A (short) summary on B and case studies on protocols

Dominique Méry

26th IFIP WG 6.1 International Conference on Formal Methods for Networked and Distributed Systems
September 26-29 2006, Paris, France

Summary on the tools and on the notations

- ✓ Atelier B supports the classical B method: MACHINE, OPERATIONS, REFINEMENT, PROOFS,
- ✓ The event-based B method is not yet supported by a tool
- ✓ Atelier B and B4free are used to generate proof obligations for the invariant preservation
- ✓ Additional proof obligations are added in the invariant
- ✓ An Events System Model: events which are triggered
- ✓ An Abstract Machine: operations which are called

Summary: Set-theoretical notations

Name	Syntax	Definition
Binary Relation	$s \leftrightarrow t$	$\mathcal{P}(s \times t)$
Composition of relations	$r_1 ; r_2$	$\{x, y \mid x \in a \ \wedge \ y \in b \ \wedge \ \exists z.(z \in c \ \wedge \ x, z \in r_1 \ \wedge \ z, y \in r_2)\}$
Inverse relation	r^{-1}	$\{x, y \mid x \in \mathcal{P}(a) \ \wedge \ y \in \mathcal{P}(b) \ \wedge \ y, x \in r\}$
Domain	$\mathsf{dom}(r)$	$\{a \mid a \in s \ \wedge \ \exists b.(b \in t \ \wedge \ a \mapsto b \in r)\}$
Range	$\mathsf{ran}(r)$	$\mathsf{dom}(r^{-1})$
Identity	$\mathsf{id}(s)$	$\{x, y \mid x \in s \ \wedge \ y \in s \ \wedge \ x = y\}$
Restriction	$s \lhd r$	$\mathsf{id}(s) ; r$
Co-restriction	$r \rhd s$	$r ; \mathsf{id}(s)$
Anti-restriction	$s \lhd\!\!\!- r$	$(\mathsf{dom}(r) - s) \lhd r$
Anti-co-restriction	$r \rhd\!\!\!- s$	$r \rhd (\mathsf{ran}(r) - s)$
Image	$r[w]$	$\mathsf{ran}(w \lhd r)$
Overriding	$q \lhd\!\!\!+ r$	$(\mathsf{dom}(r) \lhd\!\!\!- q) \cup r$
Partial Function	$s \nrightarrow t$	$\{r \mid r \in s \leftrightarrow t \ \wedge \ (r^{-1} ; r) \subseteq \mathsf{id}(t)\}$

Summary: events

Event : E	Before-After Predicate
BEGIN $x : P(x_0, x)$ **END**	$P(x, x')$
SELECT $G(x)$ **THEN** $x : P(x_0, x)$ **END**	$G(x) \quad \wedge \quad P(x, x')$
ANY t **WHERE** $G(t, x)$ **THEN** $x : P(x_0, x, t)$ **END**	$\exists\, t \cdot (\, G(t, x) \quad \wedge \quad P(x, x', t)\,)$

Summary: tools

- ✓ Proving lemmas or theorems from definitions of mathematical structures (constants and properties)
- ✓ **Stop for the demo**
- ✓ Proving proof obligations generated from an events system model

Summary: Sequent Calculus

✓ Sequent:

$$HY\,P \;\vdash\; P$$

✓ Rules using sequents:

$$\frac{\begin{array}{c} HY\,P_1 \;\vdash\; P_1 \\ \bullet \\ \bullet \\ \bullet \\ HY\,P_n \;\vdash\; P_n \end{array}}{HY\,P \;\vdash\; P}$$

Notations

	Antecedents	Consequent
name	$\begin{cases} HY\,P_1 \vdash P_1 \\ \vdots \\ HY\,P_n \vdash P_n \end{cases}$	$HY\,P \vdash P$

$$Predicate \wedge Predicate$$

$$Predicate \Rightarrow Predicate$$

$$\neg Predicate$$

Basic rules

	Antecedents	Consequent
BR1		$P \vdash P$
BR2	$HYP \vdash P$ $HYP \subseteq HYP'$	$HYP' \vdash P$
BR3	$P \in HYP$	$HYP \vdash P$
BR4	$HYP \vdash P$ $HYP, P \vdash Q$	$HYP \vdash Q$

