

# Modelling Software-based Systems

## Lecture 2

### Proof Obligation Generation

Master Informatique

Dominique Méry  
Telecom Nancy, Université de Lorraine

24 novembre 2025  
dominique.mery@loria.fr

# General Summary

- 1 Overview of machines, contexts and proof obligations
- 2 Proof Obligations for Contexts and Machines

PO thm/THM (context)

PO thm/THM (machine)

PO evt/inv/INV

PO evt/act/FIS

PO evt/NAT

PO NAT

PO evt/VAR (arithmetic)

PO evt/VAR (set-theoretic)

- 3 Proof Obligations for Refinement

PO evt/grd/GRD

PO evt/act/SIM

PO evt/NAT

PO NAT

PO evt/VAR (arithmetic)

PO evt/VAR (set-theoretic)

PO evt/x/WFIS

- 1 Overview of machines, contexts and proof obligations
- 2 Proof Obligations for Contexts and Machines
- 3 Proof Obligations for Refinement

# Analysis of the Event-B Models



MACHINE

$m$

REFINES

$am$

SEES

$c$

VARIABLES

$u$

INVARIANTS

$I(s, c, u)$

THEOREMS

$Q(s, c, u)$

VARIANT

$exp(s, c, u)$

EVENTS

INITIALIZATION

...

$e$

...

END

MACHINE

$m$

REFINES

$am$

SEES

$c$

VARIABLES

$u$

INVARIANTS

$I(s, c, u)$

THEOREMS

$Q(s, c, u)$

VARIANT

$exp(s, c, u)$

EVENTS

INITIALIZATION

...

$e$

...

END

- $\Gamma(m)$  : environment for the machine  $m$  defined by the context  $c$  and it provides a list of seen axioms  $Ax(s, c)$  and a list of seen theorems  $Th(s, c)$  for the sets  $s$  and constants  $c$ .
- $\Gamma(m) \vdash \forall u. \text{INIT}(s, c, u) \Rightarrow I(s, c, u)$
- For each event  $e$  in  $E$  :  
 $\Gamma(m) \vdash \forall u, u'. I(s, c, u) \wedge BA(e)(u, u') \Rightarrow I(u')$
- For each event  $e$  in  $E$  :  
 $\Gamma(m) \vdash \forall u. I(s, c, u) \wedge GRD(e)(s, c, u) \Rightarrow \exists u'. BA(e)(u, u')$
- $\Gamma(m) \vdash \forall u. I(s, c, u) \Rightarrow Q(s, c, u)$
- Generated proof obligations are derived from those conditions.

# Three kinds of events

Events are divided into three kinds of events :

- An event is **ordinary** and, when it is observed, it modifies variables according to a guard and an action.
- An event is **anticipated** and, when it is observed, it means that something is observed but later in the further refinement.
- An event is **convergent** and, when it is observed, it decreases a variant which is member of naturals or is a set..





- ① Overview of machines, contexts and proof obligations
- ② Proof Obligations for Contexts and Machines
- ③ Proof Obligations for Refinement

# PO thm/THM (context)

CONTEXTS

$c$   
EXTENDS

$ac$   
SETS

$s$   
CONSTANTS

$c$   
AXIOMS

$Ax(s, c)$   
THEOREMS

$th_1 : P_1(s, c)$

...

$th_n : P_n(s, c)$

$th : P(s, c)$

...

END

$s$	<i>seen sets</i>
$c$	<i>seen constants</i>
$Ax(s, c)$	<i>seen axioms</i>
$Th(s, c)$	<i>previous proved theorems</i>
	$Th(s, c) = \{P_i(s, c)   i 1..n\}$
$P(s, c)$	<i>property over <math>s</math> and <math>c</math></i>

