Enhancing EB4EB Framework by Introducing Lexicographic Variants

Peter Rivière[1] Yamine Aït Ameur[2] Toshiaki Aoki[1]
Guillaume Dupont[2] Neeraj Kumar Singh[2] Takashi Tomita[1]
Duong Dinh Tran[1]

[1] *JAIST - Japan Advanced Institute of Science and Technology, Ishikawa, Japan*

[2] *INPT-ENSEEIHT/IRIT, University of Toulouse, France*

May 18-22, 2026

International Conference

ABZ'2026

Outline

Introduction

Motivation

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

Variant ⇒ Simple Variant

- Do not cover non-primitive recursive function
- Force to find a linear combination of variant

Our proposal

Move to lexicographic variant

- Algebraic theory of lexicographic variant
- Revision of existing analyses with the new formalisation

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 $\widehat{=}$ $\ldots$ $evt^A \widehat{=}$ **ANY** α^A **WHERE** $G^A(x^A, \alpha^A)$ **THEN** $x^A :\mid BAP^A($ $\alpha^A, x^A, x^{A\prime})$ **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 $\mathbb{R}$
OPERATORS
$Rzero : \mathbb{R}$,
$Rone : \mathbb{R}$,
$plus : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$,
$times : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$,
$leq : \mathbb{R} \leftrightarrow \mathbb{R}$,
...
AXIOMS
order : **order**(leq) $\wedge$ **total**(leq)
reals : **Field**($plus$, $times$, $Rzero$, $Rone$)$\wedge$
integral($times$, $Rzero$)$\wedge$
ringCompatible($plus$, $times$, $Rzero$, $Rone$, leq)
...

MACHINE M
VARIABLES x , y ,
INVARIANTS
inv1 : $x, y \in \mathbb{R}$
inv2 : $Rzero \ leq \ x \wedge x \neq Rzero$
inv3 : $y = Rzero \vee y = Rone$
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) $\widehat{=}$
$\triangleright$ *Cons_machine*(
Init:*EVENT*,
Progress:$\mathbb{P}$(*EVENT*),
AP:$\mathbb{P}$(*STATE*),
BAP:
$\mathbb{P}$(*EVENT* $\times$ (*STATE* $\times$ *STATE*)),
Grd:
$\mathbb{P}$(*EVENT* $\times$ *STATE*),
Inv:$\mathbb{P}$(*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 $\widehat{=}$
THEN
$x :| AP(\alpha, x)$
END
$EVT_i \widehat{=}$
ANY α
WHERE $G_i(x, \alpha)$
THEN
$x :| BAP_i(\alpha, x, x)$
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: $check\_Machine\_consistency(m)$
END

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

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

Variant Theory

Definition

For all $e_1, \ldots, e_n \in E_1 \times \ldots \times E_n$ and $e'_1, \ldots, e'_n \in E_1 \times \ldots \times E_n$,
$(e_1, \ldots, e_n) > e'_1, \ldots, e'_n$ iff
$\exists i \in 1..n \cdot (\forall j \in 1..i-1 \cdot e_j = e'_j) \land e_i > e'_i)$

Variant Theory

Definition

For all $e_1, \ldots, e_n \in E_1 \times \ldots \times E_n$ and $e'_1, \ldots, e'_n \in E_1 \times \ldots \times E_n$,
$(e_1, \ldots, e_n) > e'_1, \ldots, e'_n$ iff
$\exists i \in 1..n \cdot (\forall j \in 1..i-1 \cdot e_j = e'_j) \land e_i > e'_i)$

Lexicographic Variant

```
THEORY DTVariant
TYPE PARAMETERS S
DATATYPE
    DtVariant(S)
        | NilVar
        | ConsVar(

            variant: ℙ(S × ℕ),
            tailVariant: DtVariant(S)
```

Variant Theory

Definition

For all $e_1, \ldots, e_n \in E_1 \times \ldots \times E_n$ and $e'_1, \ldots, e'_n \in E_1 \times \ldots \times E_n$,
$(e_1, \ldots, e_n) > e'_1, \ldots, e'_n$ iff
$\exists i \in 1..n \cdot (\forall j \in 1..i-1 \cdot e_j = e'_j) \land e_i > e'_i)$

