

# Summary of Event-B Proof Obligations

Jean-Raymond Abrial (ETHZ)

March 2008

- Prerequisite:

  - (1) Summary of **Mathematical Notation** (a quick review)

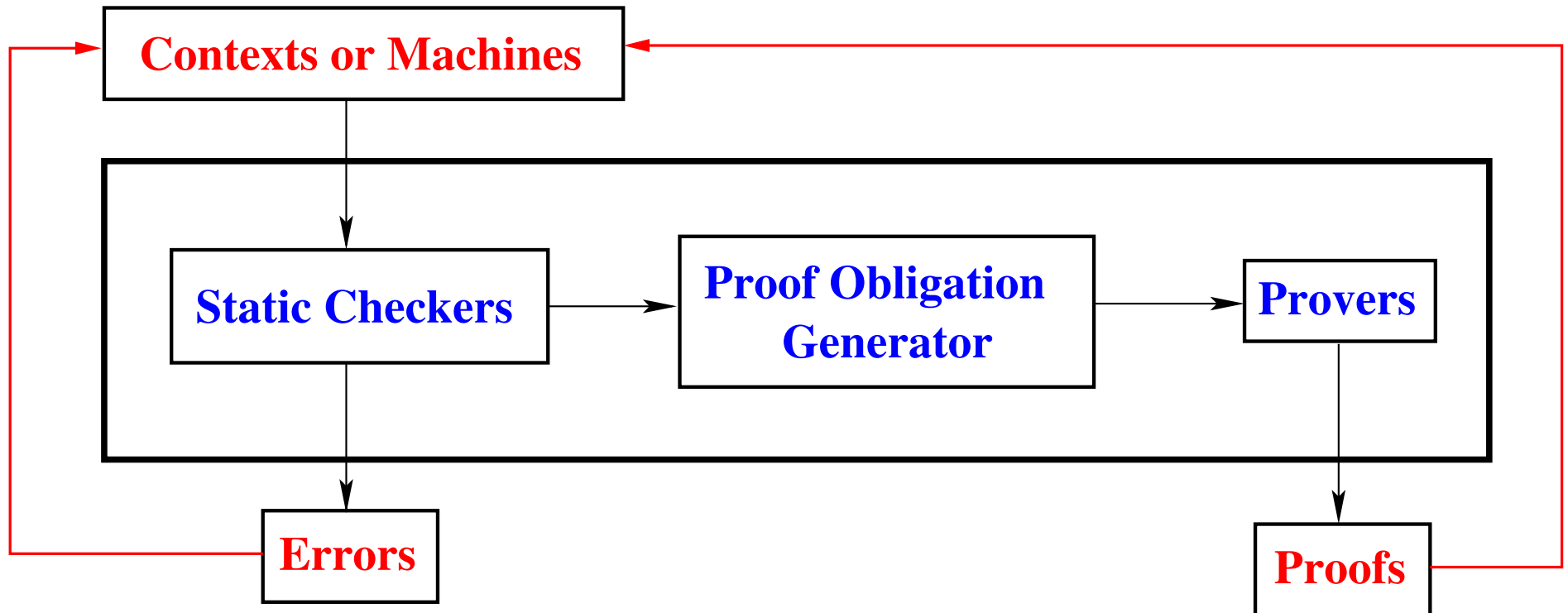
  - (2) Summary of **Event-B Notation**

    - Examples developed in (2) will be used here

- Showing the various Event-B **proof obligations**  
(sometimes also called **verification conditions**)

- The POs are **automatically** generated by a **Rodin Platform tool** called the **Proof Obligation Generator**
- This tool **static checks contexts** or **machine** texts
- It decides then **what is to be proved**
- The **outcome** are various **sequents**, which are transmitted to the **provers** performing **automatic** or **interactive** proofs

- The **Static Checkers**:
  - lexical analyser
  - syntactic analyser
  - type checker
- The **Proof Obligation Generator**
- The **Provers**



- Proofs which cannot be done help **improving the model**

- Invariant preservation (**INV** slide **8**)
- Non-deterministic action feasibility (**FIS** slide **13**)
- Guard strengthening in a refinement (**GRD** slide **17**)
- Simulation (**SIM** slide **21**)
- Numeric variant (**NAT** slide **25**)
- Set variant (**FIN** slide **29**)

- Variant decreasing (**VAR** slide **33**)
- Feasibility of a non-deterministic witness (**WFIS** slide **41**)
- Proving theorems (**THM** slide **45**)
- Well-definedness (**WD** slide **53**)
- Guard strengthening when merging abstract events (**MRG** slide **57**)

- Purpose and naming
- Formal definition
- Where generated in the "search" example
- Application to the example

- Ensuring that each invariant is preserved by each event.
- For an event "evt" and an invariant "inv" the name of this PO is:

evt/inv/INV

```

evt
  any  $x$  where
     $G(x, s, c, v)$ 
  then
     $v : | BAP(x, s, c, v, v')$ 
  end
    
```

$s$  : seen sets  
 $c$  : seen constants  
 $v$  : variables  
 $A(s, c)$  : seen axioms and thms  
 $I(s, c, v)$  : invariants and thms.  
 $evt$  : specific event  
 $x$  : event parameters  
 $G(x, s, c, v)$  : event guards  
 $BAP(x, s, c, v, v')$  : event before-after predicate  
 $inv(s, c, v')$  : modified specific invariant

<p>                     Axioms                      Invariants                      Guards of the event                      Before-after predicate of the event                      ⊢                      Modified Specific Invariant                 </p>	$evt/inv/INV$
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------

