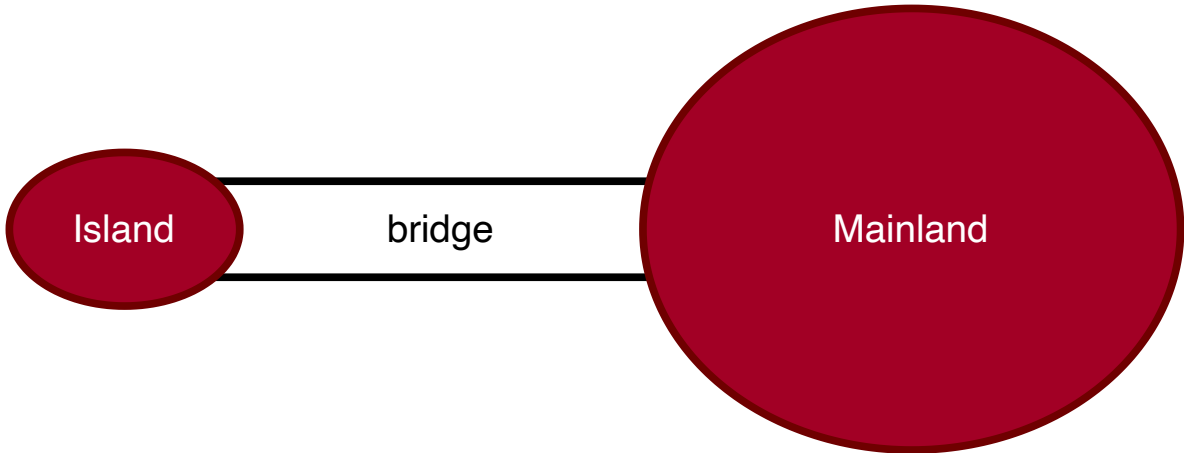
CentraleSupélec
université PARIS-SACLAY

A SITUATION AS SEEN FROM THE SKY



CentraleSupélec
université PARIS-SACLAY

A SITUATION AS SEEN FROM THE SKY



CentraleSupélec
université PARIS-SACLAY

OUR REFINEMENT STRATEGY

- **Initial model** $\longrightarrow$ Limiting the number of cars (**FUN-2**)
- **First refinement** $\longrightarrow$ Introducing the one way bridge (**FUN-1**, **FUN-3**)
  - Our view of the system gets more accurate
  - We introduce the bridge and separate it from the island (**FUN-1**)
  - We refine the state and the events
  - We also add two new events $\longrightarrow$ `IL_in` and `IL_out`
  - We are focusing on **FUN-3** $\longrightarrow$ one-way bridge