Lexicographic Variant

```
THEORY DTVariant
TYPE PARAMETERS S
DATATYPE
    DtVariant(S)
        | NilVar
        | ConsVar(
            domain: ℙ(S),
            variant: ℙ(S × ℕ),
            tailVariant: DtVariant(S)
```

Variant Theory

Well constructed operators

OPERATORS

WellConsDTVariant(var : DtVariant(S), domain : $\mathbb{P}(S)$)

Case var

$NilVar \Rightarrow \top$

$ConsVar(d, v, t) \Rightarrow v \in d \rightarrow \mathbb{N} \wedge domain \subseteq d \wedge WellConsDTVariant(t, domain)$

Variant Theory

Well constructed operators

OPERATORS
WellConsDTVariant (var : DtVariant (S) , domain : $\mathbb{P}(S)$)
Case var
$NilVar \Rightarrow \top$
$ConsVar(d, v, t) \Rightarrow v \in d \to \mathbb{N} \land domain \subseteq d \land WellConsDTVariant(t, domain)$

Variant Operators

OPERATORS
Decreased (var : DtVariant (S) , s : S , sp : S domain : $\mathbb{P}(S)$)
well—definedness
$s \in domain \land sp \in domain \land WellConsDTVariant(var, domain)$
recursive definition
case var
$NilVar \Rightarrow \top$
$ConsVar(d, v, t)$
$\Rightarrow v(s) > v(sp) \lor (v(s) \geq v(sp) \land Decreased(set, t, s, sp) \land t \neq NilVar)$

Variant Theory

Well constructed operators

OPERATORS
WellConsDTVariant (var : DtVariant (S) , domain : $\mathbb{P}(S)$)
Case var
$NilVar \Rightarrow \top$
$ConsVar(d, v, t) \Rightarrow v \in d \to \mathbb{N} \land domain \subseteq d \land WellConsDTVariant(t, domain)$

Variant Operators

OPERATORS
Decreased (var : DtVariant (S) , s : S , sp : S domain : $\mathbb{P}(S)$)
well−definedness
$s \in domain \land sp \in domain \land WellConsDTVariant(var, domain)$
recursive definition
case var
$NilVar \Rightarrow \top$
$ConsVar(d, v, t)$
$\Rightarrow v(s) > v(sp) \lor (v(s) \geq v(sp) \land Decreased(set, t, s, sp) \land t \neq NilVar)$

Natural and Not increased also formalised

Revised EB4EB

```
THEORY EventBTheory
IMPORT THEORY DTVariant
TYPE PARAMETERS STATE, EVENT
DATATYPE
Machine(STATE, EVENT)
  | ConMachine(
      ...,
      Variant: DtVariant(STATE)
  )
```

```
OPERATORS
  VariantWellCons(m: Machine(STATE, EVENT))
    direct definition
        WellConsDTVariant(Variant(m), Inv(m))
```

In PO:

$v(s) \in \mathbb{N} \Rightarrow Natural(v, s)$ and

$v(s) > v(s') \Rightarrow Decresead(v, s, s')$

Temporal Properties

Formalisation of temporal properties with EB4EB [RSAD23]
rules to update

The Sequent Rule of $\nearrow$	Associated Operator in EB4EB
$\nearrow P \equiv \forall x, v, v' \cdot$ $(\neg P(v) \land G(x,v) \Rightarrow V(v) \in \mathbb{N}) \land$ $(\neg P(v) \land G(x,v) \land A(x,v,v') \Rightarrow$ $V(v') < V(v)) \land$ $(P(v) \land G(x,v) \land A(x,v,v') \land$ $V(v') \in \mathbb{N} \Rightarrow$ $V(v') \leq V(v))$	**TLDivergent_In_P** <**predicate**> $(m : Machine(STATE, Ev),$ $\hat{P} : \mathbb{P}(St), v : \mathbb{P}(St \times \mathbb{Z}))$ **well−definedness** $v \in St \to \mathbb{Z}$ **direct definition** $TLConvergent\_In\_P(m, St \setminus \hat{P}, v) \land$ $\forall e \cdot e \in Progress(m) \Rightarrow ((\forall s, s' \cdot$ $s \in Inv(m) \land s \in \hat{P} \land s \in Grd(m)[\{e\}]$ $\land s' \in BAP(m)[\{e\}][\{s\}] \land v(s') \in \mathbb{N}$ $\Rightarrow v(s') \leq v(s)))$
(a)	(b)