$A(s, c)$   
 $I(s, c, v)$   
 $G(x, s, c, v)$   
 $BAP(x, s, c, v, v')$   
 ⊢  
 $inv(s, c, v')$

- In case of the initialization event,  $I(s, c, v)$  is removed from the hypotheses

```

context
  ctx_0
sets
  D
constants
  n
  f
  v
axioms
  axm1 :  $n \in \mathbb{N}$ 
  axm2 :  $f \in 1..n \rightarrow D$ 
  axm3 :  $v \in \text{ran}(f)$ 
theorems
  axm1 :  $n \in \mathbb{N}1$ 
end
  
```

```

machine
  m_0a
sees
  ctx_0
variables
  i
invariants and thms.
  inv1 :  $i \in 1..n$ 
events
  ...
end
  
```

```

initialisation  $\hat{=}$ 
status
  ordinary
then
  act1 :  $i := 1$ 
end
  
```

```

search  $\hat{=}$ 
status
  ordinary
any
  k
where
  grd1 :  $k \in 1..n$ 
  grd2 :  $f(k) = v$ 
then
  act1 :  $i := k$ 
end
  
```

- Two invariant preservation POs are generated:
  - initialisation/inv1/INV
  - search/inv1/INV

**axm1**  
**axm2**  
**axm3**  
**thm1**  
BA predicate  
 $\vdash$   
modified **inv1**

$n \in \mathbb{N}$   
 $f \in 1..n \rightarrow D$   
 $v \in \text{ran}(f)$   
 $n \in \mathbb{N1}$   
 $i' = 1$   
 $\vdash$   
 $i' \in 1..n$

$n \in \mathbb{N}$   
 $f \in 1..n \rightarrow D$   
 $v \in \text{ran}(f)$   
 $n \in \mathbb{N1}$   
 $\vdash$   
 $1 \in 1..n$

Simplification performed  
by the PO Generator

**initialisation**  $\hat{=}$   
**status**  
**ordinary**  
**then**  
**act1** :  $i := 1$   
**end**

- Note that **inv1** is **not part of the hypotheses** (we are in the **initialisation** event)

```

axm1
axm2
axm3
thm1
inv1
grd1
grd2
BA predicate
┆
modified inv1
    
```

```

n ∈ ℕ
f ∈ 1..n → D
v ∈ ran(f)
n ∈ ℕ1
i ∈ 1..n
k ∈ 1..n
-----
f(k) = v
i' = k
┆
i' ∈ 1..n
    
```

```

n ∈ ℕ
f ∈ 1..n → D
v ∈ ran(f)
n ∈ ℕ1
i ∈ 1..n
k ∈ 1..n
-----
f(k) = v
┆
k ∈ 1..n
    
```

Simplification performed  
by the PO Generator

```

search ≙
  status
  ordinary
  any
  k
  where
    grd1 : k ∈ 1..n
    grd2 : f(k) = v
  then
    act1 : i := k
  end
    
```

- In what follows, we'll show the **simplified form** only

- Ensuring that each non-deterministic action is feasible.
- For an event "evt" and a non-deterministic action "act" in it, the name of this PO is:

evt/act/FIS

```

evt
  any  $x$  where
     $G(x, s, c, v)$ 
  then
     $v :| BAP(x, s, c, v, v')$ 
  end
    
```

$s$  : seen sets  
 $c$  : seen constants  
 $v$  : variables  
 $A(s, c)$  : seen axioms and thms  
 $I(s, c, v)$  : invariants and thms.  
 $evt$  : specific event  
 $x$  : event parameters  
 $G(x, s, c, v)$  : event guards  
 $BAP(x, s, c, v, v')$  : event action

Axioms Invariants Guards of the event $\vdash$ $\exists v' \cdot$ Before-after predicate	$evt/act/FIS$
------------------------------------------------------------------------------------------------------	---------------

$A(s, c)$   
 $I(s, c, v)$   
 $G(x, s, c, v)$   
 $\vdash$   
 $\exists v' \cdot BAP(x, s, c, v, v')$

```

context
  ctx_0
sets
  D
constants
  n
  f
  v
axioms
  axm1 :  $n \in \mathbb{N}$ 
  axm2 :  $f \in 1..n \rightarrow D$ 
  axm3 :  $v \in \text{ran}(f)$ 
theorems
  thm1 :  $n \in \mathbb{N}1$ 
end

```

```

machine
  m_0b
sees
  ctx_0
variables
  i
invariants and thms.
  inv1 :  $i \in 1..n$ 
events
  ...
end

```

```

initialisation  $\hat{=}$ 
  status
  ordinary
  then
  act1 :  $i := 1$ 
  end

```

```

search  $\hat{=}$ 
  status
  ordinary
  then
  act1 :  $i : | i' \in 1..n \wedge f(i') = v$ 
  end

