Event B: Sets, Relations, Functions, Arithmetic[1]

Manuel Carro
manuel.carro@upm.es

Universidad Politécnica de Madrid &
IMDEA Software Institute

[1] With material from J. R. Abrial book *Modeling in Event-B: system and software engineering*.

Set theory: membership

For a complete reference and succinct but rigorous definitions of all the constructions presented in these slides, please check the Event B mathematical toolit

- Event-B formal reasoning is built based on:
  - First-order logic inference rules (seen).
  - Set theory (to be briefly reviewed now).
- Set theory as a foundation for relations, functions (and, therefore, data structures).
  - Proofs often reduced to proving goals on sets.

Set theory: membership

- A set is a well-defined collection of distinct objects.
- Set theory is based on the membership predicate

$$E \in S$$

- E is an expression, S is a set.

There are three basic constructs in set theory, defined by equivalences.
S and T are sets, x is a variable, P is a predicate, F is an expression.

Cartesian product: $S \times T$

$$E \mapsto F \in S \times T \equiv E \in S \land F \in T$$

Powerset: $\mathbb{P}(T)$

$$S \in \mathbb{P}(T) \equiv \forall x \cdot x \in S \Rightarrow x \in T$$

Comprehension:

Version 1: $\{x \mid x \in S \land P(x)\}$

$$E \in \{x \mid x \in S \land P(x)\} \equiv E \in S \land P(E)$$

Version 2: $\{x \cdot x \in S \land P(x) \mid F(x)\}$

$$E \in \{x \cdot x \in S \land P(x) \mid F(x)\} \equiv \exists x \cdot x \in S \land P(x) \land E = F(x)$$

Set theory: basic constructs

Examples

$$
\begin{aligned}
\{1,2,3\} \times \{a,b\} &= \{1 \mapsto a, 1 \mapsto b, 2 \mapsto a, 2 \mapsto b, 3 \mapsto a, 3 \mapsto b\} \\
\mathbb{P}(\{1,2,3\}) &= \{\{1,2,3\},\{1,2\},\{1,3\},\{2,3\},\{1\},\{2\},\{3\},\varnothing\} \\
\{x \mid x \in \{2,3,4,5\} \wedge x \mod 2 = 0\} &= \{2,4\} \\
\{x \cdot x \in \{2,3,4,5\} \wedge x \mod 2 = 1 \mid x^2\} &= \{25,9\}
\end{aligned}
$$

Reminder: $A \mapsto B$ is a tuple.
It is sometimes written as (A, B) in other formalisms.

Shortcut: $m..n \equiv \{x \in \mathbb{Z} \mid m \leq x \wedge x \leq n\}$

- $\{x \mid x \in \mathbb{N} \wedge x < 2\} \times 8..10$
- $\{x \cdot x \in 3..5 \mid x \mapsto x * x\}$
- $\{n \cdot n \in \mathbb{N} \mid (0..n) \mapsto n\}$

Operations on sets

$$
\begin{aligned}
S \subseteq T &\equiv S \in \mathbb{P}(T) \\
S = T &\equiv S \subseteq T \wedge T \subseteq S \\
S \cup T &\equiv \{x \mid x \in S \vee x \in T\} \\
S \cap T &\equiv \{x \mid x \in S \wedge x \in T\} \\
S \setminus T &\equiv \{x \mid x \in S \wedge x \notin T\} \\
E \in \{a, \ldots, z\} &\equiv E = a \vee \ldots \vee E = z \\
E \in \varnothing &\equiv \perp
\end{aligned}
$$

- Operators based on membership and logic operations.
- Note: $E \notin T \equiv \neg(E \in T)$.
- Also: generalized / conditional union and intersection (see reference cards).