CentraleSupélec
université PARIS-SACLAY

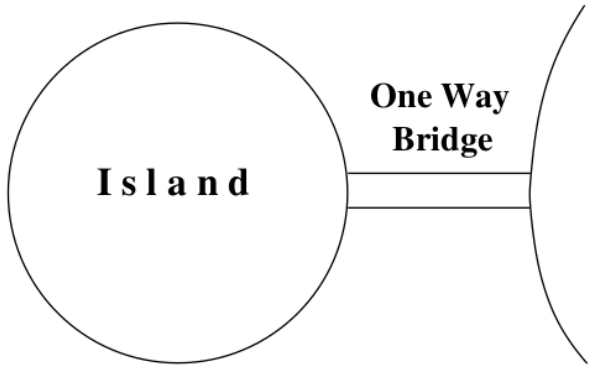
PLAN

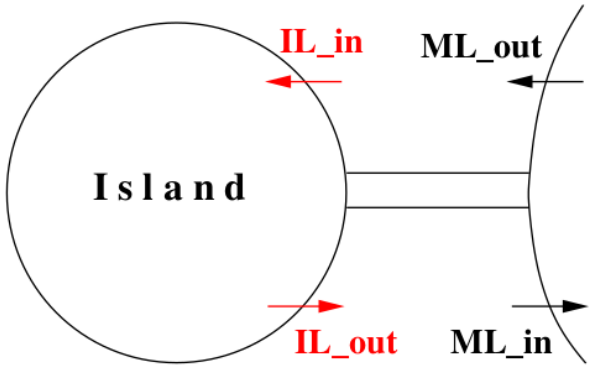
Retour au plan - Retour à l'accueil

CentraleSupélec
université PARIS-SACLAY

FIRST REFINEMENT

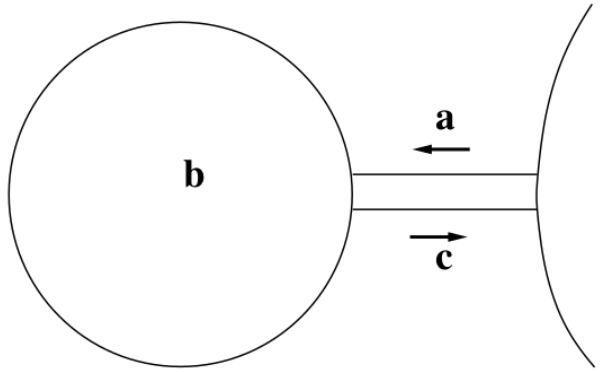
INTRODUCING A ONE-WAY BRIDGE





CentraleSupélec
université PARIS-SACLAY

INTRODUCING THREE NEW VARIABLES



- 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

CentraleSupélec
université PARIS-SACLAY

REFINING THE STATE

- Variables a, b, and c denote natural numbers

VARIABLES

$a \;\; b \;\; c$

INVARIANTS

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

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

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

CentraleSupélec

université PARIS-SACLAY

REFINING THE STATE

INTRODUCING NEW INVARIANTS

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

  INVARIANTS

  ...

  `inv1_4:` $a + b + c = n$

- Formalizing the new invariant $\longrightarrow$ one way bridge (this is **FUN-3**)

  INVARIANTS

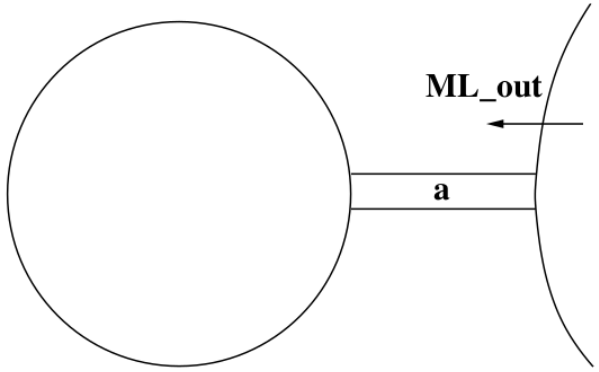
  ...

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

- Invariants `inv1_1` to `inv1_5` are called the concrete invariants
- `inv1_4` **glues** the abstract state, n, to the concrete state, a, b, c

CentraleSupélec
université PARIS-SACLAY

PROPOSAL FOR REFINING EVENT ML_out



ML_out $\widehat{=}$

when

grd1_1: $a + b < d$

grd1_2: $c = 0$

then

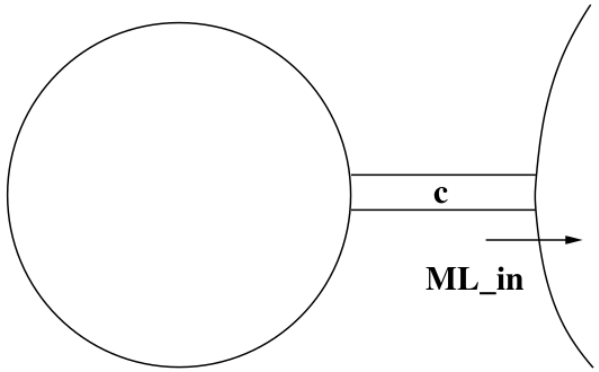
act1_1: $a := a + 1$

end

CentraleSupélec

université PARIS-SACLAY

PROPOSAL FOR REFINING EVENT ML_in



ML_in $\widehat{=}$
  when
    grd1_1: $0 < c$
  then
    act1_1: $c := c - 1$
  end