**PO** th/THM

$Ax(s, c), Th(s, c) \vdash P(s, c)$

# PO thm/THM (machine)

MACHINE

$m$

...  
VARIABLES

$u$

INVARIANTS

$I(s, c, u)$

THEOREMS

$Q(s, c, u)$

$th : P(s, c, u)$

...  
END

$s$

*seen sets*

$c$

*seen constants*

$u$

*variables*

$Ax(s, c)$

*seen axioms*

$Th(s, c)$

*seen theorems*

$I(s, c, u)$

*invariants*

$Q(s, c, u)$

*theorems*

$P(s, c, u)$

*property over  $s, c$  and  $u$*

**PO** th/THM

$Ax(s, c), Th(s, c), I(s, c, u) \vdash P(s, c, u)$

# PO evt/inv/INV

```
EVENT evt
  ANY x WHERE
    G(x, s, c, u)
  THEN
    u : |BAP(x, s, c, u, u')
  END
```

$BA(\text{evt}) \hat{=}$

$\exists x. \left( \begin{array}{l} \wedge G(x, s, c, u) \\ \wedge BAP(x, s, c, u, u') \end{array} \right)$

$GRD(\text{evt}) \hat{=} G(x, s, c, u)$

$ACT(\text{evt}) \hat{=} BAP(x, s, c, u, u')$

$s$

$c$

$u$

$Ax(s, c)$

$Th(s, c)$

$I(s, c, u)$

$Q(s, c, u)$

evt

x

$G(x, s, c, u)$

$BAP(x, s, c, u, u')$

$inv : inv(s, c, u')$

*seen sets*

*seen constants*

*variables*

*seen axioms*

*seen theorems*

*invariants*

*theorems*

*event name*

*event parameter*

*event guard*

*event before-after predicate*

*specific modified invariant*

**PO** evt/inv/INV

$Ax(s, c), Th(s, c), I(s, c, u), G(x, s, c, u), BAP(x, s, c, u, u') \vdash inv(s, c, u')$

**EVENT** evt  
**ANY**  $x$  **WHERE**  
     $G(x, s, c, u)$   
**THEN**  
     $u : |BAP(x, s, c, u, u')$   
**END**

$BA(\text{evt}) \hat{=}$

$\left( \begin{array}{l} \wedge G(x, s, c, u) \\ \wedge BAP(x, s, c, u, u') \end{array} \right)$

$GRD(\text{evt}) \hat{=} G(x, s, c, u)$

$ACT(\text{evt}) \hat{=}$

$BAP(x, s, c, u, u')$

$s$

$c$

$u$

$Ax(s, c)$

$Th(s, c)$

$I(s, c, u)$

$Q(s, c, u)$

evt

$x$

$G(x, s, c, u)$

$BAP(x, s, c, u, u')$

*seen sets*

*seen constants*

*variables*

*seen axioms*

*seen theorems*

*invariants*

*theorems*

*event name*

*event parameter*

*event guard*

*event before-after predicate*

**PO** evt/act/FIS

$Ax(s, c), Th(s, c), I(s, c, u), G(x, s, c, u), \vdash \exists u'. BAP(x, s, c, u, u')$

```

EVENT ae
  ANY x WHERE
    G(x, s, c, u)
  THEN
    u : |BAP(x, s, c, u, u')
  END
...
VARIANT
  exp(s, c, u)
    
```

$s$

$c$

$u$

$Ax(s, c)$

$Th(s, c)$

$I(s, c, u)$

$Q(s, c, u), R(s, c, u, v)$

evt, ce

$x$

$G(x, s, c, u)$

$BAP(x, s, c, u, u')$

$exp(s, c, u)$

*seen sets*

*seen constants*

*abstract variables*

*seen axioms*

*seen theorems*

*abstract invariants*

*abstract and concrete theorems*

*event name*

*event parameters*

*abstract event guard*

*event before-after predicate*

*arithmetic expression*

**PO** evt/NAT  $Ax(s, c), Th(s, c), I(s, c, u), G(x, s, c, u) \vdash exp(s, c, u) \in \mathbb{N}$

```
EVENT ae
  ANY x WHERE
     $G(x, s, c, u)$ 
  THEN
     $u : |BAP(x, s, c, u, u')$ 
  END
...
VARIANT
   $exp(s, c, u)$ 
```

