



# Modelling Software-based Systems Proof Obligation Generation

Telecom Nancy (IL et LE)

Dominique Méry Telecom Nancy, Université de Lorraine

18 septembre 2024 dominique.mery@loria.fr

# General Summary

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

PO thm/THM (context)

PO th/THM (machine)

PO evt/inv/INV

PO evt/act/FIS

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

# **Current Summary**

- 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



#### Machines en Event B

```
MACHINE
 m
REFINES
 am
SEES
 c
VARIABLES
INVARIANTS
 I(s, c, u)
THÈOREMS
 Q(s, c, u)
VARIANT
 exp(s, c, u)
EVENTS
 INITIALIZATION
END
```

#### Machines en Event B

```
MACHINE
REFINES
 am
SEES
 c
VARIABLES
INVARIANTS
 I(s, c, u)
THEOREMS
 Q(s,c,u)
VARIANT
 exp(s, c, u)
EVENTS
 INITIALIZATION
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 :  $\Gamma(m) \vdash \forall u, u'. I(s, c, u) \land BA(e)(u, u') \Rightarrow I(u')$
- For each event e in E:  $\Gamma(m) \vdash \forall u. I(s, c, u) \land GRD(e)(s, c, u) \Rightarrow \exists u'. BA(e)(u, u')$
- $\Gamma(m) \vdash \forall u. I(s, c, u) \Rightarrow Q(s, c, u)$
- Generated proof obligations are derived from those conditions.

# Checking the well formation of Event-B expressions

- Event-B expressions are contexts, machines, properties, equations, set-theoretical expressions . . .
- e is an Event-B expression and wd(e) is a logical property expressing the well definition of e.
- $\operatorname{wd}(1=2) \stackrel{\triangle}{=} \operatorname{wd}(1) \wedge \operatorname{wd}(2)$
- $\operatorname{wd}(a/b) \stackrel{\triangle}{=} b \neq 0 \land \operatorname{wd}(a) \land \operatorname{wd}(b)$
- $\bullet \ \operatorname{wd}(f(g)) \stackrel{\triangle}{=} g \in dom(f) \land f \in A \nrightarrow B$

# **Current Summary**

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

#### PO th/THM

```
\begin{array}{c} \textbf{CONTEXTS} \\ c \\ \textbf{EXTENDS} \\ ac \\ \textbf{SETS} \\ s \\ \textbf{CONSTANTS} \\ c \\ \textbf{AXIOMS} \\ Ax(s,c) \\ \textbf{THEOREMS} \\ th_1: P_1(s,c) \\ \dots \\ th_n: P_n(s,c) \\ th: P(s,c) \\ \dots \\ \textbf{END} \end{array}
```

```
\begin{array}{lll} s & \textit{seen sets} \\ c & \textit{seen constants} \\ Ax(s,c) & \textit{seen axioms} \\ Th(s,c) & \textit{previous proved theorems} \\ & PTh(s,c) = \{P_i(s,c)|i \ 1..n\} \\ P(s,c) & \textit{property over s and c} \end{array}
```

```
PO th/THM
```

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

# PO th/THM (machine)

```
\begin{array}{l} \textbf{MACHINE} \\ m \\ \dots \\ \textbf{VARIABLES} \\ u \\ \textbf{INVARIANTS} \\ I(s,c,u) \\ \textbf{THEOREMS} \\ Q(s,c,u) \\ th: P(s,c,u) \\ \dots \\ \textbf{END} \end{array}
```

```
\begin{array}{lll} 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 \end{array}
```

#### PO th/THM

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

#### PO evt/inv/INV

```
EVENT evt  \begin{array}{c} \textbf{ANY } x \ \textbf{WHERE} \\ G(x,s,c,u) \\ \textbf{THEN} \\ u:|BAP(x,s,c,u,u') \\ \textbf{END} \end{array}
```

$$\begin{array}{l} BA(\mathsf{evt}) \ \widehat{=} \\ \exists x. \left( \begin{array}{l} \land \ G(x,s,c,u) \\ \land \ BAP(x,s,c,u,u') \end{array} \right) \\ GRD(\mathsf{evt}) \ \widehat{=} \ G(x,s,c,u) \\ ACT(\mathsf{evt}) \ \widehat{=} \ BAP(x,s,c,u,u') \end{array}$$

```
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

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

$$\textbf{PO} \ \ \mathsf{Q}/\mathsf{THM} \ \ Ax(s,c), Th(s,c), I(s,c,u) \vdash Q(s,c,u)$$

