Extending EB4EB for Parameterised Events

Peter Rivière[1] Neeraj Kumar Singh[2] Yamine Aït-Ameur[2]
Guillaume Dupont[2]

[1] Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan
[2] Toulouse National Polytechnique Institute
IRIT-CNRS, University of Toulouse, France

June 10-13, 2023

International Conference
ABZ'2025

Outline

Introduction

Motivation

EB4EB Framework ⇒ Access to Event-B Component + Advanced Reasoning + New PO

Event Parameter ⇒ Missing explicit manipulation in EB4EB

- Local bind to Event
- Each event parameters have different types

Our proposal

Upgrade EB4EB framework with explicit manipulation of Event-Parameter

- Polymorphic operator to instantiate at the event level
- Axiomatic data-type instead of Constructive data-type

Event-B

Contexts (definitions/axioms/theorems) + Machines (state and events/inductive invariants)

Context	Machine
CONTEXT Ctx **SETS** s **CONSTANTS** c **AXIOMS** A **THEOREMS** T_{ctx} **END**	**MACHINE** M^A **SEES** Ctx **VARIABLES** x^A **INVARIANTS** $I^A(x^A)$ **THEOREMS** $T_{mch}(x^A)$ **VARIANT** $V(x^A)$ **EVENTS** INITIALISATION $\widehat{=}$ ... evt^A $\widehat{=}$ **ANY** α^A **WHERE** $G^A(x^A, \alpha^A)$ **THEN** x^A :| $BAP^A($ $\alpha^A, x^A, x^{A\prime})$ **END** **END**

Theorems Context (THM_ctx)	$A \Rightarrow T_{ctx}$
Theorems machine (THM_mch)	$A \wedge I^A(x^A) \Rightarrow T_{mch}(x^A)$
Initialisation (INIT)	$A \wedge AP^A(\alpha^A, x^{A\prime}) \Rightarrow I^A(x^{A\prime})$
Invariant preservation (INV)	$A \wedge I_A(x^A) \wedge G_A(x^A, \alpha^A)$ $\wedge BAP^A(x^A, \alpha^A, x^{A\prime}) \Rightarrow I^A(x^{A\prime})$
Event feasibility (FIS)	$A \wedge I_A(x^A) \wedge G^A(x^A, \alpha^A)$ $\Rightarrow \exists x^{A\prime} \cdot BAP^A(x^A, \alpha^A, x^{A\prime})$
Variant progress (VAR)	$A \wedge I^A(x^A) \wedge G^A(x^A, \alpha^A)$ $\wedge BAP^A(x^A, \alpha^A, x^{A\prime})$ $\Rightarrow V(x^{A\prime}) < V(x^A)$

- Automatic generation of POs.
- Rodin $\Longrightarrow$ Editor + POs generator + automatic/interactive provers

Event-B Theories

Formalisation of new data-types for Event-B $\Longrightarrow$ Theories

Theory
THEORY Th
IMPORT Th1, ...
TYPE PARAMETERS E, F, ...
DATATYPES
Type2(E, ...)
constructors
cstr1(p_1 : T_1, ...)
OPERATORS
Op1 <nature> (p_1 : T_1, ...)
well−definedness $WD(p_1, \ldots)$
direct definition D_1
AXIOMATIC DEFINITIONS
TYPES A_1, ...
OPERATORS
AOp2 <nature> (p_1 : T_1, ...): T_r
well−definedness $WD(p_1, \ldots)$
AXIOMS A_1, ...
THEOREMS T_1, ...
END

- Algebraic definition for data-types $\Longrightarrow$ Constructive definitions and/or axiomatic definitions.
- Each operator may be associated with Well-Definedness (WD) conditions (partial definitions).
- Relevant proved theorems.
- Relevant inference/rewrite rules.
- Tool-support in Rodin environment.