CentraleSupélec
université PARIS-SACLAY

BEFORE-AFTER PREDICATES

PRESERVED VARIABLES

ML_out $\widehat{=}$
  when
    grd1_1: $a + b < d$
    grd1_2: $c = 0$
  then
    act1_1: $a := a + 1$
  end

ML_in $\widehat{=}$
  when
    grd1_1: $0 < c$
  then
    act1_1: $c := c - 1$
  end

Before-after predicates showing the unmodified variables

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

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

CentraleSupélec
université PARIS-SACLAY

INTUITION ABOUT REFINEMENT

- 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 $\longrightarrow$ following slides)
  2. to every concrete initial state corresponds an abstract one (initial state refinement $\longrightarrow$ later)
- We will make these two conditions more precise and formalize them as proof obligations.

CentraleSupélec
université PARIS-SACLAY

INTUITION ABOUT REFINEMENT

ML_out $\widehat{=}$ //abstract
when
grd0_1: $n < d$
then
act0_1: $n := n + 1$
end

ML_out $\widehat{=}$ //concrete
when
grd1_1: $a + b < d$
grd1_2: $c = 0$
then
act1_1: $a := a + 1$
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

CentraleSupélec
université PARIS-SACLAY

INTUITION ABOUT REFINEMENT

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

```
ML_in ≙ //concrete
  when
    grd1_1: 0 < c
  then
    act1_1: c := c − 1
  end
```

- Same remarks as in the previous slide
- But this has to be confirmed by well-defined proof obligations

CentraleSupélec
université PARIS-SACLAY

PROOF OBLIGATIONS FOR REFINEMENT

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

CentraleSupélec
université PARIS-SACLAY

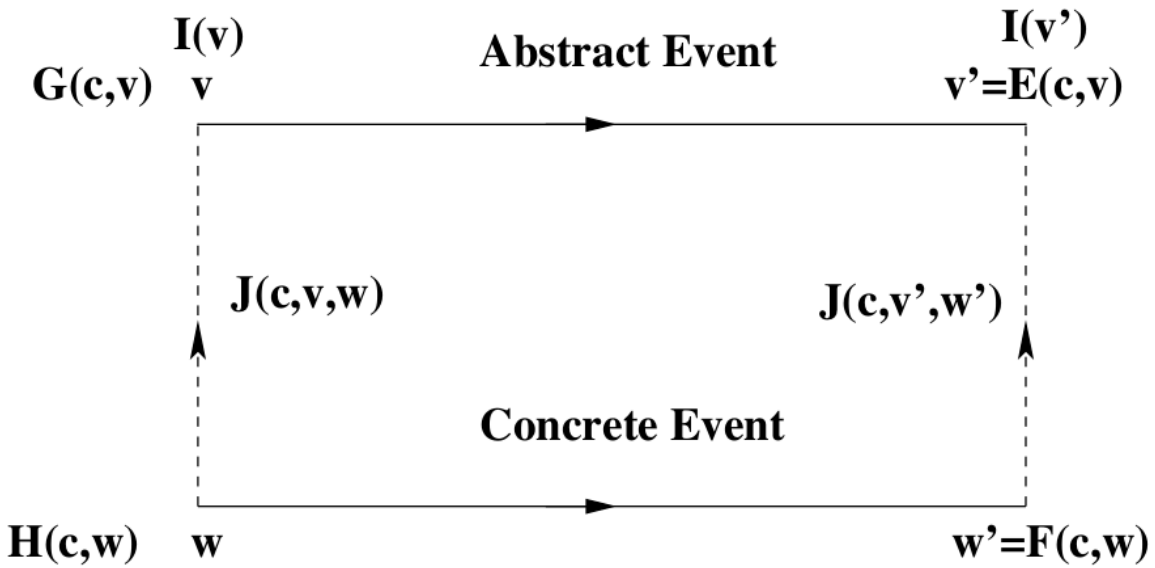
PROVING CORRECT REFINEMENT

THE SITUATION

- 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 $G(c, v) \longrightarrow G_1(c, v), G_2(c, v), \ldots$
- Abstract event with before-after predicate $v' = E(c, v)$
- Concrete event with guards $H(c, w)$ and b-a predicate $w' = F(c, w)$

CentraleSupélec
université PARIS-SACLAY

CORRECTNESS OF EVENT REFINEMENT



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)

