

More on Event-B: Relations

© Michael Butler

University of Southampton

November 2, 2010

Ordered Pairs and Cartesian Products

An **ordered pair** is an element consisting of two parts:
a **first** part and a **second** part.

An ordered pair with first part x and second part y is
written: $x \mapsto y$

The **Cartesian product** of two sets is the **set of pairs** whose first
part is in S and second part is in T .

The Cartesian product of S with T is written: $S \times T$

Cartesian Products: Definition and Examples

Defining Cartesian product:

| Predicate | Definition |
|------------------------------|--------------------------|
| $x \mapsto y \in S \times T$ | $x \in S \wedge y \in T$ |

Examples:

$$\{a, b, c\} \times \{1, 2\} = \{ a \mapsto 1, a \mapsto 2, b \mapsto 1, \\ b \mapsto 2, c \mapsto 1, c \mapsto 2, \}$$

$$\{a, b, c\} \times \{\} = ?$$

$$\{ \{a\}, \{a, b\} \} \times \{1, 2\} = ?$$

Cartesian Product is a Type Constructor

$S \times T$ is a new type constructed from types S and T .

Cartesian product is the type constructor for ordered pairs.

Given $x \in S$, $y \in T$, we have

$$\boxed{x \mapsto y \in S \times T}$$

$$4 \mapsto 7 \in ?$$

$$\{5, 6, 3\} \mapsto 4 \in ?$$

$$\{4 \mapsto 8, 3 \mapsto 0, 2 \mapsto 9\} \in ?$$

Sets of Order Pairs

A database can be modelled as a **set of ordered pairs**:

$$\begin{aligned} \text{directory} = \{ & \text{mary} \mapsto 287573, \\ & \text{mary} \mapsto 398620, \\ & \text{john} \mapsto 829483, \\ & \text{jim} \mapsto 398620 \} \end{aligned}$$

directory has type

$$\text{directory} \in \mathbb{P}(\text{Person} \times \text{PhoneNum})$$

Relations

A **relation** is a set of ordered pairs.

A relation is a common modelling structure so Event-B has a special notation for it:

$$\boxed{T \leftrightarrow S} = \mathbb{P}(T \times S)$$

So we can write:

$$directory \in Person \leftrightarrow PhoneNum$$

Do not confuse the arrow symbols:

\leftrightarrow combines **two sets** to form a **set**.

\mapsto combines **two elements** to form an **ordered pair**.

Domain and Range

- ▶ The **domain** of a relation R is the set of first parts of all the pairs in R , written $\boxed{dom(R)}$
- ▶ The **range** of a relation R is the set of second parts of all the pairs in R , written $\boxed{ran(R)}$

| Predicate | Definition |
|----------------|-------------------------------------|
| $x \in dom(R)$ | $\exists y \cdot x \mapsto y \in R$ |
| $y \in ran(R)$ | $\exists x \cdot x \mapsto y \in R$ |

Examples:

$$dom(directory) = \{mary, john, jim\}$$

$$ran(directory) = \{287573, 398620, 829483\}$$

Telephone Directory Model

- ▶ Phone directory relates people to their phone numbers.
- ▶ Each person can have zero or more numbers.
- ▶ People can share numbers.

context *PhoneContext*

sets *Person PhoneNum*

end

machine *PhoneBook*

variables *dir*

invariants $dir \in Person \leftrightarrow PhoneNum$

initialisation $dir := \{\}$

Extending the Directory

Add an entry to the directory:

$$\begin{array}{l} \textit{AddEntry} \hat{=} \textbf{any } p, n \textbf{ where} \\ \quad p \in \textit{Person} \\ \quad n \in \textit{PhoneNum} \\ \textbf{then} \\ \quad \textit{dir} := \textit{dir} \cup \{p \mapsto n\} \\ \textbf{end} \end{array}$$

Relational Image

Assume $R \in S \leftrightarrow T$ and $A \subseteq S$

The **relational image** of set A under relation R is written $R[A]$

| Predicate | Definition |
|--------------|--|
| $y \in R[A]$ | $\exists x \cdot x \in A \wedge x \mapsto y \in R$ |

Example:

$$\begin{aligned} \text{directory} = \{ & \text{mary} \mapsto 287573, \\ & \text{mary} \mapsto 398620, \\ & \text{john} \mapsto 829483, \\ & \text{jim} \mapsto 398620 \} \end{aligned}$$

$$\text{directory}[\{\text{mary}\}] = \{287573, 398620\}$$

Modelling Queries using Relational Image

Determine all the numbers associated with a person in the directory:

$$\begin{aligned} \text{GetNumbers} \hat{=} & \text{ any } p, ns! \text{ where} \\ & p \in \text{Person} \\ & ns! = \text{dir}[\{p\}] \\ & \text{end} \end{aligned}$$