Event-B theories: an example

Extend Event-B with mathematical structures or domain knowledge
$\Rightarrow$ *reals, differential equations, kinematics, ontologies...*

```
THEORY Reals IMPORT Relations, Rings
AXIOMATIC DEFINITIONS
  TYPES ℝ
  OPERATORS
          Rzero : ℝ,
          Rone : ℝ,
          plus : ℝ × ℝ → ℝ,
          times : ℝ × ℝ → ℝ,
          leq : ℝ ↔ ℝ,
          ...
  AXIOMS
    order : order(leq) ∧ total(leq)
    reals : Field(plus, times, Rzero, Rone)∧
     integral(times, Rzero)∧
     ringCompatible(plus, times, Rzero, Rone, leq)
    ...
```

```
MACHINE M
VARIABLES x, y,
INVARIANTS
  inv1 : x, y ∈ ℝ
  inv2 : Rzero leq x ∧ x ≠ Rzero
  inv3 : y = Rzero ∨ y = Rone
EVENTS
  ...
 Evt
 WHERE
   grd1 : x times y = Rzero
   grd2 : v = Open
 THEN y := y plus Rone
 END
```

The EB4EB Meta Theory

- Explicit manipulation of Event-B features
  $\Longrightarrow$ a constructive data-type for Event-B machines.

```
EventBTheory
THEORY EventBTheory
TYPES PARAMETERS STATE,EVENT
DATATYPES
Machine(STATE,EVENT) ≙
    ▷Cons_machine(
      Init:EVENT,
      Progress:ℙ(EVENT),
      AP:ℙ(STATE),
      BAP:
       ℙ(EVENT × (STATE × STATE)),
      Grd:
       ℙ(EVENT × STATE),
      Inv:ℙ(STATE),
      ...)
OPERATORS
...
 check_Machine_consistency(m)...
```

- STATE: machine states
- EVENT: machine events
- Init: initialisation event
- Progress: progress events
- AP (After Predicate): initialisation event action
- BAP (Before-After Predicate): actions of progress events
- Grd: progress events guards
- Inv: machine invariants
- **Check_Machine_Consistency**: core Event-B POs

```
Machine
MACHINE M SEES ...
VARIABLES x
INVARIANTS I(x)
...
EVENTS
  INITIALISATION ≙
    THEN
      x :| AP(α, x)
    END
  EVT_i ≙
    ANY α
    WHERE G_i(x, α)
    THEN
      x :| BAP_i(α, x, x)
  END
  ...
```

Check_Machine_Consistency(m) $\equiv$ PO_INV(m) $\wedge$ PO_THM(m) $\wedge$ PO_FIS(m) $\wedge$ PO_VAR(m) $\wedge$ PO_NAT(m)

Instantiation of the EB4EB Meta Theory

Event-B Machines as a FOL expression

- An Event-B machine is seen as a FOL formula (sets, constants, and axioms)

CONTEXT Deep
SETS St Ev
CONSTANTS m, ...
AXIOMS
axm1 : $partition(Ev, \ldots)$
axm2 : $m \in Machine(St, Ev)$
axm3 : $Event(m) = Ev$
axm4 : $State(m) = \ldots$
axm5 : $Init(m) = \ldots$
axm6 : $Progress(m) = \{\ldots\}$
axm7 : $Thm(m) = \{\ldots\}$
axm8 : $Inv(m) = \{\ldots\}$
axm9 : $AP(m) = \{\ldots\}$
axm10 : $Grd(m) = \{\ldots\}$
axm11 : $BAP(m) = \{\ldots\}$
axm12 : $Convergent(m) = \{\ldots\}$
axm13 : $Ordinary(m) = \{\ldots\}$
axm14 : $Variant(m) = \{\ldots\}$
THEOREMS
thm1 : $\boldsymbol{check\_Machine\_consistency}(m)$
END

- Machine consistency is ensured by discharging the *THM_ctx* PO generated for the thm1 theorem

$$axm1 \land \ldots \land axm14$$
$$\Rightarrow Check\_Machine\_consistency(m)$$

Event Parameter

Example

MACHINE M
...
EVENTS
ev1 $\triangleq$ **ANY** p **WHEN** $p \in T^{ev1}_{Par}$
...
evn $\triangleq$ **ANY** p **WHEN** $p \in T^{evn}_{Par}$

$$Param(M) = \{ev1 \mapsto T^{ev1}_{Par}\} \cup \ldots \cup \{evn \mapsto T^{evn}_{Par}\}$$