Binary relations

- A binary relation r is a subset of their Cartesian product: $r \subseteq S \times T$
- Different syntax to highlight structure.
- $S \leftrightarrow T$ is the set of all possible relations between S and T.
  - r would be one of them: $r \in S \leftrightarrow T$.
  - $S \leftrightarrow T = \mathbb{P}(S \times T)$
- $r \in 1..3 \leftrightarrow 7..11$
  - $r = \{1 \mapsto 10, 2 \mapsto 7, 2 \mapsto 11\}$
  - $4 \mapsto 10 \notin r$

$$
\begin{array}{rcl}
x \in dom(r) & \equiv & \exists y \cdot x \mapsto y \in r \\
y \in ran(r) & \equiv & \exists x \cdot x \mapsto y \in r \\
r^{-1} & \equiv & \{y \mapsto x \mid x \mapsto y \in r\}
\end{array}
$$

- $r \in \{\text{meat}, \text{fish}, \text{pasta}, \text{bacon}\} \leftrightarrow \{\text{carbs}, \text{protein}, \text{fat}\}$ – write a couple of relations.
- $dom(r)$, $ran(r)$, relation with S and T
- How many different r may there be?

Types of relations

Total	$S \twoheadleftarrow\!\!\!\rightarrow T$	$r \in S \leftrightarrow T \wedge dom(r) = S$
Surjective	$S \leftarrow\!\!\!\twoheadrightarrow T$	$r \in S \leftrightarrow T \wedge ran(r) = T$
Both	$S \twoheadleftarrow\!\!\!\twoheadrightarrow T$	$r \in S \twoheadleftarrow\!\!\!\rightarrow T \wedge r \in S \leftarrow\!\!\!\twoheadrightarrow T$

Hint: sets and relations are very useful modeling tools!

Operations on relations

Domain restriction	$S \lhd r$	$\{x \mapsto y \in r \mid x \in S\}$
Domain subtraction	$S \mathbin{\lhd\!\!\!-} r$	$\{x \mapsto y \in r \mid x \notin S\}$
Range restriction	$r \rhd T$	$\{x \mapsto y \in r \mid y \in T\}$
Range subtraction	$r \mathbin{\rhd\!\!\!\!-} T$	$\{x \mapsto y \in r \mid y \notin T\}$

Assume $Prey \in Animal \leftrightarrow Animal$.

We mean hunter $\mapsto$ *hunted. The syntax of the relation does not reveal its intended semantics.*

- $Mammal \lhd Prey$
- $Mammal \mathbin{\lhd\!\!\!-} Prey$
- $Prey \rhd Spiders$
- $Fish \lhd (Prey \rhd Spiders)$
- $Spiders \mathbin{\lhd\!\!\!-} (Prey \rhd Spiders)$

Operations on relations

Image	$p[S]$	$\{y \mid x \mapsto y \in p \wedge x \in S\}$

if $p = \{a \mapsto 1, b \mapsto 2, c \mapsto 3, d \mapsto 4\}$ and $S = \{b, c\}$ then $p[S] = \{2, 3\}$

Composition	$p; q$	$\{x \mapsto z \mid x \mapsto y \in p \wedge y \mapsto z \in q\}$

if $q = \{1 \mapsto 1, 2 \mapsto 4, 3 \mapsto 9, 4 \mapsto 16\}$ then $p; q = \{a \mapsto 1, b \mapsto 4, c \mapsto 9, d \mapsto 16\}$

Identity	$id(S)$	$\{x \mapsto x \mid x \in S\}$

if $S = \{a, b, c\}$ then $id(S) = \{a \mapsto a, b \mapsto b, c \mapsto c\}$

Overriding	$p \mathbin{\lhd\!\!\!+} q$	$(dom(q) \mathbin{\lhd\!\!\!-} p) \cup q$

$p \mathbin{\lhd\!\!\!+} \{a \mapsto -1, c \mapsto -3, e \mapsto -5\} = \{a \mapsto -1, b \mapsto 2, c \mapsto -3, d \mapsto 4, e \mapsto -5\}$