$s$   
 $c$   
 $u, v$   
 $Ax(s, c)$   
 $Th(s, c)$   
 $I(s, c, u)$   
 $Q(s, c, u), R(s, c, u, v)$   
evt, ce  
 $x$   
 $G(x, s, c, u)$   
 $BAP(x, s, c, u, u')$   
 $setexp(s, c, u)$

*seen sets*  
*seen constants*  
*abstract variables*  
*seen axioms*  
*seen theorems*  
*abstract invariants*  
*abstract and concrete theorems*  
*event name*  
*event parameters*  
*abstract event guard*  
*event before-after predicate*  
*set expression*

## PO evt/NAT

$Ax(s, c), Th(s, c), I(s, c, u), G(x, s, c, u) \vdash finite(setexp(s, c, u))$

```
EVENT ae
  ANY x WHERE
     $G(x, s, c, u)$ 
  THEN
     $u : |BAP(x, s, c, u, u')$ 
  END
...
VARIANT
   $exp(s, c, u)$ 
```

$s$

$c$

$u, v$

$Ax(s, c)$

$Th(s, c)$

$I(s, c, u)$

evt

$x$

$G(x, s, c, u)$

$BAP(x, s, c, u, u')$

$exp(s, c, u)$

*seen sets*

*seen constants*

*abstract and concrete variables*

*seen axioms*

*seen theorems*

*abstract invariants*

*event name*

*event parameters*

*abstract event guard*

*event before-after predicate*

*arithmetic expression*

## PO evt/VAR

$Ax(s, c), Th(s, c), I(s, c, u), G(x, s, c, u), BAP(x, s, c, u, u') \vdash$   
 $exp(s, c, u') < exp(s, c, u)$



```
EVENT ae
  ANY x WHERE
     $G(x, s, c, u)$ 
  THEN
     $u : |BAP(x, s, c, u, u')$ 
  END
...
VARIANT
   $setexp(s, c, u)$ 
```

$s$

$c$

$u, v$

$Ax(s, c)$

$Th(s, c)$

$I(s, c, u)$

evt

x

$G(x, s, c, u)$

$BAP(x, s, c, u, u')$

$setexp(s, c, u)$

*seen sets*

*seen constants*

*abstract and concrete variables*

*seen axioms*

*seen theorems*

*abstract invariants*

*event name*

*event parameters*

*abstract event guard*

*event before-after predicate*

*set-theoretic expression*

## PO evt/VAR

$Ax(s, c), Th(s, c), I(s, c, u), G(x, s, c, u), BAP(x, s, c, u, u') \vdash$   
 $setexp(s, c, u') \subset setexp(s, c, u)$

- ① Overview of machines, contexts and proof obligations
- ② Proof Obligations for Contexts and Machines
- ③ Proof Obligations for Refinement

# PO evt/grd/GRD

```
EVENT ae
  ANY x WHERE
     $G(x, s, c, u)$ 
  THEN
     $u : |ABAP(x, s, c, u, u')$ 
  END
```

```
EVENT ce
  REFINES
    ae
  ANY y WHERE
     $H(y, s, c, v)$ 
  WITH
     $x : W(x, y, s, c, v)$ 
  THEN
     $v : |CBAP(y, s, c, v, v')$ 
  END
```

$s$   
 $c$   
 $u, v$   
 $Ax(s, c)$   
 $Th(s, c)$   
 $I(s, c, u)$   
 $J(s, c, u, v)$   
 $Q(s, c, u), R(s, c, u, v)$   
 $ae, ce$   
 $x, y$   
 $G(x, s, c, u)$   
 $H(y, s, c, v)$   
 $ABAP(x, s, c, u, u')$   
 $CBAP(x, s, c, u, u')$   
 $W(x, y, s, c, v)$

*seen sets*

*seen constants*

*abstract and concrete variables*

*seen axioms*

*seen theorems*

*abstract invariants*

*concrete invariants*

*abstract and concrete theorems*

*abstract and concrete event names*

*event parameters*

*abstract event guard*

*concrete event guard*

*abstract event before-after predicate*

*concrete event before-after predicate*

*witness predicate*

## PO evt/grd/GRD

$Ax(s, c), Th(s, c), I(s, c, u), J(s, c, u, v), W(x, y, s, c, v), H(y, s, c, v), \vdash$   
 $G(x, s, c, u, u')$