Event Parameter

Example

```
MACHINE M
...
EVENTS
   ev1 ≙ ANY p WHEN p ∈ T_Par^ev1
   ...
   evn ≙ ANY p WHEN p ∈ T_Par^evn
```

$$Param(M) = \{ev1 \mapsto T_{Par}^{ev1}\} \cup \ldots \cup \{evn \mapsto T_{Par}^{evn}\}$$

$$Param(M) ⨾ Progress(M) \to Type$$

Event Parameter

Example

MACHINE M
...
EVENTS
ev1 $\widehat{=}$ **ANY** p **WHEN** $p \in T_{Par}^{ev1}$
...
evn $\widehat{=}$ **ANY** p **WHEN** $p \in T_{Par}^{evn}$

$$Param(M) = \{ev1 \mapsto T_{Par}^{ev1}\} \cup \ldots \cup \{evn \mapsto T_{Par}^{evn}\}$$
$$Param(M) ⦂ Progress(M) \rightarrow Type$$
$$BAP(M) ⦂ (e ⦂ Progress(M) \leftrightarrow$$
$$(Param(M)(e) \times STATE \leftrightarrow STATE))$$

Event Parameter

Example

MACHINE M
...
EVENTS
ev1 $\triangleq$ **ANY** p **WHEN** $p \in T^{ev1}_{Par}$
...
evn $\triangleq$ **ANY** p **WHEN** $p \in T^{evn}_{Par}$

$Param(M) = \{ev1 \mapsto T^{ev1}_{Par}\} \cup \ldots \cup \{evn \mapsto T^{evn}_{Par}\}$
$Param(M) \,; Progress(M) \rightarrow Type$
$BAP(M) \,; (e \,; Progress(M) \leftrightarrow$
$(Param(M)(e) \times STATE \leftrightarrow STATE))$
Natively Impossible ⚠

Event Parameter

Example

MACHINE M
...
EVENTS
ev1 $\widehat{=}$ **ANY** p **WHEN** $p \in T_{Par}^{ev1}$
...
evn $\widehat{=}$ **ANY** p **WHEN** $p \in T_{Par}^{evn}$

$$Param(M) = \{ev1 \mapsto T_{Par}^{ev1}\} \cup \ldots \cup \{evn \mapsto T_{Par}^{evn}\}$$
$$Param(M) \mathbin{⨾} Progress(M) \rightarrow Type$$
$$BAP(M) \mathbin{⨾} (e \mathbin{⨾} Progress(M) \leftrightarrow (Param(M)(e) \times STATE \leftrightarrow STATE))$$

Natively Impossible ⚠

Solutions

1) Global parameters

THEORY EventBTheory
TYPE PARAMETERS *PARAM* , ...

$$PARAM(M) \equiv T_{Par}^{ev1} \times \ldots \times T_{Par}^{evn}$$

Event Parameter

Example

```
MACHINE M
...
EVENTS
   ev1 ≙ ANY p WHEN p ∈ T_Par^ev1
   ...
   evn ≙ ANY p WHEN p ∈ T_Par^evn
```

$Param(M) = \{ev1 \mapsto T_{Par}^{ev1}\} \cup \ldots \cup \{evn \mapsto T_{Par}^{evn}\}$

$Param(M) \mathbin{⦂} Progress(M) \rightarrow Type$

$BAP(M) \mathbin{⦂} (e \mathbin{⦂} Progress(M) \leftrightarrow$

$(Param(M)(e) \times STATE \leftrightarrow STATE))$

Natively Impossible ⚠

Solutions

1) Global parameters

```
THEORY EventBTheory
TYPE PARAMETERS PARAM, ...
```

$PARAM(M) \equiv T_{Par}^{ev1} \times \ldots \times T_{Par}^{evn}$

No local definition of Event Parameter

Event Parameter

Example

```
MACHINE M
...
EVENTS
  ev1 ≙ ANY p WHEN p ∈ T_Par^ev1
  ...
  evn ≙ ANY p WHEN p ∈ T_Par^evn
```

$$Param(M) = \{ev1 \mapsto T_{Par}^{ev1}\} \cup \ldots \cup \{evn \mapsto T_{Par}^{evn}\}$$

$$Param(M) ⦂ Progress(M) \to Type$$

$$BAP(M) ⦂ (e ⦂ Progress(M) \leftrightarrow (Param(M)(e) \times STATE \leftrightarrow STATE))$$