```

- Among others, **one feasibility PO** is generated:
  - **search/act1/FIS**

axm1  
axm2  
axm3  
thm1  
inv1  
grd

⊢  
∃*i'* · before-after predicate

$n \in \mathbb{N}$   
 $f \in 1..n \rightarrow D$   
 $v \in \text{ran}(f)$

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

$i \in 1..n$

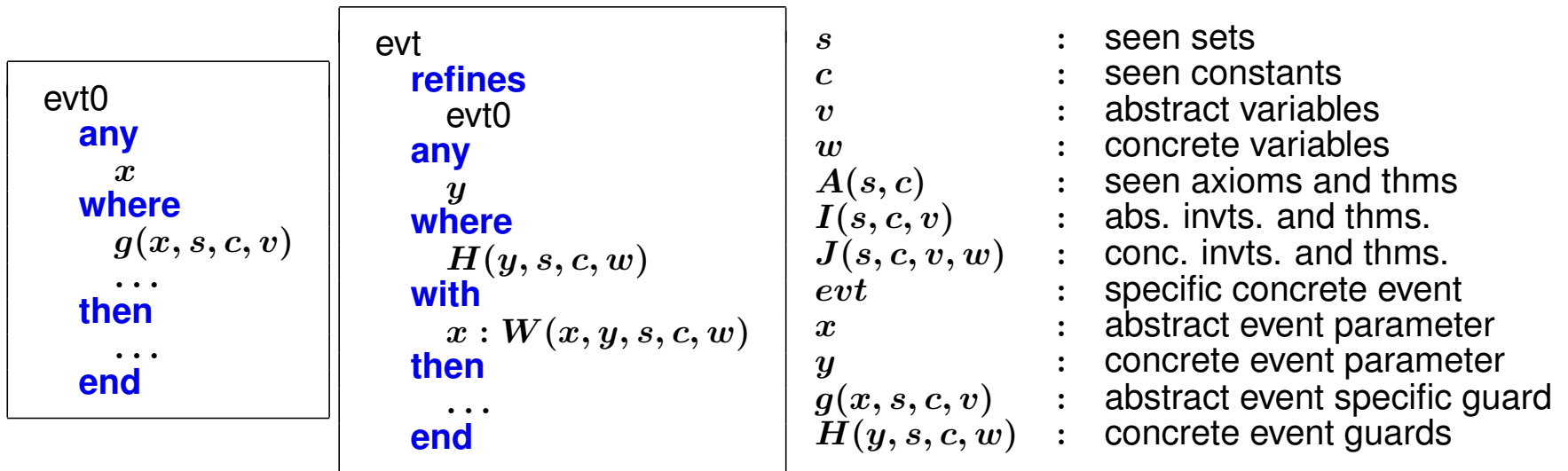
no guard in event **search**

⊢  
∃*i'* ·  $i' \in 1..n \wedge f(i') = v$

**search**  $\hat{=}$   
**status**  
ordinary  
**then**  
act1 :  $i : | i' \in 1..n \wedge f(i') = v$   
**end**

- Ensuring that each **abstract guard** is **stronger** than the **concrete ones** in the refining event.
- This ensures that when a **concrete event is enabled** then so is the **corresponding abstract one**.
- For a concrete event "**evt**" and an abstract guard "**grd**" in the corresponding abstract event, the name of this PO is:

evt/grd/FIS



<p>Axioms          Abstract invariants and thms.          Concrete invariants and thms.          Concrete event guards          witness predicate</p> <p>⊢          Abstract event specific guard</p>	<i>evt/grd/GRD</i>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------

```

A(s, c)
I(s, c, v)
J(s, c, v, w)
H(y, s, c, w)
W(x, y, s, c, w)
⊢
g(x, s, c, v)
        
```

- It is simplified when there are no parameters

```

machine
  m_1a
refines
  m_0a
sees
  ctx_0
variables
  i
  j
invariants and thms.
  inv1 :  $j \in 0 .. n$ 
  inv2 :  $v \notin f[1 .. j]$ 
theorems
  thm1 :  $v \in f[j + 1 .. n]$ 
variant
   $n - j$ 
events
  ...
end

```

```

initialisation  $\hat{=}$ 
  status
  ordinary
then
  act1 :  $i := 1$ 
  act2 :  $j := 0$ 
end

```

```

search  $\hat{=}$ 
  status
  ordinary
refines
  search
when
  grd1 :  $f(j + 1) = v$ 
with
   $k : j + 1 = k$ 
then
  act1 :  $i := j + 1$ 
end

```

```

(abstract-)search  $\hat{=}$ 
  status
  ordinary
any
   $k$ 
where
  grd1 :  $k \in 1 .. n$ 
  grd2 :  $f(k) = v$ 
then
  act1 :  $i := k$ 
end

```

- Among others, **two guard strengthening POs** are generated:
  - **search/grd1/GRD**
  - **search/grd2/GRD**

```

progress  $\hat{=}$ 
  status
  convergent
when
  grd1 :  $f(j + 1) \neq v$ 
then
  act1 :  $j := j + 1$ 
end

```

**axm1**  
**axm2**  
**axm3**  
**thm1** of **ctx\_0**  
**inv1** (abstract)  
**inv1** (concrete)  
**inv2** (concrete)  
**thm1** of **m\_1a**  
**grd1** (concrete)  
 witness predicate  
 $\vdash$   
**grd2** (abstract)

$n \in \mathbb{N}$   
 $f \in 1..n \rightarrow D$   
 $v \in \text{ran}(f)$   
 $n \in \mathbb{N1}$   
 $i \in 1..n$   
 $j \in 0..n$   
 $v \notin f[1..j]$   
 $v \in f[j+1..n]$   
 $f(j+1) = v$   
 $\frac{j+1 = k}{f(k) = v}$   
 $\vdash$

**search**  $\hat{=}$   
**status**  
   **ordinary**  
**refines**  
   **search**  
**when**  
   **grd1** :  $f(j+1) = v$   
**with**  
    $k$  :  $j+1 = k$   
**then**  
   **act1** :  $i := j+1$   
**end**

**(abstract-)search**  $\hat{=}$   
**status**  
   **ordinary**  
**any**  
    $k$   
**where**  
   **grd1** :  $k \in 1..n$   
   **grd2** :  $f(k) = v$   
**then**  
   **act1** :  $i := k$   
**end**