# PO evt/act/SIM

**EVENT** ae

**ANY**  $x$  **WHERE**

$G(x, s, c, u)$

**THEN**

$u : |ABAP(x, s, c, u, u')$

**END**

**EVENT** ce

**REFINES**

ae

**ANY**  $y$  **WHERE**

$H(y, s, c, v)$

**WITH**

$x : WP(x, y, s, c, v)$

$u' : WV(y, u', s, c, v)$

**THEN**

$v : |CBAP(y, s, c, v, v')$

**END**

$s$

$c$

$u, v$

$Ax(s, c)$

$Th(s, c)$

$I(s, c, u)$

$J(s, c, u, v)$

$Q(s, c, u), R(s, c, u, v)$

ae, ce

$x, y$

$G(x, s, c, u)$

$H(y, s, c, v)$

$ABAP(x, s, c, u, u')$

$CBAP(x, s, c, u, u')$

$WP(x, y, s, c, v)$

$WV(y, u', s, c, v)$

*seen sets*

*seen constants*

*abstract and concrete variables*

*seen axioms*

*seen theorems*

*abstract invariants*

*concrete invariants*

*abstract and concrete theorems*

*abstract and concrete event name*

*event parameters*

*abstract event guard*

*concrete event guard*

*abstract event before-after predic*

*concrete event before-after predic*

*witness parameter predicate*

*witness variable predicate*

**PO** evt/act/SIM

$$\left( \begin{array}{l} Ax(s, c), Th(s, c), I(s, c, u), J(s, c, u, v) \\ WP(x, y, s, c, v), WV(y, u', s, c, v) \\ H(y, s, c, v), CBAP(y, s, c, v, v') \end{array} \right) \vdash ABAP(x, s, c, u, u')$$

```
EVENT ae
  ANY x WHERE
    G(x, s, c, u)
  THEN
    u : |BAP(x, s, c, u, u')
  END
...
VARIANT
  exp(s, c, u)
```

$s$   
 $c$   
 $u, v$   
 $Ax(s, c)$   
 $Th(s, c)$   
 $I(s, c, u)$   
 $J(s, c, u, v)$   
 $Q(s, c, u), R(s, c, u, v)$   
evt, ce  
 $x$   
 $G(x, s, c, u)$   
 $BAP(x, s, c, u, u')$   
 $exp(s, c, u)$

*seen sets*  
*seen constants*  
*abstract and concrete variables*  
*seen axioms*  
*seen theorems*  
*abstract invariants*  
*concrete invariants*  
*abstract and concrete theorems*  
*event name*  
*event parameters*  
*abstract event guard*  
*event before-after predicate*  
*arithmetic expression*

## PO evt/NAT

$Ax(s, c), Th(s, c), I(s, c, u), J(s, c, u, v), G(x, s, c, u) \vdash exp(s, c, u) \in \mathbb{N}$

```

EVENT ae
  ANY x WHERE
     $G(x, s, c, u)$ 
  THEN
     $u : |BAP(x, s, c, u, u')$ 
  END
...
VARIANT
   $exp(s, c, u)$ 
    
```

$s$

$c$

$u, v$

$Ax(s, c)$

$Th(s, c)$

$I(s, c, u)$

$J(s, c, u, v)$

$Q(s, c, u), R(s, c, u, v)$

evt, ce

$x$

$G(x, s, c, u)$

$BAP(x, s, c, u, u')$

$setexp(s, c, u)$

*seen sets*

*seen constants*

*abstract and concrete variables*

*seen axioms*

*seen theorems*

*abstract invariants*

*concrete invariants*

*abstract and concrete theorems*

*event name*

*event parameters*

*abstract event guard*

*event before-after predicate*

*set expression*

**PO** evt/NAT  $Ax(s, c), Th(s, c), I(s, c, u), J(s, c, u, v), G(x, s, c, u) \vdash$   
 $finite(setexp(s, c, u))$

```

EVENT ae
  ANY x WHERE
    G(x, s, c, u)
  THEN
    u : |BAP(x, s, c, u, u')
  END
...
VARIANT
  exp(s, c, u)
    
```