Some useful results, definitions

$$
\begin{aligned}
(r^{-1})^{-1} &= r \\
\mathrm{dom}(r^{-1}) &= \mathrm{ran}(r) \\
(S \lhd r)^{-1} &= r^{-1} \rhd S \\
(p; q)^{-1} &= q^{-1}; p^{-1} \\
p; (q; r) &= (p; q); r \\
p; (q \cup r) &= (p; q) \cup (p; r) \\
(p; q)[S] &= q[p[S]] \\
r[S \cup T] &= r[S] \cup r[T]
\end{aligned}
$$

$r = r^{-1}$	symmetric
$r \cap r^{-1} = \varnothing$	asymmetric
$\mathrm{id}(S) \subseteq r$	reflexive
$r; r \subseteq r$	transitive

Set-theoretic notation more readable than predicate calculus

$$r = r^{-1} \equiv \forall x, y \cdot x \in S \land y \in S \Rightarrow (x \mapsto y \in r \Leftrightarrow y \mapsto x \in r)$$

Functions

- Functions: one type of relations.
- Every element in the domain relates to one element in the range only:

$$x \mapsto y \in f \land x \mapsto z \in f \Rightarrow y = z$$

- Notation:
  - $f(x) = y \equiv x \mapsto y \in f$.
  - $f(x) := y \;\equiv\; f := f \mathbin{<\!\!\!+} \{x \mapsto y\}$
- WD conditions for $f(x)$:
  - $f \in S \mathrel{+\!\!\!\!\!\rightarrow} T$
  - $x \in \mathrm{dom}(f)$

Classes of functions

Total function: $dom(f) = S$ $\quad S \rightarrow T$
Partial function: $dom(f) \subset S$ $\quad S ⇸ T$

Injection: if $f(x) = f(y)$, then $x = y$

Total injection $\quad S \rightarrowtail T$
Partial injection $\quad S ⤔ T$

Surjection: $ran(f) = T$

Total surjection $\quad S \twoheadrightarrow T$
Partial surjection $\quad S ⤀ T$

Bijection $\quad S ⤖ T$

Selecting the right type of function imposes (useful) constraints / invariants to the domain and make it possible to use different proofs.

Arithmetic

- The usual (+, -, *, $\div$) plus: mod, ˆ (power).
- card(set), min(set), max(set)

A Phone Agenda

- Model a phone agenda.
- Associates phone numbers and people.
- We do not care what phone numbers and people are.
  - E.g., phone numbers do **not** have to be numbers.
  - We don't make arithmetic with them!
- Plus a set of integrity constraints, operations.

A Phone Agenda

Requirements

FUN 1	We should model a library to handle people and their phone numbers, providing a series of operations.

FUN 2	The library should allow us to add a person and their phone number.

FUN 3	The library should allow us to remove a phone number from the agenda.

FUN 4	The library should allow us to remove a person from the agenda.

FUN 5	The library should allow us to mark a phone number as the preferred contact for the person to whom the phone number belongs

A Phone Agenda

Requirements

FUN 6	The library should allow us to **unmark** a phone number as preferred contact
FUN 7	The library should allow us to transfer one phone number to a new owner
FUN 8	There cannot be persons in the agenda without an associated phone number
FUN 9	There cannot be phone numbers in the agenda without an associated owner

A Phone Agenda

Requirements

FUN 10	One person can have several phone numbers

FUN 11	Every phone number can be the contact of one person only

FUN 12	Any person must have at most one preferred number

- We don't include events to consult the agenda (e.g., "Give me person X's phone number"). They are trivial.
- The events will have guards as non-restrictive as possible as long as a sensible outcome can be achieved.
  - E.g., removing a phone does not need to check that it is in the agenda; if it is not, it becomes a no-op.