CentraleSupélec
université PARIS-SACLAY

PROOF OBLIGATION

GUARD STRENGTHENING

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

CentraleSupélec
université PARIS-SACLAY

APPLYING GUARD STRENGTHENING TO EVENT ML_out

PROOF OF ML_out/GRD

ML_out $\widehat{=}$ //abstract
  when
    grd0_1: $n < d$
  then
    act0_1: $n := n + 1$
  end

ML_out $\widehat{=}$ //concrete
  when
    grd1_1: $a + b < d$
    grd1_2: $c = 0$
  then
    act1_1: $a := a + 1$
  end

CentraleSupélec
université PARIS-SACLAY

APPLYING GUARD STRENGTHENING TO EVENT ML_out

PROOF OF ML_out/GRD

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

MON $\Longrightarrow$

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

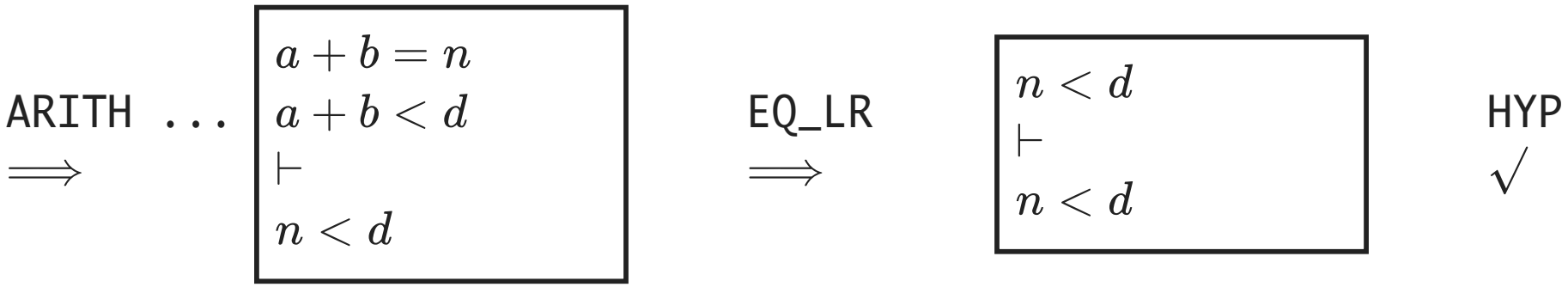
EQ_LR $\Longrightarrow$

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

ARITH ... $\Longrightarrow$

APPLYING GUARD STRENGTHENING TO EVENT ML_out

PROOF OF ML_out/GRD



CentraleSupélec
université PARIS-SACLAY

APPLYING GUARD STRENGTHENING TO EVENT ML_in

PROOF OF ML_in/GRD

ML_in $\widehat{=}$ //abstract
  when
    grd0_1: $0 < n$
  then
    act0_1: $n := n - 1$
  end

ML_in $\widehat{=}$ //concrete
  when
    grd1_1: $0 < c$
  then
    act1_1: $c := c - 1$
  end