Basic rules for $\wedge$

	Antecedents	Consequent
R1	$\begin{cases} HYP \vdash P \\ HYP \vdash Q \end{cases}$	$HYP \vdash P \wedge Q$
$R2$ $R2'$	HYP $\vdash P \wedge Q$	$\begin{cases} HYP \vdash P \\ HYP \vdash Q \end{cases}$

Basic rules for $\Rightarrow$

	Antecedents	Consequent
DED	$HYP, P \vdash Q$	$HYP \vdash P \Rightarrow Q$
R4	$HYP \vdash P \Rightarrow Q$	$HYP, P \vdash Q$
MP	$\begin{cases} HYP \vdash P \\ HYP \vdash P \Rightarrow Q \end{cases}$	$HYP \vdash Q$

DED (R3) = DEDUCTION RULE

MP = Modus Ponens (BR4 + R4)

Basic rules for $\neg$

	Antecedents	Consequent
R5	$\begin{cases} HYP, \neg Q \vdash P \\ HYP, \neg Q \vdash \neg P \end{cases}$	HYP $\vdash Q$
R6	$\begin{cases} HYP, Q \vdash P \\ HYP, Q \vdash \neg P \end{cases}$	$HYP \vdash \neg Q$

RULES OF CONTRADICTION “reductio ad absurdum”)

A proof

$$HYP \vdash P \Rightarrow \neg\neg P$$ Sequent 1

Apply DED (R3)

$$HYP, P \vdash \neg\neg P$$ Sequent 2

Apply R6 with X

$$HYP, P, \neg P \vdash X$$

$$HYP, P, \neg P \vdash \neg X$$

X is P and we use BR3

$$HYP, P, \neg P \vdash P$$ Sequent 3

$$HYP, P, \neg P \vdash \neg P$$ Sequent 4

Automating proofs obligations checking

- B tools provide automatic proof procedure for sequent calculus
- pr
- pp (predicate prover)
- Atelier B (ClearSy): `http://www.atelierb.societe.com/`
- Click'n'Prove: `http://www.loria.fr/~cansell/cnp.html`
- B4free (ClearSy): `http://www.b4free.com/`

Constructing a protocol

- Paradigms: parachutist, myope, ...
- A protocol is a process which manages a communication between partners
- The main purpose is to ensure the communication of an item or items from an agent to another agent
- Finding a first very abstract model with main events
- Refining abstract model to take into account assumptions on communications
- Localization of variables.

Protocols

- FIFO Protocol without loss of messages, duplication and reordering
- FIFO Protocol without loss of messages and duplication and with reordering
- Data Transfer Protocol of Stenning and more . . .

FIFO Protocol: no loss, duplication and reordering

MACHINE

$fifoprotocol0$

SETS

DATA

CONSTANTS

NN, *FILE*

PROPERTIES

$NN \in \mathbb{N} \wedge$

$NN \neq 0 \wedge$

$FILE \in 1..NN \longrightarrow DATA$

VARIABLES

file

INVARIANT

$file \in 1..NN \nrightarrow DATA$

INITIALISATION

$file := \emptyset$

OPERATIONS

TRANSMISSION = **BEGIN** *file* := *FILE* **END**;

sndd = **BEGIN** *file* $\in$ (*file* $\in$ 1..*NN* $\nrightarrow$ *DATA*) **END**;

rcvd = **BEGIN** *file* $\in$ (*file* $\in$ 1..*NN* $\nrightarrow$ *DATA*) **END**

END

FIFO Protocol: no loss, duplication and reordering

REFINEMENT $fifoprotocol1$

REFINES $fifoprotocol0$

VARIABLES *ss*, *rr*, *file*, *dch*

INVARIANT

$$ss \in \mathbb{N} \wedge rr \in \mathbb{N} \wedge$$

$$file \in 1..NN \pfun DATA \wedge$$

$$dch \in 1..NN \pfun DATA \wedge$$

$$rr \leq ss \wedge$$

$$file = (1..rr) \domres FILE \wedge$$

$$dch \subseteq (1..ss-1) \domres FILE \wedge$$

$$file \subseteq dch \wedge ss \leq NN+1 \wedge$$

$$ss \geq 1 \wedge rr \leq NN$$

ASSERTIONS

$$rr = NN \Rightarrow file = FILE$$

INITIALISATION

$$file := \emptyset \parallel ss := 1 \parallel rr := 0 \parallel dch := \emptyset$$