- Ensuring that each **action** in a concrete event **simulates** the corresponding abstract action
- This ensures that when a **concrete event is "executed"** then what it does is **not contradictory** with what the corresponding **abstract event does**.
- For a concrete event "**evt**" and an action "**act**" in both concrete and abstract events, the name of this PO is:

evt/act/SIM

```

evt0
  any
    x
  where
    ...
  then
    v :| BA1(v, v', ...)
  end
    
```

```

evt
  refines
    evt0
  any
    y
  where
    H(y, s, c, w)
  with
    x : W1(x, y, s, c, w)
    v' : W2(y, v', s, c, w)
  then
    w :| BA2(w, w', ...)
  end
    
```

*s* : seen sets  
*c* : seen constants  
*v* : abstract vrbls  
*w* : concrete vrbls  
*A(s, c)* : seen axioms and thms  
*I(s, c, v)* : abs. invts. and thms.  
*J(s, c, v, w)* : conc. invts. and thms.  
*evt* : concrete event  
*x* : abstract prm  
*y* : concrete prm  
*H(y, s, c, w)* : concrete guards  
*BA1(v, v')* : abstract action  
*BA2(w, w')* : concrete action

Axioms Abstract invariants and thms. Concrete invariants and thms. Concrete event guards witness predicate witness predicate Concrete before-after predicate ⊢ Abstract before-after predicate	<i>evt/act/SIM</i>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------

*A(s, c)*  
*I(s, c, v)*  
*J(s, c, v, w)*  
*H(y, s, c, w)*  
*W1(x, y, s, c, w)*  
*W2(y, v', s, c, w)*  
*BA2(w, w', ...)*  
 ⊢  
*BA1(v, v', ...)*

```

machine
  m_1a
refines
  m_0a
sees
  ctx_0
variables
  i
  j
invariants and thms.
  inv1 :  $j \in 0..n$ 
  inv2 :  $v \notin f[1..j]$ 
theorems
  thm1 :  $v \in f[j+1..n]$ 
variant
   $n - j$ 
events
  ...
end
    
```

```

initialisation  $\hat{=}$ 
  status
  ordinary
then
  act1 :  $i := 1$ 
  act2 :  $j := 0$ 
end
    
```

```

search  $\hat{=}$ 
  status
  ordinary
refines
  search
when
  grd1 :  $f(j+1) = v$ 
with
   $k : j+1 = k$ 
then
  act1 :  $i := j+1$ 
end
    
```

```

(abstract-)search  $\hat{=}$ 
  status
  ordinary
any
   $k$ 
where
  grd1 :  $k \in 1..n$ 
  grd2 :  $f(k) = v$ 
then
  act1 :  $i := k$ 
end
    
```

```

progress  $\hat{=}$ 
  status
  convergent
when
  grd1 :  $f(j+1) \neq v$ 
then
  act1 :  $j := j+1$ 
end
    
```

- Among others, **one simulation PO** is generated:

- **search/act1/SIM**

**axm1**  
**axm2**  
**axm3**  
**thm1** of **ctx\_0**  
**inv1** (abstract)  
**inv1** (concrete)  
**inv2** (concrete)  
**thm1** of **m\_1a**  
**grd1** (concrete)  
 witness predicate  
 $\vdash$   
 before-after predicate (abstract)

$n \in \mathbb{N}$   
 $f \in 1..n \rightarrow D$   
 $v \in \text{ran}(f)$   
 $n \in \mathbb{N}1$   
 $i \in 1..n$   
 $j \in 0..n$   
 $v \notin f[1..j]$   
 $v \in f[j+1..n]$   
 $f(j+1) = v$   
 $j+1 = k$   
 $\vdash$   
 $k = j + 1$

**search**  $\hat{=}$   
**status**  
 ordinary  
**refines**  
 search  
**when**  
 $\text{grd1} : f(j+1) = v$   
**with**  
 $k : j+1 = k$   
**then**  
 $\text{act1} : i := j+1$   
**end**

(abstract-)**search**  $\hat{=}$   
**status**  
 ordinary  
**any**  
 $k$   
**where**  
 $\text{grd1} : k \in 1..n$   
 $\text{grd2} : f(k) = v$   
**then**  
 $\text{act1} : i := k$   
**end**

- Ensuring that under the guards of each **convergent event** a proposed numeric variant is indeed a **natural number**
- For a convergent event "**evt**", the name of this PO is:

evt/NAT

```

machine
  m
refines
  ...
sees
  ...
variables
  v
invariants and thms.
   $I(s, c, v)$ 
theorems
  ...
events
  ...
variant
   $n(s, c, v)$ 
end
    
```

```

evt
  status
  convergent
  any x where
     $G(x, s, c, v)$ 
  then
     $A$ 
  end
    
```

```

s           : seen sets
c           : seen constants
v           : variables
 $A(s, c)$     : seen axioms and thms
 $I(s, c, v)$  : abs. invts. and thms.
 $J(s, c, v, w)$  : conc. invts. and thms.
evt        : specific event
x          : event parameters
 $G(x, s, c, v)$  : event guards
 $n(s, c, v)$  : numeric variant
    
```

<p>Axioms          Abstract invariants and thms.          Concrete invariants and thms.          Event guards</p> <p>⊢          a numeric variant is a natural number</p>	$evt/NAT$
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------

```

 $A(s, c)$ 
 $I(s, c, v)$ 
 $J(s, c, v, w)$ 
 $G(x, s, c, v)$ 
⊢
 $n(s, c, v) \in \mathbb{N}$ 
    
```

```

machine
  m_1a
refines
  m_0a
sees
  ctx_0
variables
  i
  j
invariants and thms.
  inv1 :  $j \in 0 .. n$ 
  inv2 :  $v \notin f[1 .. j]$ 
theorems
  thm1 :  $v \in f[j + 1 .. n]$ 
variant
   $n - j$ 
events
  ...
end
    
```