Natively Impossible ⚠

Solutions

1) Global parameters

```
THEORY EventBTheory
TYPE PARAMETERS PARAM, ...
```

$PARAM(M) \equiv T_{Par}^{ev1} \times \ldots \times T_{Par}^{evn}$

No local definition of Event Parameter

2) Axiomatic definition

```
AXIOMATIC DATATYPE
  Cons_mch(EVENT, STATE)
  Cons_Event(PARAM, STATE)
```

$\text{Cons\_Event}(T_{Par}^{ev1}, \ldots)$

Event Parameter

Example

```
MACHINE M
...
EVENTS
  ev1 ≙ ANY p WHEN p ∈ T_Par^ev1
  ...
  evn ≙ ANY p WHEN p ∈ T_Par^evn
```

$Param(M) = \{ev1 \mapsto T^{ev1}_{Par}\} \cup \ldots \cup \{evn \mapsto T^{evn}_{Par}\}$

$Param(M) ⨾ Progress(M) \rightarrow Type$

$BAP(M) ⨾ (e ⨾ Progress(M) \leftrightarrow$

$(Param(M)(e) \times STATE \leftrightarrow STATE))$

Natively Impossible ⚠

Solutions

1) Global parameters

```
THEORY EventBTheory
TYPE PARAMETERS PARAM, ...
```

$PARAM(M) \equiv T^{ev1}_{Par} \times \ldots \times T^{evn}_{Par}$

No local definition of Event Parameter

2) Axiomatic definition

```
AXIOMATIC DATATYPE
  Cons_mch(EVENT, STATE)
  Cons_Event(PARAM, STATE)
```

$\text{Cons\_Event}(T^{ev1}_{Par}, \ldots)$

Solutions presented here

Inspiration

Record in classical Event-B

SETS Records, A, B // *record A {dest1 : A ; dest2 : B}*
CONSTANTS dest1, dest2
AXIOMS
  axm1 $dest1 \in Records \to A$
  axm1 $dest2 \in Records \to B$

Polymorphic operator

TYPE PARAMETERS $\alpha, \beta, \ldots$
OPERATORS
  $op1(x : \alpha) \triangleq \ldots$
  $op2(y : \mathbb{P}(\alpha), z : \alpha \times \beta) \triangleq \ldots$
  $op3(w : \beta \times \alpha \times \mathbb{Z} \times \ldots) \triangleq \ldots$

CONTEXT Ctx
AXIOMS
  axm1: $op1(x : \mathbb{Z} \times \mathbb{Z}) \ldots$ // $\alpha := \mathbb{Z} \times \mathbb{Z}$
  axm2: $op1(x : \mathbb{P}(\mathbb{Z})) \ldots$ // $\alpha := \mathbb{P}(\mathbb{Z})$
  axm3: $op2(y : \mathbb{P}(\mathbb{Z} \times BOOL), z : (\mathbb{Z} \times BOOL) \times BOOL) \ldots$ // $\alpha := \mathbb{Z} \times BOOL,\ \beta := BOOL$

Revised Datatype

1) Definition of the record type in Data-Type with Type parameter commun for all the data-type

```
THEORY EvtBTheoPar
TYPE PARAMETERS STATE, EVENT, PARAM
DATATYPES
 Machine(STATE, EVENT) //No PARAM
 CONSTRUCTORS
    Cons_machine(
      Event : ℙ(EVENT),
      State : ℙ(STATE))
```