Correctness of the proposed formalisation

Semantics & Traces

Let M be a machine. A trace of M is a sequence of states $tr = s_0 \mapsto s_1 \mapsto \ldots \mapsto s_n \mapsto \ldots$ such that [RSA22]:

- the initial state s_0 satisfies the after predicate (AP) of the initialisation event
- each pair of consecutive states s_i, s_{i+1} corresponds to the activation of an event e of M, i.e.
  - the guard of e holds for s_i
  - the BAP of e holds for $s_i \mapsto s_{i+1}$
- if tr is *finite*, then its final state must correspond to a deadlock (i.e., the system cannot progress any more)

Correctness of the proposed formalisation

Inductive reasoning on traces.

THEORY *EvtBTraces* **IMPORT** *EventBTheory*
TYPE PARAMETERS *STATE*,*EVENT*
OPERATORS

IsANextState **predicate** $(m : Machine(STATE, EVENT), s : STATE, sp : STATE)$
direct definition
$\exists e \cdot e \in Progress(m) \land s \in Grd(m)[\{e\}] \land s \mapsto sp \in BAP(m)[\{e\}]$

IsATrace **predicate** $(m : Machine(STATE, EVENT), tr : \mathbb{P}(\mathbb{N} \times STATE))$
direct definition
$(tr \in \mathbb{N} \to STATE \lor$
$(\exists n \cdot n \in \mathbb{N} \land tr \in 0..n \to STATE \land tr(n) \notin Grd(m)[Progress(m)])$ // *finite tr*
$) \land$
$tr(0) \in AP(m) \land$
$(\forall i, j \cdot i \in dom(tr) \land j \in dom(tr) \land j = i + 1 \Rightarrow IsANextState(m, tr(i), tr(j)))$

THEOREMS
...
END

Correctness of the proposed formalisation

Soundness is expressed on each defined operator

- A template for proving soundness

THEORY Theo4[PO]Correctness **IMPORT** EvtBTraces, Theo4[PO]
TYPE PARAMETERS *STATE*, *EVENT*
THEOREMS
thm_of_Correctness_of_[PO] : $\forall m, tr \cdot m \in Machine(STATE, Ev) \land$
$check\_Machine\_Consistency(m) \land IsATrace(tr, m) \land$ **[PO](m,args)**
$\Rightarrow$ ***PO_Spec_On_Traces*(...)**

- For each operator $\Longrightarrow$ a proven soundness correctness theorem.

Correctness of the proposed formalisation

Soundness is expressed on each defined operator

- A template for proving soundness

THEORY Theo4[PO]Correctness **IMPORT** EvtBTraces, Theo4[PO]
TYPE PARAMETERS *STATE*, *EVENT*
THEOREMS
thm_of_Correctness_of_[PO]: $\forall m, tr \cdot m \in Machine(STATE, Ev) \land$
$check\_Machine\_Consistency(m) \land IsATrace(tr, m) \land$ **[PO](m,args)**
$\Rightarrow$ ***PO_Spec_On_Traces(...)***

- For each operator $\Longrightarrow$ a proven soundness correctness theorem.
- Case of Divergence of P.

THEOREMS
thm_of_Correctness_of_DIvergence: $\forall m, tr, v, \hat{P} \cdot v \in STATE \to \mathbb{Z} \land$
$m \in Machine(STATE, EVENT) \land check\_Machine\_Consistency(m) \land IsATrace(m, tr) \land$
$dom(tr) = \mathbb{N} \land$ **TLDivergence($m, \hat{P}, v$)**
$\Rightarrow$ **($\exists i \cdot i \in dom(tr) \land (\forall j \cdot j \geq i \land j \in dom(tr) \land tr(j) \in \hat{P})$)**
...