CentraleSupélec
université PARIS-SACLAY

APPLYING GUARD STRENGTHENING TO EVENT ML_in

PROOF OF ML_in/GRD

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

MON $\Longrightarrow$

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

OR_L $\Longrightarrow$

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

EQ_LR ... $\Longrightarrow$

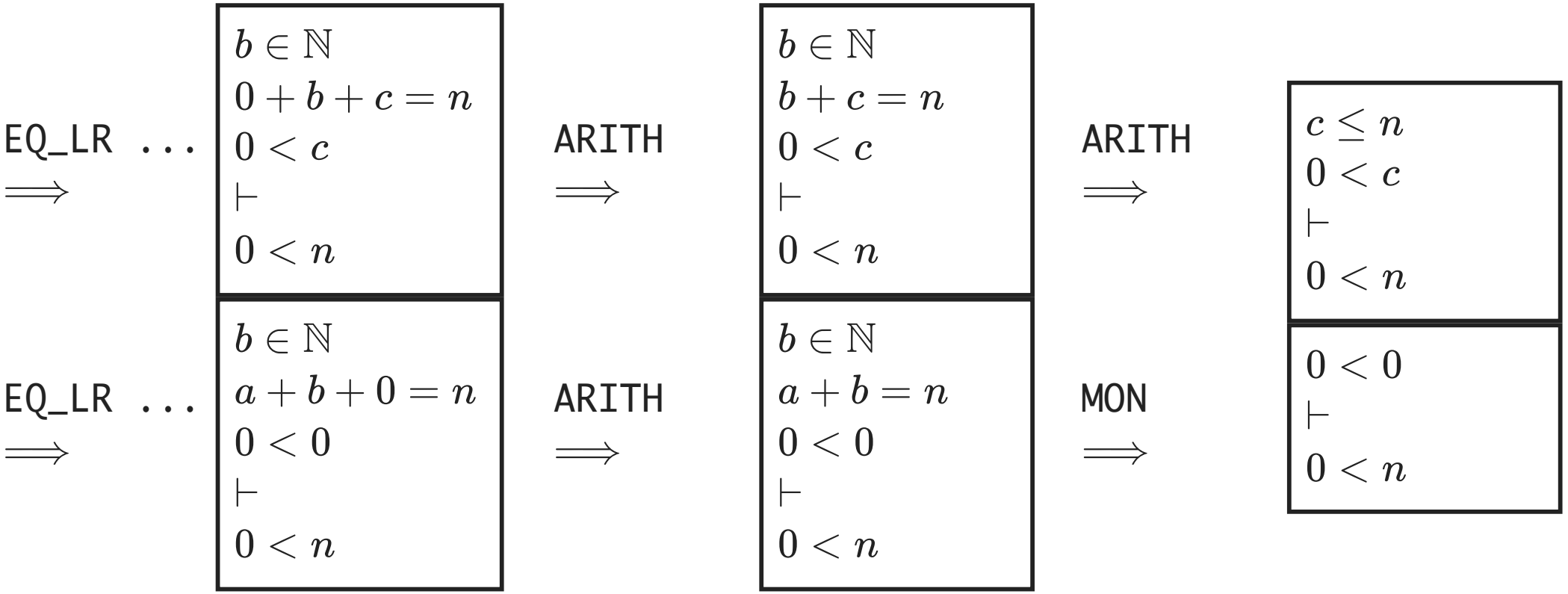
$$
\begin{array}{|l}
\hline
b \in \mathbb{N} \\
a + b + c = n \\
c = 0 \\
0 < c \\
\vdash \\
0 < n \\
\hline
\end{array}
$$