OPERATIONS

TRANSMISSION = **SELECT** *rr* *NN* **THEN** SKIP **END**;

sndd = **SELECT** $ss \leq NN$

THEN $dch(ss) := FILE(ss) \parallel$

$ss := ss+1$ **END**;

rcvd = **SELECT** $rr+1 \in \mathrm{DOM}(dch)$

THEN $file(rr+1) := dch(rr+1) \parallel$

$rr := rr+1$ **END**

END

FIFO Protocol: no loss and duplication and with re-ordering

MACHINE

$reliable0$

SETS

DATA

CONSTANTS

NN, *FILE*

PROPERTIES

$NN \in \mathbb{N} \wedge$

$NN \neq 0 \wedge$

$FILE \in 1..NN \longrightarrow DATA$

VARIABLES

file

INVARIANT

$file \in 1..NN \nrightarrow DATA$

INITIALISATION

$file := \emptyset$

OPERATIONS

TRANSMISSION = **BEGIN** *file* := *FILE* **END**;

sndd = **BEGIN** *file* $\in$ (*file* $\in$ 1..*NN* $\rightarrow$ *DATA*) **END**;

rcvd = **BEGIN** *file* $\in$ (*file* $\in$ 1..*NN* $\rightarrow$ *DATA*) **END**

END

FIFO Protocol: no loss and duplication and with re-ordering

REFINEMENT $reliable1$

REFINES $reliable0$

VARIABLES

$ss, file, dch$

INVARIANT

$file \in 1..NN \pfun DATA \land$

$dch \subseteq (1..NN) \times DATA \land$

$ss \in \mathbb{N} \land$

$ss \geq 1 \land$

$ss \leq NN+1 \land$

$dch \cup file = (1..(ss-1)) \domres FILE$

INITIALISATION

$file := \emptyset \parallel dch := \emptyset \parallel ss := 1$

OPERATIONS

TRANSMISSION = **SELECT** $ss = NN+1 \land dch = \emptyset$ **THEN** SKIP **END**;

$sndd$ = **SELECT** $ss \leq NN$

THEN $dch := dch \cup \{(ss \mapsto FILE(ss))\} \parallel$

$ss := ss+1$ **END**;

$rcvd$ = **ANY** rr, mes

WHERE $rr \in \mathbb{N} \land mes \in DATA \land rr \mapsto mes \in dch$

THEN $file(rr) := mes \parallel$

$dch := dch - \{rr \mapsto mes\}$ **END**

END

Data transfer protocol Stenning

- Goal: transfer data file from an agent to another agent
- an agent sender and an agent receiver
- Both agents **S** and **R** are localized on two differents sites

First model

◇ The file is a total function over $\{1, \ldots, n\}$ into $DATA$.

$n \in \mathbb{N} \wedge$ size of file

$FILE \in 1..n \longrightarrow DATA$ file to send

◇ The result $file$ is a partial function from $\{1, \ldots, n\}$ into $DATA$.

VARIABLES

file

INVARIANT

$file \in 1..n \longrightarrow DATA$

First model

$$transmission = \textbf{BEGIN}\ file := FILE\ \textbf{END}$$

Second model: refining the data transfer

- Data are sent one by one and one assumes that no datum is lost

- Two new variables are introduced for controling the sending and the receiving actions.

VARIABLES

s, r	control of messages traffic

- $FILE$ is progressively transmitted and $file$ contains a part of $FILE$, during the protocol.

INVARIANT

$$file = (1..r) \lhd FILE$$

Second model: refining the data transfer

- Communication channel between S and R is modelles by a variable dch and a confirmation channel between R and S allows us to synchronise sent data.

VARIABLES

dch, *ach*

INVARIANT

$dch \subseteq (1..s) \lhd FILE \land$

$ach \subseteq 1..r$

Events

- Initial state:

INITIALISATION

$$file := \emptyset \parallel ss := 1 \parallel rr := 0 \parallel dch := \emptyset \parallel ach := \emptyset$$

- Events $sndd$, $rcvd$, $snda$, $rcva$are new and model the effective communications.