Phone Agenda

CONTEXT phone_Ctx

SETS

People Infinite. If finite, all the POs can be discharged anyway

Phones Infinite. If finite, all the POs can be discharged anyway

END

VARIABLES

agenda The agenda where we store names and phone numbers

preferred A set of numbers that we prefer for calling some people

INVARIANTS

$\texttt{invAgenda}$: $agenda \in Phones \rightarrowtail\!\!\!\!\!\!+ \; People$

- Every phone belongs to one person only
- There are no phones without an owner
- There are no persons in the agenda without a phone

$\texttt{invPref}$: $preferred \subseteq dom(agenda)$

The preferred numbers have to be in the agenda

$\texttt{uniquePref}$: $\forall p1, p2 \cdot (p1 \in preferred \land p2 \in preferred \land p1 \neq p2) \Rightarrow agenda(p1) \neq agenda(p2)$

Every person has at most one preferred contact

Initialisation

We start with an empty agenda

begin

$\texttt{act1}$: $agenda := \varnothing$

$\texttt{act3}$: $preferred := \varnothing$

end

Phone Agenda

Event AddPhone ⟨ordinary⟩ $\widehat{=}$
Insert a new phone and person to who it is associated.
any
person External parameters
phone
where
grd1: $person \in People$
grd2: $phone \in Phones$
grd3: $phone \notin dom(agenda)$
Needed to respect uniquePref
then
act1: $agenda(phone) := person$
end

If we do not have grd3, uniquePref/INV cannot be discharged because of the following scenario: let us have

agenda = $\{ph1 \mapsto prs1, ph2 \mapsto prs2\}$
preferred = $\{ph1, ph2\}$

If AddPhone is invoked with parameters $prs2, ph1$, the result would be

agenda = $\{ph1 \mapsto prs2, ph2 \mapsto prs2\}$
preferred = $\{ph1, ph2\}$

which violates the invariant.

Phone Agenda

Event RemovePhone $\langle$ordinary$\rangle$ $\widehat{=}$ Remove a phone number. If it's the last one for a person, then the owner also needs to be removed.

any

phone

where

grd1: $phone \in Phones$ We do not need to require that it is already in the agenda (but we might). Nothing will happen if it is not. If it is the last phone of a person, the person has to be removed as well.

then

act1: $agenda := \{phone\} \lhd agenda$

"agenda := agenda \ {phone ↦ agenda(phone)}" also works

act22: $preferred := preferred \setminus \{phone\}$ We cannot have orphan phone numbers. invPref would be violated otherwise.

end

Phone Agenda

Event RemovePerson ⟨ordinary⟩ $\widehat{=}$ If person not in agenda, nothing changes

any

person

where

$\texttt{grd1}$: $person \in People$

then

$\texttt{act1}$: $agenda := agenda \rhd \{person\}$ Remove from agenda all entries associated with that person

$\texttt{act2}$: $preferred := preferred \setminus agenda^{-1}[\{person\}]$ If the person had a preferred phone number, we have to remove it. Get the phone numbers associated with the person, remove them from the list of preferred phone numbers.

end

$dom(agenda \rhd \{person\})$ instead of $agenda^{-1}[\{person\}]$ would also work – they are equivalent expressions.

Phone Agenda

Event MakePreferred ⟨ordinary⟩ $\widehat{=}$ Remember at most one preferred phone number per person!

any

phone

where

grd2: $phone \in dom(agenda)$ We cannot make a phone preferred if it is not in the agenda

then

act1: $preferred :=$
$(preferred \setminus agenda^{-1}[\{agenda(phone)\}]) \cup \{phone\}$ If we just do preferred := preferred $\cup$ {phone} we might end up with more than one preferred phone # per person!

end

If we have

agenda = $\{ph1 \mapsto p1, ph2 \mapsto p1\}$
preferred = $\{ph1\}$

and we want to mark $ph2$ as preferred, we have to remove $ph1$ it from the set of preferred phones. We ensure that by removing first from *preferred* all the phones that belong to the owner of $ph2$.