- Among others, **one numeric variant PO** is generated:

- **progress/NAT**

```

initialisation  $\hat{=}$ 
status
  ordinary
then
  act1 :  $i := 1$ 
  act2 :  $j := 0$ 
end
    
```

```

search  $\hat{=}$ 
status
  ordinary
refines
  search
when
  grd1 :  $f(j + 1) = v$ 
with
   $k : j + 1 = k$ 
then
  act1 :  $i := j + 1$ 
end
    
```

```

progress  $\hat{=}$ 
status
  convergent
when
  grd1 :  $f(j + 1) \neq v$ 
then
  act1 :  $j := j + 1$ 
end
    
```

**axm1**  
**axm2**  
**axm3**  
**thm1** of **ctx\_0**  
**inv1** (abstract)  
**inv1** (concrete)  
**inv2** (concrete)  
**thm1** of **m\_1a**  
**grd1** (concrete)

$\vdash$   
 variant is a natural number

$n \in \mathbb{N}$   
 $f \in 1..n \rightarrow D$   
 $v \in \text{ran}(f)$   
 $n \in \mathbb{N1}$   
 $i \in 1..n$   
 $j \in 0..n$   
 $v \notin f[1..j]$   
 $v \in f[j+1..n]$   
 $f(j+1) \neq v$

$\vdash$   
 $n - j \in \mathbb{N}$

**machine**  
**m\_1a**  
**refines**  
**m\_0a**  
 $\dots$   
**variant**  
 $n - j$   
**events**  
 $\dots$   
**end**

**progress**  $\hat{=}$   
**status**  
**convergent**  
**when**  
**grd1** :  $f(j+1) \neq v$   
**then**  
**act1** :  $j := j + 1$   
**end**

- Ensuring that a proposed **set variant** is indeed a **finite** set
- The name of this PO is:

FIN

```

machine
  m
refines
  ...
sees
  ...
variables
  v
invariants and thms.
   $J(s, c, v, w)$ 
theorems
  ...
events
  ...
variant
   $t(s, c, v)$ 
end
    
```

```

s           : seen sets
c           : seen constants
v           : variables
 $A(s, c)$     : seen axioms and thms
 $I(s, c, v)$  : abs. invts. and thms.
 $J(s, c, v, w)$  : conc. invts. and thms.
 $t(s, c, v)$  : set variant
    
```

Axioms Abstract invariants and thms. Concrete invariants and thms. $\vdash$ Finiteness of set variant	<b>FIN</b>
-------------------------------------------------------------------------------------------------------------------	------------

```

 $A(s, c)$ 
 $I(s, c, v)$ 
 $J(s, c, v, w)$ 
 $\vdash$ 
  finite( $t(s, c, v)$ )
    
```

```

machine
  m_1b
refines
  m_0b
sees
  ctx_0
variables
  i
  j
invariants and thms.
  inv1 :  $j \in 0 .. n$ 
  inv2 :  $v \notin f[i .. j]$ 
theorems
  thm1 :  $v \in f[j + 1 .. n]$ 
variant
  j .. n
events
  ...
end

```

- Among others, one finiteness PO is generated

```

initialisation  $\hat{=}$ 
  status
    ordinary
  then
    act1 :  $i := 1$ 
    act2 :  $j := 0$ 
  end

```

```

search  $\hat{=}$ 
  status
    ordinary
  refines
    search
  when
    grd1 :  $f(j + 1) = v$ 
  then
    act1 :  $i := j + 1$ 
  end

```

```

progress  $\hat{=}$ 
  status
    convergent
  when
    grd1 :  $f(j + 1) \neq v$ 
  then
    act1 :  $j := j + 1$ 
  end

```

```
axm1
axm2
axm3
thm1 of ctx_0
inv1 (abstract)
inv1 (concrete)
inv2 (concrete)
thm1 of m_1a
┌
variant is finite
```

```
 $n \in \mathbb{N}$ 
 $f \in 1..n \rightarrow D$ 
 $v \in \text{ran}(f)$ 
 $n \in \mathbb{N}1$ 
 $i \in 1..n$ 
 $j \in 0..n$ 
 $v \notin f[1..j]$ 
 $v \notin f[j+1..n]$ 
┌
finite( $j..n$ )
```

```
machine
  m_1b
  refines
  m_0b
  ...
  variant
   $j..n$ 
  events
  ...
end
```

- Ensuring that each **convergent event** decreases the proposed numeric variant
- For a convergent event "**evt**", the name of this PO is:

evt/VAR

```

evt
  status
  convergent
  any  $x$  where
     $G(x, s, c, w)$ 
  then
     $v :| BAP(x, s, c, w, w')$ 
  end
    
```

$s$  : seen sets  
 $c$  : seen constants  
 $v$  : variables  
 $A(s, c)$  : seen axioms and thms  
 $I(s, c, v)$  : abs. invariants and thms.  
 $J(s, c, v, w)$  : conc. invariants and thms.  
 $evt$  : specific event  
 $x$  : event parameters  
 $G(x, s, c, w)$  : event guards  
 $BAP(x, s, c, w, w')$  : event before-after predicate  
 $n(s, c, w)$  : numeric variant

<p>                     Axioms                      Abstract invariants and thms.                      Concrete invariants and thms.                      Guards of the event                      Before-after predicate of the event                      ⊢                      Modified variant smaller than variant                 </p>	$evt/VAR$
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------