# PO evt/act/FIS

```
\begin{aligned} & \textbf{EVENT evt} \\ & \textbf{ANY } x \textbf{ WHERE} \\ & G(x,s,c,u) \\ & \textbf{THEN} \\ & u:|BAP(x,s,c,u,u') \\ & \textbf{END} \end{aligned}
```

$$BA(\mathsf{evt}) \stackrel{\cong}{=} \\ \left( \begin{array}{c} \wedge \ G(x,s,c,u) \\ \wedge \ BAP(x,s,c,u,u') \end{array} \right) \\ GRD(\mathsf{evt}) \stackrel{\cong}{=} \ G(x,s,c,u) \\ ACT(\mathsf{evt}) \stackrel{\cong}{=} \\ BAP(x,s,c,u,u') \\ \end{array}$$

```
seen sets
s
                    seen constants
                     variables
u
Ax(s,c)
                    seen axioms
Th(s,c)
                    seen theorems
I(s,c,u)
                    invariants
Q(s,c,u)
                     theorems
                    event name
evt
                    event parameter
G(x, s, c, u) event guard
BAP(x, s, c, u, u') event before-after predicate
```

#### PO evt/act/FIS

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

# **Current Summary**

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

# PO evt/grd/GRD

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

```
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 name
event parameters
abstract event guard
concrete event guard
abstract event before-after predic
concrete event before-after predic
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

```
seen sets
                            s
EVENT ae
                            c
                                                       seen constants
 ANY x WHERE
                                                       abstract and concrete variables
                            u, v
   G(x, s, c, u)
                            Ax(s,c)
                                                      seen axioms
 THEN
   u: |ABAP(x, s, c, u, u')|
                            Th(s,c)
                                                      seen theorems
 END
                            I(s,c,u)
                                                       abstract invariants
                            J(s,c,u,v)
                                                      concrete invariants
EVENT ce
                            Q(s,c,u), R(s,c,u,v)
                                                       abstract and concrete theorems
 REFINES
   ae
                                                       abstract and concrete event name
                            ae, ce
 ANY y WHERE
                            X, y
                                                      event parameters
   H(y, s, c, v)
 WITH
                            G(x,s,c,u)
                                                      abstract event guard
   x: WP(x, y, s, c, v)
                            H(y,s,c,v)
                                                      concrete event guard
   u':WV(y,u',s,c,v)
                            ABAP(x, s, c, u, u')
                                                       abstract event before-after predic
 THEN
   v: |CBAP(y, s, c, v, v')|
                            CBAP(x, s, c, u, u')
                                                      concrete event before-after predic
 END
                            WP(x, y, s, c, v)
                                                      witness parameter predicate
                            WV(y, u', s, c, v)
                                                      witness variable predicate
```

$$\begin{array}{c} \textbf{PO} \text{ evt/act/SIM} \\ \left( \begin{array}{c} Ax(s,c), Th(s,c), I(s,c,u), J(s,c,u,v) \\ WP(x,y,s,c,v), WV(y,u',s,c,v) \\ H(y,s,c,v), CBAP(y,s,c,v,v') \end{array} \right) \vdash ABAP(x,s,c,u,u') \\ \text{Telecom Nancy 2024-2025 (Dominique Méry)} \end{array}$$

# PO evt/act/SIM

```
EVENT ae  \begin{array}{c} \textbf{ANY } x \ \textbf{WHERE} \\ G(x,s,c,u) \\ \textbf{THEN} \\ u:|BAP(x,s,c,u,u') \\ \textbf{END} \\ \dots \\ \textbf{VARIANT} \\ exp(s,c,u) \end{array}
```

```
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
Х
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 aritthmetic expression

#### PO evt/NAT

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

#### PO evt/act/SIM

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

#### PO evt/VAR

```
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
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 aritthmetic expression

#### PO evt/VAR

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

#### PO evt/VAR

```
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
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

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

#### PO evt/x/WFIS

```
EVENT ae  \begin{array}{c} \textbf{ANY } x \ \textbf{WHERE} \\ G(x,s,c,u) \\ \textbf{THEN} \\ u:|ABAP(x,s,c,u,u') \\ \textbf{END} \end{array}
```

# EVENT CE REFINES $ae \\ ANY y \text{ WHERE} \\ H(y,s,c,v) \\ \text{WITH} \\ x: WP(x,y,s,c,v) \\ u': WV(y,u',s,c,v) \\ \text{THEN} \\ v: |CBAP(y,s,c,v,v')$

```
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/x/WFIS

**END** 

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