Revised Datatype

2) Field are definition with Axiomatic Operator

```
AXIOMATIC DEFINITIONS
OPERATORS
 Init <expression> (m : Machine(STATE, EVENT)) : EVENT
        // Init ∈ Machine(STATE, EVENT) ↔ EVENT
 Progress <expression> (m : Machine(STATE, EVENT)) : ℙ(EVENT)
 Param <expression> (m : Machine(STATE, EVENT), e : EVENT,          ) : ℙ(PARAM)
        // Param ∈ Machine(STATE, EVENT) × EVENT ↔ ℙ(PARAM)
 Grd_par <expression> (m : Machine(STATE, EVENT), e : EVENT,          ) : ℙ(PARAM × STATE)
 BAP_par <expression>
      (m : Machine(STATE, EVENT), e : EVENT,          ) : ℙ((PARAM × STATE) × STATE)
 Inv <expression> (m : Machine(STATE, EVENT)) : ℙ(STATE)
 Thm <expression> (m : Machine(STATE, EVENT)) : ℙ(STATE)
   ...
```

- Type parameter Param is Bound to the definition of the field not to Machine.
- No axioms in the theory ⇒ Definition provide in instanciation.

Revised Datatype

2) Field are definition with Axiomatic Operator

AXIOMATIC DEFINITIONS
OPERATORS

Init <**expression**> $(m : Machine(STATE, EVENT)) : EVENT$
  $//\ Init \in Machine(STATE, EVENT) \leftrightarrow EVENT$
Progress <**expression**> $(m : Machine(STATE, EVENT)) : \mathbb{P}(EVENT)$
Param <**expression**> $(m : Machine(STATE, EVENT), e : EVENT, p : \mathbb{P}(PARAM)) : \mathbb{P}(PARAM)$
  $//\ Param \in Machine(STATE, EVENT) \times EVENT \leftrightarrow \mathbb{P}(PARAM)$
Grd_par <**expression**> $(m : Machine(STATE, EVENT), e : EVENT, p : \mathbb{P}(PARAM)) : \mathbb{P}(PARAM \times STATE)$
BAP_par <**expression**>
  $(m : Machine(STATE, EVENT), e : EVENT, p : \mathbb{P}(PARAM)) : \mathbb{P}((PARAM \times STATE) \times STATE)$
Inv <**expression**> $(m : Machine(STATE, EVENT)) : \mathbb{P}(STATE)$
Thm <**expression**> $(m : Machine(STATE, EVENT)) : \mathbb{P}(STATE)$
...

- Type parameter Param is Bound to the definition of the field not to Machine.
- No axioms in the theory $\Rightarrow$ Definition provide in instanciation.
- New argument for correct type checking.

New methodology of instantiation for deep medelling

- Classical Event-B

MACHINE M
...
EVENTS
  ev1 $\triangleq$ **ANY** p **WHEN** $p \in T_{Par}^{ev1}$
  ...
  evn $\triangleq$ **ANY** p **WHEN** $p \in T_{Par}^{evn}$

New methodology of instantiation for deep medelling

- Classical Event-B

```
MACHINE M
...
EVENTS
  ev1 ≙ ANY p WHEN p ∈ T_Par^ev1
  ...
  evn ≙ ANY p WHEN p ∈ T_Par^evn
```

- Revised Instantiation

CONTEXT EvtInstantiationSchema
AXIOMS

...

AxmParEv.1 : $Param(m, ev1, T_{Par}^{ev1}) = T_{Par}^{ev1}$

...

AxmParEv.n : $Param(m, evn, T_{Par}^{evn}) = T_{Par}^{evn}$

AxmParGrd.1 : $Grd\_par(m, ev1, T_{Par}^{ev1}) = \{par \mapsto s \mid par \in T_{Par}^{ev1} \wedge \ldots\}$

...

AxmParGrd.n : $Grd\_par(m, evn, T_{Par}^{evn}) = \{par \mapsto s \mid par \in T_{Par}^{evn} \wedge \ldots\}$

AxmParBAP.1 : $BAP\_par(m, ev1, T_{Par}^{ev1}) = \{(par \mapsto s) \mapsto sp \mid par \in T_{Par}^{ev1} \wedge \ldots\}$

...