EQ_LR ... $\Longrightarrow$

CentraleSupélec
université PARIS-SACLAY

APPLYING GUARD STRENGTHENING TO EVENT ML_in

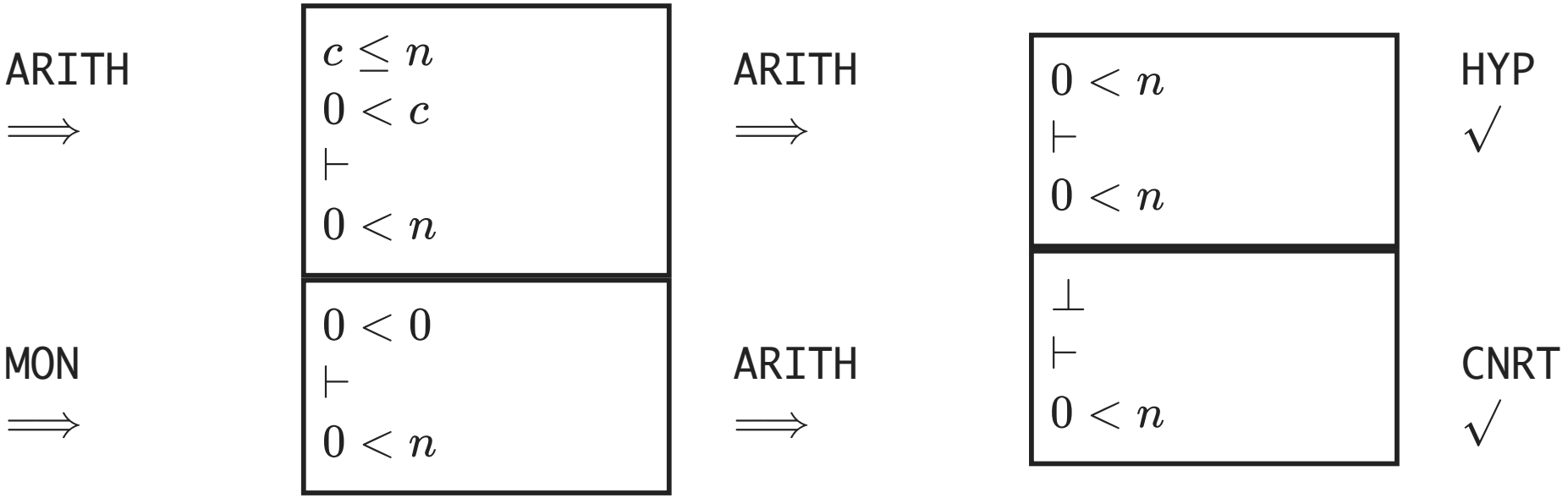
PROOF OF ML_in/GRD



CentraleSupélec
université PARIS-SACLAY

APPLYING GUARD STRENGTHENING TO EVENT ML_in

PROOF OF ML_in/GRD



- In the previous proof, we have used and additional inference rule
- It says that a false hypothesis entails any goal $\bot \vdash P$ CNTR

CentraleSupélec
université PARIS-SACLAY

PROOF OBLIGATION

INVARIANT REFINEMENT

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

CentraleSupélec
université PARIS-SACLAY

APPLYING INVARIANT REFINEMENT TO EVENT ML_out

PROOF OF ML_out/inv1_4/INV

ML_out $\widehat{=}$ //abstract
  when
    grd0_1: $n < d$
  then
    act0_1: $n := n + 1$
  end

ML_out $\widehat{=}$ //concrete
  when
    grd1_1: $a + b < d$
    grd1_2: $c = 0$
  then
    act1_1: $a := a + 1$
  end

CentraleSupélec
université PARIS-SACLAY

APPLYING INVARIANT REFINEMENT TO EVENT ML_out

PROOF OF ML_out/inv1_4/INV

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

MON $\Longrightarrow$

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

ARITH $\Longrightarrow$

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

CentraleSupélec
université PARIS-SACLAY

APPLYING INVARIANT REFINEMENT TO EVENT ML_out

PROOF OF ML_out/inv1_4/INV

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

EQ_LR

$\Longrightarrow$

$$\boxed{\begin{array}{l} \vdash \\ n+1=n+1 \end{array}}$$

EQL

$\surd$

CentraleSupélec

université PARIS-SACLAY

REFINING THE INITIALIZATION EVENT `init`

- Concrete initialization

  EVENTS

  `INITIALISATION` $\widehat{=}$

  $a := 0$

  $b := 0$

  $c := 0$

- Corresponding after predicate

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

CentraleSupélec
université PARIS-SACLAY

PROOF OBLIGATION

INITIALIZATION REFINEMENT

- 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 $w' = L(c)$

$$\begin{array}{lll} \text{Axioms} & A(c) & \\ \vdash & \vdash & \texttt{INV} \\ \text{Modified concrete invariants} & J_j(c, K(c), L(c)) & \end{array}$$

CentraleSupélec
université PARIS-SACLAY

PROOF OBLIGATION

INITIALIZATION REFINEMENT

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

$$
\begin{array}{l}
d \in \mathbb{N} \\
0 < d \\
\vdash \\
0 = 0 \lor 0 = 0 \\
(a = 0 \lor c = 0)
\end{array}
$$