$A(s, c)$   
 $I(s, c, v)$   
 $J(s, c, v, w)$   
 $G(x, s, c, w)$   
 $BAP(x, s, c, w, w')$   
 ⊢  
 $n(s, c, w') < n(s, c, w)$

```

machine
  m_1a
refines
  m_0a
sees
  ctx_0
variables
  i
  j
invariants and thms.
  inv1 :  $j \in 0 .. n$ 
  inv2 :  $v \notin f[1 .. j]$ 
theorems
  thm1 :  $v \in f[j + 1 .. n]$ 
variant
   $n - j$ 
events
  ...
end

```

- Among others, **one numeric variant decreasing PO** is generated:

- **progress/VAR**

```

initialisation  $\hat{=}$ 
status
  ordinary
then
  act1 :  $i := 1$ 
  act2 :  $j := 0$ 
end

```

```

search  $\hat{=}$ 
status
  ordinary
refines
  search
when
  grd1 :  $f(j + 1) = v$ 
with
   $k : j + 1 = k$ 
then
  act1 :  $i := j + 1$ 
end

```

```

progress  $\hat{=}$ 
status
  convergent
when
  grd1 :  $f(j + 1) \neq v$ 
then
  act1 :  $j := j + 1$ 
end

```

**axm1**  
**axm2**  
**axm3**  
**thm1** of **ctx\_0**  
**inv1** (abstract)  
**inv1** (concrete)  
**inv2** (concrete)  
**thm1** of **m\_1a**  
**grd1** (concrete)

$\vdash$   
 variant is a natural number

$n \in \mathbb{N}$   
 $f \in 1..n \rightarrow D$   
 $v \in \text{ran}(f)$   
 $n \in \mathbb{N}1$   
 $i \in 1..n$   
 $j \in 0..n$   
 $v \notin f[1..j]$   
 $v \in f[j+1..n]$   
 $f(j+1) = v$

$\vdash$   
 $n - (j + 1) < n - j$

**machine**  
**m\_1a**  
**refines**  
**m\_0a**  
 $\dots$   
**variant**  
 $n - j$   
**events**  
 $\dots$   
**end**

**progress**  $\hat{=}$   
**status**  
**convergent**  
**when**  
**grd1** :  $f(j + 1) \neq v$   
**then**  
**act1** :  $j := j + 1$   
**end**

- Ensuring that each **convergent event** decreases the proposed set variant
- For a convergent event "**evt**", the name of this PO is:

evt/VAR

```

evt
  status
  convergent
  any  $x$  where
     $G(x, s, c, w)$ 
  then
     $v :| BAP(x, s, c, w, w')$ 
  end
    
```

$s$  : seen sets  
 $c$  : seen constants  
 $v$  : variables  
 $A(s, c)$  : seen axioms and thms  
 $I(s, c, v)$  : abs. invariants and thms.  
 $J(s, c, v, w)$  : conc. invariants and thms.  
 $evt$  : specific event  
 $x$  : event parameters  
 $G(x, s, c, v)$  : event guards  
 $BAP(x, s, c, w, w')$  : event before-after predicate  
 $t(s, c, w)$  : set variant

<p>                     Axioms                      Abstract Invariants                      Concrete Invariants                      Guards of the event                      Before-after predicate of the event  <math>\vdash</math>                      Modified variant strictly included in variant                 </p>	$evt/VAR$
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------

$A(s, c)$   
 $I(s, c, v)$   
 $J(s, c, v, w)$   
 $G(x, s, c, v)$   
 $BAP(x, s, c, w, w')$   
 $\vdash$   
 $t(s, c, w') \subset t(s, c, w)$

```

machine
  m_1b
refines
  m_0b
sees
  ctx_0
variables
  i
  j
invariants and thms.
  inv1 :  $j \in 0 .. n$ 
  inv2 :  $v \notin f[1 .. j]$ 
theorems
  thm1 :  $v \in f[j + 1 .. n]$ 
variant
  j .. n
events
  ...
end
    
```

- Among others, **one variant decreasing PO** is generated:
  - **progress/VAR**

```

initialisation  $\hat{=}$ 
  status
  ordinary
  then
    act1 :  $i := 1$ 
    act2 :  $j := 0$ 
  end
    
```

```

search  $\hat{=}$ 
  status
  ordinary
  refines
  search
  when
    grd1 :  $f(j + 1) = v$ 
  then
    act1 :  $i := j + 1$ 
  end
    
```

```

progress  $\hat{=}$ 
  status
  convergent
  when
    grd1 :  $f(j + 1) \neq v$ 
  then
    act1 :  $j := j + 1$ 
  end
    
```

```

axm1
axm2
axm3
thm1 of ctx_0
inv1 (abstract)
inv1 (concrete)
inv2 (concrete)
thm1 of m_1a
inv2 (concrete)
grd1 (concrete)
┆
variant is a natural number
    
```

```

n ∈ ℕ
f ∈ 1..n → D
v ∈ ran(f)
n ∈ ℕ1
i ∈ 1..n
j ∈ 0..n
v ∉ f[1..j]
v ∈ f[j+1..n]
f(j+1) = v
┆
j+1..n ⊂ j..n
    
```

```

machine
  m_1b
refines
  m_0b
  ...
variant
  j..n
events
  ...
end
    
```

```

progress ≐
status
  convergent
when
  grd1 : f(j+1) ≠ v
then
  act1 : j := j+1
end
    
```

- Ensuring that each **witness** proposed in the witness predicate of a concrete event indeed **exists**
- For a concrete event "**evt**", and an abstract parameter ***x*** the name of this PO is:

*evt/x/WFIS*

```

evt
  refines
    evt0
  any
    y
  where
     $H(y, s, c, w)$ 
  with
     $x : W(x, y, s, c, w)$ 
  then
    ...
  end
    
```

- $s$  : seen sets
- $c$  : seen constants
- $v$  : abstract variables
- $w$  : concrete variables
- $A(s, c)$  : seen axioms and thms
- $I(s, c, v)$  : abs. invts. and thms.
- $J(s, c, v, w)$  : conc. invts. and thms.
- $evt$  : specific concrete event
- $x$  : abstract event parameter
- $y$  : concrete event parameter
- $H(y, s, c, w)$  : concrete event guards
- $W(x, y, s, c, w)$  : witness predicate

<p>Axioms          Abstract invariants and thms.          Concrete invariants and thms.          Concrete event guards</p> <p>⊢  <math>\exists x \cdot \text{Witness}</math></p>	$evt/x/WFIS$
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------

$A(s, c)$   
 $I(s, c, v)$   
 $J(s, c, v, w)$   
 $H(y, s, c, w)$

⊢  
 $\exists x \cdot W(x, y, s, c, w)$

```

machine
  m_1a
refines
  m_0a
sees
  ctx_0
variables
  i
  j
invariants and thms.
  inv1 :  $j \in 0 .. n$ 
  inv2 :  $v \notin f[1 .. j]$ 
theorems
  thm1 :  $v \in f[j + 1 .. n]$ 
variant
   $n - j$ 
events
  ...
end

```

- Among others, **one witness feasibility PO** is generated:

- **search/k/WFIS**

```

initialisation  $\hat{=}$ 
status
  ordinary
then
  act1 :  $i := 1$ 
  act2 :  $j := 0$ 
end

```

```

search  $\hat{=}$ 
status
  ordinary
refines
  search
when
  grd1 :  $f(j + 1) = v$ 
with
  k :  $j + 1 = k$ 
then
  act1 :  $i := j + 1$ 
end

```

```

progress  $\hat{=}$ 
status
  convergent
when
  grd1 :  $f(j + 1) \neq v$ 
then
  act1 :  $j := j + 1$ 
end

```

**axm1**  
**axm2**  
**axm3**  
**thm1** of **ctx\_0**  
**inv1** (abstract)  
**inv1** (concrete)  
**inv2** (concrete)  
**thm1** of **m\_1a**  
**grd1** (concrete)  
 $\vdash$   
 $\exists k \cdot$  variant predicate

$n \in \mathbb{N}$   
 $f \in 1..n \rightarrow D$   
 $v \in \text{ran}(f)$   
 $n \in \mathbb{N}1$   
 $i \in 1..n$   
 $j \in 0..n$   
 $v \notin f[1..j]$   
 $v \in f[j+1..n]$   
 $f(j+1) = v$   
 $\vdash$   
 $\exists k \cdot j+1 = k$

**search**  $\hat{=}$   
**status**  
 ordinary  
**refines**  
 search  
**when**  
**grd1** :  $f(j+1) = v$   
**with**  
 $k : j+1 = k$   
**then**  
**act1** :  $i := j+1$   
**end**

- Ensuring that a proposed **context theorem** is indeed **provable**
- Theorems are **important** in that they might **simplify some proofs**
- For a theorem "**thm**" in a context, the name of this PO is:

thm/THM

```

context
  ctx
extends
  ...
sets
  s
constants
  c
axioms
   $A(s, c)$ 
theorems
  ...
  thm :  $P(s, c)$ 
  ...
end
    
```

*s* : seen sets  
*c* : seen constants  
 $A(s, c)$  : seen axioms and previous thms  
 $P(s, c)$  : specific theorem

Axioms $\vdash$ Theorem	$thm/THM$
-------------------------------	-----------

$\vdash$   
 $A(s, c)$   
 $P(s, c)$

```
context
  ctx_0
sets
   $D$ 
constants
   $n$ 
   $f$ 
   $v$ 
axioms
  axm1 :  $n \in \mathbb{N}$ 
  axm2 :  $f \in 1..n \rightarrow D$ 
  axm3 :  $v \in \text{ran}(f)$ 
theorems
  axm1 :  $n \in \mathbb{N}1$ 
end
```

- One theorem PO is generated: **thm1**/THM

**axm1**

**axm2**

**axm3**

⊢

**thm1**

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

$f \in 1..n \rightarrow D$

$\frac{v \in \text{ran}(f)}$

⊢

$n \in \mathbb{N}1$

- Ensuring that a proposed **machine theorem** is indeed **provable**
- Theorems are **important** in that they might **simplify some proofs**
- For a theorem "**thm**" in a machine, the name of this PO is:

thm/THM

```

machine
  m0
refines
  ...
sees
  ...
variables
  v
invariants and thms.
   $I(s, c, v)$ 
theorems
  ...
  thm :  $P(s, c, v)$ 
  ...
events
  ...
end
    
```

```

s      : seen sets
c      : seen constants
v      : variables
 $A(s, c)$  : seen axioms and thms
 $I(s, c, v)$  : invariants and previous thms.
 $P(s, c, v)$  : specific theorem
    
```

Axioms Invariants $\vdash$ Theorem	$thm/THM$
---------------------------------------------	-----------

```

 $A(s, c)$ 
 $I(s, c, v)$ 
 $\vdash$ 
 $P(s, c, v)$ 
    
```

```

machine
  m_1a
refines
  m_0a
sees
  ctx_0
variables
  i
  j
invariants and thms.
  inv1 :  $j \in 0 .. n$ 
  inv2 :  $v \notin f[1 .. j]$ 
theorems
  thm1 :  $v \in f[j + 1 .. n]$ 
variant
   $n - j$ 
events
  ...
end