AxmParBAP.n : $BAP\_par(m, evn, T_{Par}^{evn}) = \{(par \mapsto s) \mapsto sp \mid par \in T_{Par}^{evn} \wedge \ldots\}$

- Param is instantiated with the correct type for each events.

$\Rightarrow$ Events components are definition individualy for each events.

Revised Proof Obligation

Case of Invariance

OPERATORS

...

Mch_INV_One_Ev_Par_Def **predicate** $(m : Machine(STATE, EVENT), e : EVENT, \quad\quad))$

well−definedness $e \in Progress(m)$

direct definition

$BAP\_par(m, e)[(Param(m, e, \mathbb{P}(PARAM)) \times Inv(m)) \cap Grd\_par(m, e, \mathbb{P}(PARAM))] \subseteq Inv(m)$

Mch_INV_Init **predicate** $(m : Machine(STATE, EVENT))$

direct definition

$AP(m) \subseteq Inv(m)$

Mch_INV **predicate** $(m : Machine(STATE, EVENT))$

direct definition

$Mch\_INV\_Init(m) \wedge (\forall e \cdot e \in Progress(m) \Rightarrow Mch\_INV\_One\_Ev\_Def(m, e, \quad\quad)))$

...

Revised Proof Obligation

Case of Invariance

OPERATORS

...

Mch_INV_One_Ev_Par_Def predicate $(m : Machine(STATE, EVENT), e : EVENT, p : \mathbb{P}(PARAM))$

well−definedness $e \in Progress(m)$

direct definition

$BAP\_par(m, e)[(Param(m, e, \mathbb{P}(PARAM)) \times Inv(m)) \cap Grd\_par(m, e, \mathbb{P}(PARAM))] \subseteq Inv(m)$

Mch_INV_Init predicate $(m : Machine(STATE, EVENT))$

direct definition

$AP(m) \subseteq Inv(m)$

Mch_INV predicate $(m : Machine(STATE, EVENT))$

direct definition

$Mch\_INV\_Init(m) \wedge (\forall e \cdot e \in Progress(m) \Rightarrow Mch\_INV\_One\_Ev\_Def(m, e, \mathbb{P}(PARAM)))$

...

Revised Proof Obligation

Case of Invariance

OPERATORS

...

Mch_INV_One_Ev_Par_Def **predicate** $(m : Machine(STATE, EVENT), e : EVENT, p : \mathbb{P}(PARAM))$

well−definedness $e \in Progress(m)$

direct definition

$BAP\_par(m, e)[(Param(m, e, \mathbb{P}(PARAM)) \times Inv(m)) \cap Grd\_par(m, e, \mathbb{P}(PARAM))] \subseteq Inv(m)$

Mch_INV_Init **predicate** $(m : Machine(STATE, EVENT))$

direct definition

$AP(m) \subseteq Inv(m)$

Mch_INV **predicate** $(m : Machine(STATE, EVENT))$

direct definition

$Mch\_INV\_Init(m) \land (\forall e \cdot e \in Progress(m) \Rightarrow Mch\_INV\_One\_Ev\_Def(m, e, \mathbb{P}(PARAM)))$

...

⚠Not possible ! $\Longrightarrow$ require opacity on type parameter PARAM

Revised Proof Obligation

Opacity Principle

AXIOMATIC OPERATORS
  Mch_[PO]_One_Ev_Par_Sig **predicate** $(m : Machine(STATE, EVENT), e : EVENT)$
OPERATORS
  ...
  Mch_[PO]_One_Ev_Par_Def **predicate** $(m : Machine(STATE, EVENT), e : EVENT, p : \mathbb{P}(PARAM))$
    well−definedness $e \in Progress(m)$
    direct definition
      [update PO definition with parameter]
  Mch_[PO] **predicate** $(m : Machine(STATE, EVENT))$
    direct definition
      $Mch\_[PO]\_Init(m) \land (\forall e \cdot e \in Progress(m) \Rightarrow \qquad)$
  ...
AXIOMS

- Two operators defined :
  - Def : PARAM appears and definition provide.
  - Sig : PARAM does not appear and no definition.