Divergent in P

The Sequent Rule of $\nearrow$	Associated Operator in EB4EB
$\nearrow P \equiv \forall x, v, v' \cdot$ $(\neg P(v) \land G(x,v) \Rightarrow V(v) \in \mathbb{N}) \land$ $(\neg P(v) \land G(x,v) \land A(x,v,v') \Rightarrow$ $V(v') < V(v)) \land$ $(P(v) \land G(x,v) \land A(x,v,v') \land$ $V(v') \in \mathbb{N} \Rightarrow$ $V(v') \leq V(v))$	**TLDivergent_In_P** <**predicate**> $(m : Machine(STATE, Ev),$ $\hat{P} : \mathbb{P}(St), v : \mathbb{P}(St \times \mathbb{Z}))$ **well—definedness** $v \in St \to \mathbb{Z}$ **direct definition** $TLConvergent\_In\_P(m, St \setminus \hat{P}, v) \land$ $\forall e \cdot e \in Progress(m) \Rightarrow ((\forall s, s' \cdot$ $s \in Inv(m) \land s \in \hat{P} \land s \in Grd(m)[\{e\}]$ $\land s' \in BAP(m)[\{e\}][\{s\}] \land v(s') \in \mathbb{N}$ $\Rightarrow v(s') \leq v(s)))$
(a)	(b)

Divergent in P

The Sequent Rule of $\nearrow$	Associated Operator in EB4EB
$\nearrow P \equiv \forall x, v, v' \cdot$ $(\neg P(v) \land G(x,v) \Rightarrow V(v) \in \mathbb{N}) \land$ $(\neg P(v) \land G(x,v) \land A(x,v,v') \Rightarrow$ $V(v') < V(v)) \land$ $(P(v) \land G(x,v) \land A(x,v,v') \land$ $\Rightarrow$ $V(v') \leq V(v))$	**TLDivergent_In_P** <**predicate**> $(m : Machine(STATE, Ev),$ $\hat{P} : \mathbb{P}(St), v : \mathbb{P}(St \times \mathbb{Z}))$ **well—definedness** $v \in St \to \mathbb{Z}$ **direct definition** $TLConvergent\_In\_P(m, St \setminus \hat{P}, v) \land$ $\forall e \cdot e \in Progress(m) \Rightarrow ((\forall s, s' \cdot$ $s \in Inv(m) \land s \in \hat{P} \land s \in Grd(m)[\{e\}]$ $\land s' \in BAP(m)[\{e\}][\{s\}] \land$ $\Rightarrow v(s') \leq v(s)))$
(a)	(b)

Divergent in P

The Sequent Rule of $\nearrow$	Associated Operator in EB4EB
$\nearrow P \equiv \forall x, v, v' \cdot$ $(\neg P(v) \land G(x, v) \Rightarrow V(v) \in \mathbb{N}) \land$ $(\neg P(v) \land G(x, v) \land A(x, v, v') \Rightarrow$ $V(v') < V(v)) \land$ $(P(v) \land G(x, v) \land A(x, v, v') \land$ $\Rightarrow$ $V(v') \leq V(v))$	**TLDivergent_In_P** <**predicate**> $(m : Machine(STATE, Ev),$ $\hat{P} : \mathbb{P}(St), v : \mathbb{P}(St \times \mathbb{Z}))$ **well—definedness** $v \in St \to \mathbb{Z}$ **direct definition** $TLConvergent\_In\_P(m, St \setminus \hat{P}, v) \land$ $\forall e \cdot e \in Progress(m) \Rightarrow ((\forall s, s' \cdot$ $s \in Inv(m) \land s \in \hat{P} \land s \in Grd(m)[\{e\}]$ $\land s' \in BAP(m)[\{e\}][\{s\}] \land$ $\Rightarrow v(s') \leq v(s)))$
(a)	(b)

$\forall v, v' \cdot v \in \mathbb{N} \land v' \notin \mathbb{N} \Rightarrow v \geq v'$ Not true for lexicographic variant

Proof Process

Previous proof strategy

- Used the variant as bound in number of step

Proof Process

Previous proof strategy

- Used the variant as bound in number of step

Update proof strategy

- Variant can not be used as a bound in trace
- Based on well-founded properties (Abrial Event-B theory)

Conclusion

Conclusion

- Update EB4EB with lexicographic Variant
- Update and adapt Temporal properties
- Proof the soundness of Temporal properties.

Peter Riviere, Neeraj Kumar Singh, and Yamine Aït Ameur.

Reflexive Event-B: Semantics and Correctness the EB4EB Framework.

IEEE Transactions on Reliability, pages 1–16, 2022.

Peter Riviere, Neeraj Kumar Singh, Yamine Aït Ameur, and Guillaume Dupont.

Formalising liveness properties in event-b with the reflexive EB4EB framework.

In NFM, volume 13903 of Lecture Notes in Computer Science, pages 312–331. Springer, 2023.