Phone Agenda

Event RemovePreferred ⟨ordinary⟩ $\widehat{=}$

any

phone

where

grd1: $phone \in Phones$ It could also be "phone ∈ dom(agenda)". If it is not in the agenda, nothing will happen.

then

act1: $preferred := preferred \setminus \{phone\}$

end

Phone Agenda

Event TrasferPhone $\langle$ordinary$\rangle$ $\widehat{=}$ Change the owner of a phone #. We do not set it as preferred.

any

phone
next_owner

where

grd1: $phone \in dom(agenda)$
grd2: $next\_owner \in People$
grd3: $next\_owner \neq agenda(phone)$ It does not make sense to transfer a phone to its current owner. We could accept it, but it complicates the specification. For the sake of clarity, it seems simpler just not to allow that transition.

then

act1: $agenda(phone) := next\_owner$
act2: $preferred := preferred \setminus \{phone\}$

end

Extra slides / example

An example of functions and relations: an old society

- Every person is either a man or a woman.
- No person is man and woman at the same time.
- Only women have husbands, who must be men.
- Woman have at most one husband.
- Men have at most one wife.
- Mother are married women.

An example of functions and relations: an old society

Every person is man or woman $\qquad$ $men \subseteq PERSON$

An example of functions and relations: an old society

Every person is man or woman
No person is man and woman

$$men \subseteq PERSON$$
$$women = PERSON \setminus men$$

An example of functions and relations: an old society

Every person is man or woman
No person is man and woman
Women have husbands (men)
At most one husband per woman
Men at most one wife

$$men \subseteq PERSON$$
$$women = PERSON \setminus men$$

$$husband \in women \rightarrowtail\!\!\!\!\!\!\shortrightarrow men$$

An example of functions and relations: an old society

Every person is man or woman	$men \subseteq PERSON$
No person is man and woman	$women = PERSON \setminus men$
Women have husbands (men)	
At most one husband per woman	$husband \in women \rightarrowtail\!\!\!\!\!\!\!\!+\ men$
Men at most one wife	
Mother are married women	$mother \in PERSON \,+\!\!\!\!\rightarrow \mathrm{dom}(husband)$

An example of functions and relations: an old society

Every person is man or woman	$men \subseteq PERSON$
No person is man and woman	$women = PERSON \setminus men$
Women have husbands (men)	
At most one husband per woman	$husband \in women \rightarrowtail\!\!\!\!\rightarrow men$
Men at most one wife	
Mother are married women	$mother \in PERSON \nrightarrow \mathrm{dom}(husband)$

Let us derive some relations (Double check with Rodin)

$wife =$

$spouse =$

$father =$

$children =$

$daughter =$

$sibling =$

$brother =$

An example of functions and relations: an old society

Every person is man or woman	$men \subseteq PERSON$
No person is man and woman	$women = PERSON \setminus men$
Women have husbands (men)	
At most one husband per woman	$husband \in women \rightarrowtail\!\!\!\!\!+ men$
Men at most one wife	
Mother are married women	$mother \in PERSON \nrightarrow \mathrm{dom}(husband)$

Let us derive some relations (Double check with Rodin)

$wife = husband^{-1}$
$spouse =$
$father =$
$children =$

$daughter =$
$sibling =$
$brother =$

An example of functions and relations: an old society

Every person is man or woman	$men \subseteq PERSON$
No person is man and woman	$women = PERSON \setminus men$
Women have husbands (men)	
At most one husband per woman	$husband \in women \rightarrowtail\mapsto men$
Men at most one wife	
Mother are married women	$mother \in PERSON \pluto \mathrm{dom}(husband)$

Let us derive some relations (Double check with Rodin)