Revised Proof Obligation

Opacity Principle

```
AXIOMATIC OPERATORS
  Mch_[PO]_One_Ev_Par_Sig  predicate  (m : Machine(STATE, EVENT), e : EVENT)
OPERATORS
  ...
  Mch_[PO]_One_Ev_Par_Def  predicate  (m : Machine(STATE, EVENT), e : EVENT, p : ℙ(PARAM)))
      well−definedness  e ∈ Progress(m)
      direct definition
        [update PO definition with parameter]
  Mch_[PO]  predicate  (m : Machine(STATE, EVENT))
      direct definition
        Mch_[PO]_Init(m) ∧ (∀e · e ∈ Progress(m) ⇒                    )
  ...
 AXIOMS
  PO[PO]Opacity: ∀m, e · m ∈ Machine(STATE, EVENT) ∧ e ∈ Progress(m)
       ⇒ (Mch_[PO]_One_Ev_Par_Sig(m, e) ⇔ Mch_[PO]_One_Ev_Par_Def(m, e, ℙ(PARAM)))
```

- Two operators defined :
  - Def : PARAM appears and definition provide.
  - Sig : PARAM does not appear and no definition.
- New axiom links the definition Def and the opaque operator Sig.

Revised Proof Obligation

Opacity Principle

AXIOMATIC OPERATORS
  Mch_[PO]_One_Ev_Par_Sig **predicate** (m : $Machine(STATE, EVENT)$, e : $EVENT$)
OPERATORS
  ...
  Mch_[PO]_One_Ev_Par_Def **predicate** (m : $Machine(STATE, EVENT)$, e : $EVENT$, p : $\mathbb{P}(PARAM)$))
    well−definedness $e \in Progress(m)$
    direct definition
      [update PO definition with parameter]
  Mch_[PO] **predicate** (m : $Machine(STATE, EVENT)$)
    direct definition
      $Mch\_[PO]\_Init(m) \wedge (\forall e \cdot e \in Progress(m) \Rightarrow Mch\_[PO]\_One\_Ev\_Par\_Sig(m, e))$
  ...
AXIOMS
  PO[PO]Opacity: $\forall m, e \cdot m \in Machine(STATE, EVENT) \wedge e \in Progress(m)$
    $\Rightarrow (Mch\_[PO]\_One\_Ev\_Par\_Sig(m, e) \Leftrightarrow Mch\_[PO]\_One\_Ev\_Par\_Def(m, e, \mathbb{P}(PARAM)))$

- Two operators defined :
  - Def : PARAM appears and definition provide.
  - Sig : PARAM does not appear and no definition.
- New axiom links the definition Def and the opaque operator Sig.
- Sig can be use for statue the property for all event.

Conclusion

Axiomatic Data-type Comparaison

	Axiomatic DT	Classical DT	Record in Ctx
Polymorphism	Heterogeneous Type for field	Homogeneous type	No polymorphims
WD	Mix with WD and external operator	Need external operator	Axioms/invariant
Set	Partially the field	Only the field	Completly
Property	Operator with opacity	Operator	Classical Event-B.

Conclusion

Axiomatic Data-type Comparaison

	Axiomatic DT	Classical DT	Record in Ctx
Polymorphism	Heterogeneous Type for field	Homogeneous type	No polymorphims
WD	Mix with WD and external operator	Need external operator	Axioms/invariant
Set	Partially the field	Only the field	Completly
Property	Operator with opacity	Operator	Classical Event-B.

Limitations

- Equality on the Data-type machine
- Multiple instationtion of PARAM for same event

$\Rightarrow$ No WD on the type instantiation mechanism

Conclusion

EB4EB: Proof Obligation and analyses

- Using same principle, all Native proof obligation updated with event parameter
- All analyses have been updated, deadlock, Invariant weakness, Reachability [RSAD24]
- Revised Clock models
- Bridge case study

Peter Riviere, Neeraj Kumar Singh, Yamine Aït Ameur, and Guillaume Dupont.
Extending the EB4EB framework with parameterised events.
Sc. of Computer Programming, 2024.