CentraleSupélec
université PARIS-SACLAY

PLAN

Retour au plan - Retour à l'accueil

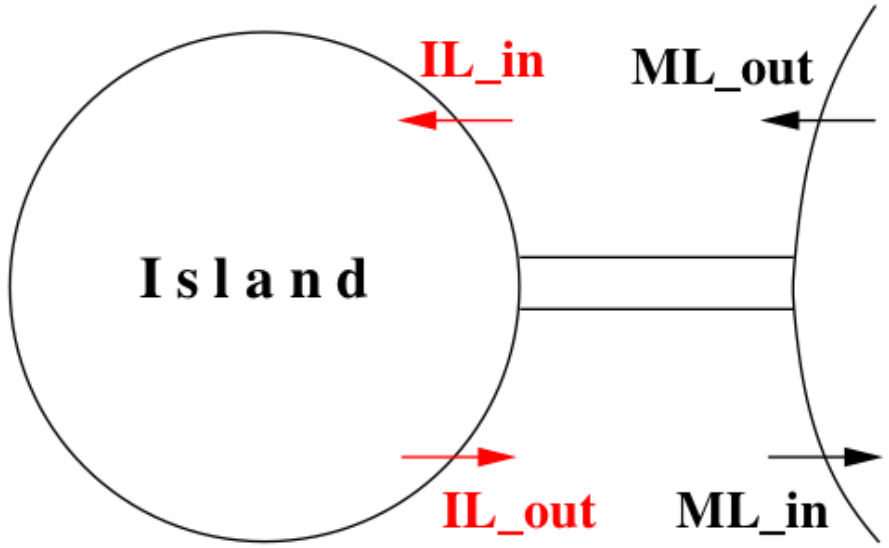
CentraleSupélec
université PARIS-SACLAY

ADDING NEW EVENTS

- 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

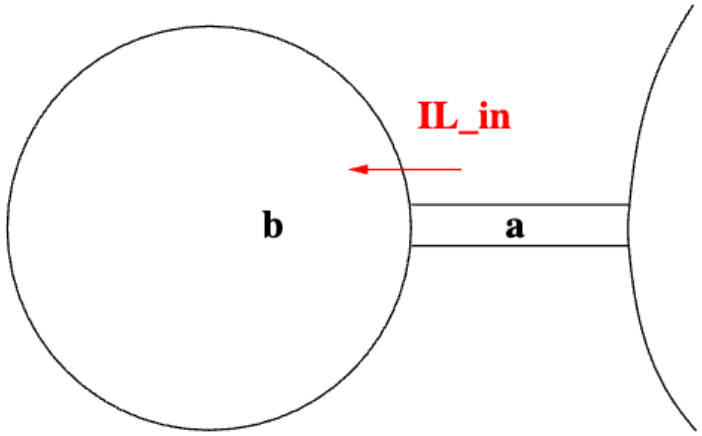
CentraleSupélec
université PARIS-SACLAY

ADDING NEW EVENTS



CentraleSupélec
université PARIS-SACLAY

NEW EVENT IL_in



IL_in $\widehat{=}$

when

grd1_1: $0 < a$

then

act1_1: $a := a - 1$

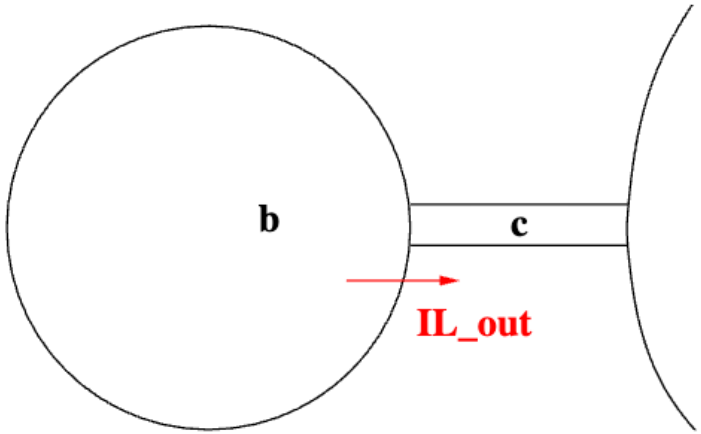
act1_2: $b := b + 1$

end

CentraleSupélec

université PARIS-SACLAY

NEW EVENT IL_out



IL_out $\widehat{=}$
  when
    grd1_1: $0 < b$
    grd1_2: $a = 0$
  then
    act1_1: $b := b - 1$
    act1_2: $c := c + 1$
  end

CentraleSupélec
université PARIS-SACLAY

BEFORE-AFTER PREDICATES

IL_in $\widehat{=}$
  when
    grd1_1: $0 < a$
  then
    act1_1: $a := a - 1$
    act1_2: $b := b + 1$
  end

IL_out $\widehat{=}$
  when
    grd1_1: $0 < b$
    grd1_2: $a = 0$
  then
    act1_1: $b := b - 1$
    act1_2: $c := c + 1$
  end

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

CentraleSupélec
université PARIS-SACLAY

THE EMPTY ASSIGNMENT `skip`

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

$$n' = n$$

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

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

- The guard of the `skip` event is `true`.