- Event $TRANSMISSION$ models the end of the transmission.

Events for sending data

$$sndd \;=\; \textbf{SELECT}\; ss \leq NN \;\textbf{THEN}\; dch(ss) \;:=\; FILE(ss)\; \textbf{END};$$

$$rcvd \;=\; \textbf{SELECT}\; rr+1 \in \mathrm{DOM}(dch)$$

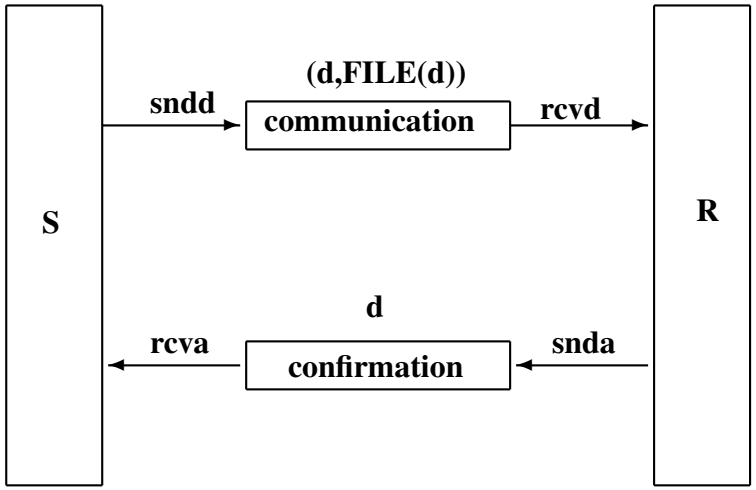
$$\textbf{THEN}\; file(rr+1) \;:=\; dch(rr+1) \parallel rr \;:=\; rr+1\; \textbf{END};$$

Events for sending acknowledgments

$snda = \textbf{SELECT}\ rr \neq 0\ \textbf{THEN}\ ach := ach \cup \{rr\}\ \textbf{END};$

$rcva = \textbf{SELECT}\ ss \in ach\ \textbf{THEN}\ ss := ss + 1\ \textbf{END};$

View of the second model



Second refinement and third model

- Channels are not reliable!
- Data may be reordered, duplicated and destroyed!
  - A channel is a set: duplication
  - Daemons remove data

Daemons

$$rmvd \;=\; \textbf{ANY}\ ii, dd\ \textbf{WHERE}\ ii \mapsto dd \in dch\ \textbf{THEN}\ dch := dch - \{ ii \mapsto dd \}\ \textbf{END};$$

$$rmva \;=\; \textbf{ANY}\ ii\ \textbf{WHERE}\ ii \in ach\ \textbf{THEN}\ ach := ach - \{ ii \}\ \textbf{END}$$

Invariant of the final model

INVARIANT

$$
\begin{aligned}
& ss \in \mathbb{N} \wedge \\
& rr \in \mathbb{N} \wedge \\
& file \in 1..NN \pfun DATA \wedge \\
& dch \in 1..NN \pfun DATA \wedge \\
& rr \leq ss \wedge \\
& ss \leq rr + 1 \wedge \\
& file = (1..rr) \domres FILE \wedge \\
& dch \subseteq (1..ss) \domres FILE \wedge \\
& ach \subseteq 1..rr \wedge \\
& ss \leq NN + 1 \wedge \\
& ss \geq 1 \wedge \\
& rr \leq NN
\end{aligned}
$$

Safety properties of the last model

ASSERTIONS

$$rr = NN \Rightarrow file = FILE;$$

$$rr \leq ss\,;$$

$$ss \leq rr + 1\,;$$

$$rr = ss \Rightarrow rr \neq 0;$$

$$\forall\,(mm, kk).(kk \mapsto mm \in dch \Rightarrow kk \leq ss);$$

$$\forall\,kk.(kk \in 1..NN \wedge kk \in ach \Rightarrow kk \leq rr)$$

Questions

- How to get the alternating bit protocol?
- How to model protocols related to WEB services?
- What about probablilistic algorithms?
- **Next the leader election**