Modelling Software-based Systems
Lecture 1
The Modelling Language Event-B

Master Informatique

Dominique Méry
Telecom Nancy,Université de Lorraine

17 septembre 2025

dominique.mery@loria.fr

General Summary

Current Summary

Expressing models in the event B notation

- Models are defined in two ways :
  - an abstract machine
  - a refinement of an existing model
- Models use **constants** which are defined in structures called **contexts**
- B structures are related by the three possible relations :
  - the **sees** relationship for expressing the use of constants, sets satisfying axioms and theorems.
  - the **extends** relationship for expressing the extension of contexts by adding new constants and new sets
  - the **refines** relationship stating that a B model is refined by another one.

Machines and contexts

Machines

- REFINES
- SEES a context
- VARIABLES of the model
- INVARIANTS satisfied by the variables
- THEOREMS satisfied by the variables
- EVENTS modifying the variables
- VARIANT

Contexts

- EXTENDS another context
- SETS declares new sets
- CONSTANTS define a list of constants
- AXIOMS define the properties of constants and sets
- THEOREMS list the theorems which should be derived from axioms

$$
\begin{array}{|l|}
\hline
\textrm{MACHINE} \\
\quad m \\
\textrm{REFINES} \\
\quad am \\
\textrm{SEES} \\
\quad c \\
\textrm{VARIABLES} \\
\quad u \\
\textrm{INVARIANTS} \\
\quad I(u) \\
\textrm{THEOREMS} \\
\quad Q(u) \\
\textrm{VARIANT} \\
\quad < variant > \\
\textrm{EVENTS} \\
\quad < event > \\
\textrm{END} \\
\hline
\end{array}
$$

- $\Gamma(m)$: environment for the machine m defined by the context c
- $\Gamma(m) \vdash \forall u \in \textsc{Values} : \textsc{Init}(u) \Rightarrow \mathrm{I}(u)$
- For each event e in E :
  $\Gamma(m) \vdash \forall u, u' \in \textsc{Values} : \mathrm{I}(x) \wedge BA(e)(u, u') \Rightarrow \mathrm{I}(u')$
- $\Gamma(m) \vdash \forall u \in \textsc{Values} : \mathrm{I}(u) \Rightarrow \mathrm{Q}(u)$

Contexts in Event B

CONTEXTS
  c
EXTENDS
  ac
SETS
  s
CONSTANTS
  k
AXIOMS
  $ax1 : \ldots$
THEOREMS
  $th1 : \ldots$
END

- ac : c is extending ac and add new features
- s : sets are defined either by intension or by extension
- k : constants are defined and
- axioms characterize constants and sets
- theorems are derived from axioms in the current context

Events

before-after relation for e

For each event e, a before-after relation is defined over (flexible) variables. Three events are possible

- e $\triangleq$ $\mathrm{BEGIN}\ x :| P(x,x')\ \mathrm{END}$: $\mathsf{BA}(\mathsf{e})(x,x') \triangleq P(x;x')$
- e $\triangleq$ $\mathrm{WHEN}\ G(x)\ \mathrm{THEN}\ x :| P(x,x')\ \mathrm{END}$:
  $\mathsf{BA}(\mathsf{e})(x,x') \triangleq G(x) \wedge P(x;x')$
- e $\triangleq$ $\mathrm{ANY}\ p\ \mathrm{WHEN}\ G(p,x)\ \mathrm{THEN}\ x :| P(p,x,x')\ \mathrm{END}$:
  $\mathsf{BA}(\mathsf{e})(x,x') \triangleq \exists p.G(p,x) \wedge P(x;x')$

Guards of event

guard for e

For each event e, a guard is defined over (flexible) variables.
Three events are possible

- e $\triangleq$ $\text{BEGIN } x :| P(x, x') \text{ END}$: grd$(x) \triangleq TRUE$
- e $\triangleq$ $\text{WHEN } G(x) \text{ THEN } x :| P(x, x') \text{ END}$: grd(e)$(x) \triangleq G(x)$
- e $\triangleq$ $\text{ANY } p \text{ WHEN } G(p, x) \text{ THEN } x :| P(p, x, x') \text{ END}$: grd(e)$(x) \triangleq \exists p.G(p, x)$

Proof obligations for a B model

inv1 $\Gamma(s,c) \vdash Init(x) \Rightarrow I(x)$

inv2 $\Gamma(s,c) \vdash I(x) \wedge BA(e)(x,x') \Rightarrow I(x')$

fis $\Gamma(s,c) \vdash I(x) \wedge \mathsf{grd}\,(E) \Rightarrow \exists x' \cdot P(x,x')$

safe $\Gamma(s,c) \vdash I(x) \Rightarrow A(x)$

dead $\Gamma(s,c) \vdash I(x) \Rightarrow (\mathsf{grd}(e_1) \vee \ldots \mathsf{grd}(e_n))$