CentraleSupélec
université PARIS-SACLAY

REFINEMENT PROOF OBLIGATIONS FOR NEW EVENTS

- A new event must refine an implicit event, made of a `skip` action
  - Guard strengthening is trivial
  - Need to prove invariant refinement
- 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

EVENT IL_in REFINES skip

IL_in $\widehat{=}$

when

grd1_1: $0 < a$

then

act1_1: $a := a - 1$

act1_2: $b := b + 1$

end

CentraleSupélec

université PARIS-SACLAY

PROOF OF IL_in/inv1_4/INV

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

MON $\Longrightarrow$

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

ARITH $\Longrightarrow$

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

HYP $\surd$

CentraleSupélec

université PARIS-SACLAY

PROOF OBLIGATION

CONVERGENCE OF NEW EVENTS

- Axioms $A(c)$, invariants $I(c, v)$, concrete invariant $J(c, v, w)$
- New event with guard $H(c, w)$
- Variant $V(c, w)$

Axioms	$A(c)$	
$\text{Abstract Invariants}$	$I(c, v)$	
$\text{Concrete Invariants}$	$J(c, v, w)$	NAT
Concrete Guard	$H(c, w)$	
$\vdash$	$\vdash$	
$\text{Variant} \in \mathbb{N}$	$V(c, w) \in \mathbb{N}$	

CentraleSupélec
université PARIS-SACLAY

PROOF OBLIGATION

CONVERGENCE OF NEW EVENTS

- Axioms $A(c)$, invariants $I(c, v)$, concrete invariant $J(c, v, w)$
- New event with guard $H(c, w)$ and b-a predicate $w' = F(c, w)$
- Variant $V(c, w)$

Axioms	$A(c)$	
$\text{Abstract Invariants}$	$I(c, v)$	
$\text{Concrete Invariants}$	$J(c, v, w)$	VAR
Concrete Guard	$H(c, w)$	
$\vdash$	$\vdash$	
$\text{Modified Var} < \text{Var}$	$V(c, F(c, w)) < V(c, w)$	

CentraleSupélec
université PARIS-SACLAY

PROPOSED VARIANT

Weighted sum of a and b

VARIANT

variant_1: $2 \times a + b$

CentraleSupélec

université PARIS-SACLAY

DECREASING OF THE VARIANT BY EVENT IL_in

IL_in $\widehat{=}$
  when
    grd1_1: $0 < a$
  then
    act1_1: $a := a - 1$
    act1_2: $b := b + 1$
  end

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

CentraleSupélec
université PARIS-SACLAY

DECREASING OF THE VARIANT BY EVENT IL_out

IL_out $\widehat{=}$
  when
    grd1_1: $0 < b$
    grd1_2: $a = 0$
  then
    act1_1: $b := b - 1$
    act1_2: $c := c + 1$
  end

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

CentraleSupélec
université PARIS-SACLAY

PLAN

Retour au plan - Retour à l'accueil

CentraleSupélec
université PARIS-SACLAY

SUPERPOSITION
(NOT COVERED IN THIS TUTORIAL)

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

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

Axioms
Abstract Invariants
Concrete Invariants
Concrete Guard
$\vdash$
Same actions in common variables

$A(c)$
$I(c, v)$
$J(c, v, w)$
$H(c, w)$
$\vdash$
$M(c, u, v) = N(c, v, w)$

SIM

CentraleSupélec
université PARIS-SACLAY

MERCI

Version PDF des slides

Retour à l'accueil - Retour au plan

CentraleSupélec
université PARIS-SACLAY