$wife = husband^{-1}$
$spouse = husband \cup wife$
$father =$
$children =$

$daughter =$
$sibling =$
$brother =$

An example of functions and relations: an old society

Every person is man or woman	$men \subseteq PERSON$
No person is man and woman	$women = PERSON \setminus men$
Women have husbands (men)	
At most one husband per woman	$husband \in women \rightarrowtail\!\!\!\!\!\!\rightarrow men$
Men at most one wife	
Mother are married women	$mother \in PERSON \nrightarrow \mathrm{dom}(husband)$

Let us derive some relations (Double check with Rodin)

$wife = husband^{-1}$

$spouse = husband \cup wife$

$father = mother ; husband$

$children =$

$daughter =$

$sibling =$

$brother =$

An example of functions and relations: an old society

Every person is man or woman	$men \subseteq PERSON$
No person is man and woman	$women = PERSON \setminus men$
Women have husbands (men)	
At most one husband per woman	$husband \in women \rightarrowtail\!\!\!\!\!\rightarrow men$
Men at most one wife	
Mother are married women	$mother \in PERSON \nrightarrow \mathrm{dom}(husband)$

Let us derive some relations (Double check with Rodin)

$wife = husband^{-1}$
$spouse = husband \cup wife$
$father = mother ; husband$
$children = (mother \cup father)^{-1}$

$daughter =$
$sibling =$
$brother =$

An example of functions and relations: an old society

Every person is man or woman	$men \subseteq PERSON$
No person is man and woman	$women = PERSON \setminus men$
Women have husbands (men)	
At most one husband per woman	$husband \in women \rightarrowtail\mathrel{\mkern-14mu}\shortmid\; men$
Men at most one wife	
Mother are married women	$mother \in PERSON \nrightarrow \mathrm{dom}(husband)$

Let us derive some relations (Double check with Rodin)

$wife = husband^{-1}$

$spouse = husband \cup wife$

$father = mother ; husband$

$children = (mother \cup father)^{-1}$

$daughter = children \rhd women$

$sibling =$

$brother =$

An example of functions and relations: an old society

Every person is man or woman	$men \subseteq PERSON$
No person is man and woman	$women = PERSON \setminus men$
Women have husbands (men)	
At most one husband per woman	$husband \in women \rightarrowtail\!\!\!\!\!\!\!\!\;+\; men$
Men at most one wife	
Mother are married women	$mother \in PERSON \mathrel{+\!\!\!\rightarrow} \mathrm{dom}(husband)$

Let us derive some relations (Double check with Rodin)

$wife = husband^{-1}$

$spouse = husband \cup wife$

$father = mother ; husband$

$children = (mother \cup father)^{-1}$

$daughter = children \rhd women$

$sibling = (children^{-1} ; children) \setminus \mathrm{id}(PERSON)$

$brother =$

An example of functions and relations: an old society

Every person is man or woman	$men \subseteq PERSON$
No person is man and woman	$women = PERSON \setminus men$
Women have husbands (men)	
At most one husband per woman	$husband \in women \mathrel{\rightarrowtail\!\!\!\!\!\!+} men$
Men at most one wife	
Mother are married women	$mother \in PERSON \mathrel{\rightarrow\!\!\!\!\!\!+} \mathrm{dom}(husband)$

Let us derive some relations (Double check with Rodin)

$wife = husband^{-1}$
$spouse = husband \cup wife$
$father = mother ; husband$
$children = (mother \cup father)^{-1}$

$daughter = children \rhd women$
$sibling = (children^{-1} ; children) \setminus \mathrm{id}(PERSON)$
$brother = men \rhd sibling$

Properties

$$
\begin{aligned}
mother &= father; wife \\
spouse &= spouse^{-1} \\
father; father^{-1} &= mother; mother^{-1} \\
father; mother^{-1} &= \varnothing \\
mother; father^{-1} &= \varnothing \\
father; children &= mother; children \\
sibling &= sibling^{-1} \\
cousin &= cousin^{-1}
\end{aligned}
$$