```

- Among others, **one theorem PO** is generated: **thm1/THM**

**axm1**

**axm2**

**axm3**

**thm1** of **ctx\_0**

**inv1** (abstract)

**inv1** (concrete)

**inv2** (concrete)

⊢

**thm1** of **m\_1a**

$n \in \mathbb{N}$

$f \in 1..n \rightarrow D$

$v \in \text{ran}(f)$

$n \in \mathbb{N}1$

$i \in 1..n$

$j \in 0..n$

$v \notin f[1..j]$

⊢

$v \in f[j + 1..n]$

- Ensuring that a **potentially ill-defined** axiom, theorem, invariant, guard, action, variant, or witness is indeed **well-defined**
- For a given modeling element (axm, thm, inv, grd, act), or a variant, or a witness  $x$  in an event evt, the names are:

axm/WD, thm/WD, inv/WD, grd/WD, act/WD, VWD, evt/ $x$ /WWD

- It depends on the **potentially ill-defined expression**

$\text{inter}(S)$	$S \neq \emptyset$
$\bigcap x \cdot x \in S \wedge P(x) \mid T(x)$	$\exists x \cdot x \in S \wedge P(x)$
$f(E)$	$f$ is a partial function $E \in \text{dom}(f)$
$E/F$	$F \neq 0$
$E \bmod F$	$F \neq 0$
$\text{card}(S)$	$\text{finite}(S)$
$\text{min}(S)$	$S \subseteq \mathbb{Z}$ $\exists x \cdot x \in \mathbb{Z} \wedge (\forall n \cdot n \in S \Rightarrow x \leq n)$
$\text{max}(S)$	$S \subseteq \mathbb{Z}$ $\exists x \cdot x \in \mathbb{Z} \wedge (\forall n \cdot n \in S \Rightarrow x \geq n)$

```

context
  ctx_0
sets
  D
constants
  n
  f
  v
axioms
  axm1 :  $n \in \mathbb{N}$ 
  axm2 :  $f \in 1..n \rightarrow D$ 
  axm3 :  $v \in \text{ran}(f)$ 
theorems
  axm1 :  $n \in \mathbb{N}1$ 
end
  
```

```

machine
  m_0a
sees
  ctx_0
variables
  i
invariants and thms.
  inv1 :  $i \in 1..n$ 
events
  ...
end
  
```

```

initialisation  $\hat{=}$ 
status
  ordinary
then
  act1 :  $i := 1$ 
end
  
```

```

search  $\hat{=}$ 
status
  ordinary
any
  k
where
  grd1 :  $k \in 1..n$ 
  grd2 :  $f(k) = v$ 
then
  act1 :  $i := k$ 
end
  
```

- One well-definedness PO is generated:
  - search/grd2/WD

**axm1****axm2****axm3****thm1****inv1****grd1**

⊢

WD conditions for **grd2** $n \in \mathbb{N}$  $f \in 1..n \rightarrow D$  $v \in \text{ran}(f)$  $n \in \mathbb{N}1$  $i \in 1..n$  $k \in 1..n$ 

⊢

 $k \in \text{dom}(f) \wedge f \in \mathbb{Z} \leftrightarrow D$

```

evt01
  any
    x
  where
    G1(x, s, c, v)
  then
    A
  end

evt02
  any
    x
  where
    G2(x, s, c, v)
  then
    A
  end
    
```

```

evt
  refines
    evt01
    evt02
  any
    x
  where
    H(x, s, c, v)
  then
    A
  end
    
```

*s* : seen sets  
*c* : seen constants  
*v* : abstract vrbls  
*A(s, c)* : seen axioms and thms  
*I(s, c, v)* : abs. invariants and thms.  
*evt* : concrete event  
*x* : similar prm  
*H(x, s, c, v)* : concrete guards  
*G1(x, s, c, v)* : abstract event guards  
*G2(x, s, c, v)* : abstract event guards  
*A* : similar abs. and cnc. actions

Axioms Abstract invariants and thms. Concrete event guards $\vdash$ Disjunction of abstract guards	<i>evt</i> /MRG
----------------------------------------------------------------------------------------------------------------	-----------------

*A(s, c)*  
*I(s, c, v)*  
*H(x, s, c, v)*  
 $\vdash$   
 $G1(x, s, c, v) \vee G2(x, s, c, v)$

- Context **ctx\_0**
  - **thm1/THM**
- Machine **m\_0a**
  - **initialisation/inv1/INV**
  - **search/gdr2/WD**
  - **search/inv1/INV**
- Machine **m\_0b**
  - **initialisation/inv1/INV**
  - **search/inv1/INV**
  - **search/act1/WD**
  - **search/act1/FIS**

- **Machine  $m_{1a}$** 
  - **thm1/THM**
  - **initialisation/inv1/INV**
  - **initialisation/inv2/INV**
  - **search/gdr1/WD**
  - **search/ $k$ /WFIS**
  - **search/gdr1/GRD**
  - **search/gdr2/GRD**
  - **search/act1/SIM**
  - **progress/gdr1/WD**
  - **progress/inv1/INV**
  - **progress/inv2/INV**
  - **progress/VAR**
  - **progress/NAT**

- **Machine m\_1b**
  - **thm1/THM**
  - **FIN**
  - **initialisation/inv1/INV**
  - **initialisation/inv2/INV**
  - **search/gdr1/WD**
  - **search/act1/SIM**
  - **progress/gdr1/WD**
  - **progress/inv1/INV**
  - **progress/inv2/INV**
  - **progress/VAR**