$s$

$c$

$u, v$

$Ax(s, c)$

$Th(s, c)$

$I(s, c, u)$

$J(s, c, u, v)$

$Q(s, c, u), R(s, c, u, v)$

evt, ce

$x$

$G(x, s, c, u)$

$BAP(x, s, c, u, u')$

$exp(s, c, u)$

*seen sets*

*seen constants*

*abstract and concrete variables*

*seen axioms*

*seen theorems*

*abstract invariants*

*concrete invariants*

*abstract and concrete theorems*

*event name*

*event parameters*

*abstract event guard*

*event before-after predicate*

*arithmetic expression*

## PO evt/VAR

$Ax(s, c), Th(s, c), I(s, c, u), J(s, c, u, v), G(x, s, c, u), BAP(x, s, c, u, u') \vdash$   
 $exp(s, c, u') < exp(s, c, u)$

```

EVENT ae
  ANY x WHERE
    G(x, s, c, u)
  THEN
    u : |BAP(x, s, c, u, u')
  END
...
VARIANT
  setexp(s, c, u)
    
```

$s$   
 $c$   
 $u, v$   
 $Ax(s, c)$   
 $Th(s, c)$   
 $I(s, c, u)$   
 $J(s, c, u, v)$   
 $Q(s, c, u), R(s, c, u, v)$   
 evt, ce  
 $x$   
 $G(x, s, c, u)$   
 $BAP(x, s, c, u, u')$   
 $setexp(s, c, u)$

*seen sets*  
*seen constants*  
*abstract and concrete variables*  
*seen axioms*  
*seen theorems*  
*abstract invariants*  
*concrete invariants*  
*abstract and concrete theorems*  
*event name*  
*event parameters*  
*abstract event guard*  
*event before-after predicate*  
*set-theoretic expression*

## PO evt/VAR

$Ax(s, c), Th(s, c), I(s, c, u), J(s, c, u, v), G(x, s, c, u), BAP(x, s, c, u, u') \vdash$   
 $setexp(s, c, u') \subset setexp(s, c, u)$



```

EVENT ae
  ANY x WHERE
     $G(x, s, c, u)$ 
  THEN
     $u : |ABAP(x, s, c, u, u')$ 
  END
    
```

```

EVENT ce
  REFINES
    ae
  ANY y WHERE
     $H(y, s, c, v)$ 
  WITH
     $x : WP(x, y, s, c, v)$ 
     $u' : WV(y, u', s, c, v)$ 
  THEN
     $v : |CBAP(y, s, c, v, v')$ 
  END
    
```

$s$   
 $c$   
 $u, v$   
 $Ax(s, c)$   
 $Th(s, c)$   
 $I(s, c, u)$   
 $J(s, c, u, v)$   
 $Q(s, c, u), R(s, c, u, v)$   
 $ae, ce$   
 $x, y$   
 $G(x, s, c, u)$   
 $H(y, s, c, v)$   
 $ABAP(x, s, c, u, u')$   
 $CBAP(x, s, c, u, u')$   
 $WP(x, y, s, c, v)$   
 $WV(y, u', s, c, v)$

*seen sets*

*seen constants*

*abstract and concrete variables*

*seen axioms*

*seen theorems*

*abstract invariants*

*concrete invariants*

*abstract and concrete theorems*

*abstract and concrete event names*

*event parameters*

*abstract event guard*

*concrete event guard*

*abstract event before-after predicate*

*concrete event before-after predicate*

*witness parameter predicate*

*witness variable predicate*

**PO** evt/x/WFIS

$Ax(s, c), Th(s, c), I(s, c, u), J(s, c, u, v), H(y, s, c, v) \vdash$   
 $\exists x. WP(x, y, s, c, v)$