The factorial model

CONTEXT
$\quad fonctions$
CONSTANTS
$\quad factorial, n$
AXIOMS

$$ax1 : n \in \mathbb{N}$$
$$ax2 : factorial \in \mathbb{N} \leftrightarrow \mathbb{N}$$
$$ax3 : 0 \mapsto 1 \in factorial$$
$$ax4 : \forall (i, fn).(i \mapsto fn \in factorial \Rightarrow i+1 \mapsto (i+1) * fi \in factorial) \land$$
$$\forall f \cdot \left(\begin{array}{l} f \in \mathbb{N} \leftrightarrow \mathbb{N} \land \\ 0 \mapsto 1 \in f \land \\ \forall (n, fn).(n \mapsto fn \in f \Rightarrow n+1 \mapsto (n+1) \times fn \in f) \\ \Longrightarrow \\ factorial \subseteq f \end{array} \right)$$

END

Current Summary

The factorial model

MACHINE
  $specification$
SEES $fonctions$
VARIABLES
  $resultat$
INVARIANT
  $resultat \in \mathbb{N}$
THEOREMS
  $th1 : factorial \in \mathbb{N} \longrightarrow \mathbb{N}$;
  $th2 : factorial(0) = 1$;
  $th3 : \forall n.(n \in \mathbb{N} \Rightarrow factorial(n+1) = (n+1) \times factorial(n))$
INITIALISATION
  $resultat :\in \mathbb{N}$
EVENTS
  $computing1 =$ BEGIN $resultat := factorial(n)$ END
END

Communications between agents

MACHINE $agents$
SEES $data$
VARIABLES
$sent$
got
$lost$
INVARIANTS
$inv1 : sent \subseteq AGENTS \times AGENTS$
$inv2 : got \subseteq AGENTS \times AGENTS$
$inv4 : (got \cup lost) \subseteq sent$
$inv6 : lost \subseteq AGENTS \times AGENTS$
$inv7 : got \cap lost = \varnothing$

INITIALISATION
BEGIN
$act1 : sent := \varnothing$
$act2 : got := \varnothing$
$act4 : lost := \varnothing$
END

Communications between agents

EVENT sending a message
ANY
  a, b
WHERE
  $grd11 : a \in AGENTS$
  $grd12 : b \in AGENTS$
  $grd1 : a \mapsto b \notin sent$
THEN
  $act11 : sent := sent \cup \{a \mapsto b\}$
END

EVENT getting a message
ANY
  a, b
WHERE
  $grd11 : a \in AGENTS$
  $grd12 : b \in AGENTS$
  $grd13 : a \mapsto b \in sent \setminus (got \cup lost)$
THEN
  $act11 : got := got \cup \{a \mapsto b\}$
END

Communications between agents

EVENT loosing a messge
ANY
a
b
WHERE $grd1 : a \in AGENTS$
$grd2 : b \in AGENTS$
$grd3 : a \mapsto b \in sent \setminus (got \cup lost)$
THEN
$act1 : lost := lost \cup \{a \mapsto b\}$
END

CONTEXTS
$data$
SETS
$MESSAGES$
$AGENTS$
$DATA$
CONSTANTS
n
$infile$
AXIOMS
$axm1 : n \in \mathbb{N}$
$axm2 : n \neq 0$
$axm3 : infile \in 1 \,..\, n \rightarrow DATA$
END

Current Summary

General form of an event

EVENT e
  ANY t
  WHERE
    $G(c, s, t, x)$
  THEN
    $x :| (P(c, s, t, x, x'))$
  END

- c et s are constantes and visible sets by e
- x is a state variable or a list of variabless
- $G(c, s, t, x)$ is the condition for observing e.
- $P(c, s, t, x, x')$ is the assertion for the relation over x and x'.
- $BA(e)(c, s, x, x')$ is the *before-after* relationship for e and is defined by $\exists t. G(c, s, t, x) \ \wedge \ P(c, s, t, x, x')$.

General form of proof obligations for an event e

Proofs obligations are simplified when they are generated by the module called POG and goals in sequents as $\Gamma \vdash G$:

1. $\Gamma \vdash G_1 \wedge G_2$ is decomposed into the two sequents $\begin{array}{l}(1)\Gamma \vdash G_1\\(2)\Gamma \vdash G_2\end{array}$
2. $\Gamma \vdash G_1 \Rightarrow G_2$ is transformed into the sequent $\Gamma, G_1 \vdash G_2$

Proof obligations in Rodin

- *INIT/I/INV* : $C(s,c), INIT(c,s,x) \vdash I(c,s,x)$
- *e/I/INV* : $C(s,c), I(c,s,x), G(c,s,t,x), P(c,s,t,x,x') \vdash I(c,s,x')$
- *e/act/FIS* : $C(s,c), I(c,s,x), G(c,s,t,x) \vdash \exists x'.P(c,s,t,x,x')$

notation

- Chapter Event B
- The Event B Modelling Notation Version 1.4
- The Event-B Mathematical Language 2006
- User Manual of the RODIN PLatform