Determine all the numbers associated with a set of people:

$$\begin{aligned} \text{GetMultiNumbers} \hat{=} & \text{ any } ps, ns! \text{ where} \\ & ps \subseteq \text{Person} \\ & ns! = \text{dir}[ps] \\ & \text{end} \end{aligned}$$

Relational Inverse

Given $R \in S \leftrightarrow T$, the **relational inverse** of R is written R^{-1}

| Predicate | Definition |
|--------------------------|---------------------|
| $y \mapsto x \in R^{-1}$ | $x \mapsto y \in R$ |

Example:

$$\begin{aligned} directory^{-1} = \{ & 287573 \mapsto mary, \\ & 398620 \mapsto mary, \\ & 829483 \mapsto john, \\ & 398620 \mapsto jim \} \end{aligned}$$

$$directory^{-1}[\{398620\}] = \{ mary, jim \}$$

Inverse Queries

Return all the people associated with a number in the directory:

$$\begin{aligned} \textit{GetNames} \hat{=} & \textbf{any } n, ps! \textbf{ where} \\ & n \in \textit{PhoneNum} \\ & ps! = \textit{dir}^{-1}[\{n\}] \\ & \textbf{end} \end{aligned}$$

Return all the people associated with a set of numbers:

$$\begin{aligned} \textit{GetMultiNames} \hat{=} & \textbf{any } ns, ps! \textbf{ where} \\ & ns \subseteq \textit{PhoneNum} \\ & ps! = \textit{dir}^{-1}[ns] \\ & \textbf{end} \end{aligned}$$

Domain Restriction

Given $R \in S \leftrightarrow T$ and $A \subseteq S$,
the **domain restriction** of R by A is written $\boxed{A \triangleleft R}$

Restrict relation R so that it only contains pairs whose first part is in the set A .

Example:

$$\text{directory} = \{ \text{mary} \mapsto 287573, \text{mary} \mapsto 398620, \\ \text{john} \mapsto 829483, \text{jim} \mapsto 398620 \}$$

$$\{\text{john}, \text{jim}, \text{jane}\} \triangleleft \text{directory} = \{ \text{john} \mapsto 829483, \\ \text{jim} \mapsto 398620 \}$$

Domain Subtraction

Given $R \in S \leftrightarrow T$ and $A \subseteq S$,
the **domain subtraction** of R by A is written $A \triangleleft R$

Remove those pairs from R whose first part is in A .

Example:

$$\{john, jim, jane\} \triangleleft directory = \{ mary \mapsto 287573, \\ mary \mapsto 398620 \}$$

Removing Entries from the Directory

Remove all the entries associated with a person in the directory:

RemovePerson $\hat{=}$ **any** *p* **where**
 p \in *Person*
 then
 dir $:=$ {*p*} \Leftarrow *dir*
 end

Remove all the entries associated with a number in the directory:

RemoveNumber $\hat{=}$ **any** *n* **where**
 n \in *PhoneNum*
 then
 dir $:=$ *dir* \rhd {*n*}
 end

Relational Composition

Given $Q \in S \leftrightarrow T$ and $R \in T \leftrightarrow U$,
the **relational composition** of Q and R is written $\boxed{Q ; R}$

We have that $Q ; R \in S \leftrightarrow U$

| Predicate | Definition |
|---------------------------|--|
| $x \mapsto z \in (Q ; R)$ | $\exists y \cdot x \mapsto y \in Q \wedge y \mapsto z \in R$ |

Example:

$$M = \{ a \mapsto l, b \mapsto m, c \mapsto n \}$$

$$N = \{ l \mapsto 4, n \mapsto 6, p \mapsto 8 \}$$

$$M ; N = ?$$

Composition and Image

Given $Q \in S \leftrightarrow T$ and $R \in T \leftrightarrow U$ and $A \subseteq S$

$$(Q ; R)[A] = R[Q[A]]$$

Example:

$$M = \{ a \mapsto l, b \mapsto m, c \mapsto n \}$$

$$N = \{ l \mapsto 4, n \mapsto 6, p \mapsto 8 \}$$

$$(M ; N) [\{a, b\}] = ?$$

$$N[M[\{a, b\}]] = ?$$

Extend directory with friends

variables *dir, friend*

invariants

friend $\in Person \leftrightarrow Person$

dir $\in Person \leftrightarrow PhoneNum$

Return the telephone numbers of all friends of *p*:

GetFriendNumbers $\hat{=}$

any *p, ns!* **where**

p $\in Person$

ns! $= (friend; dir)[\{p\}]$

end

Recap

- ▶ Cartesian product is the type constructor for pairs of elements.
- ▶ A relation is a set of pairs.
- ▶ Range of a relation, domain of a relation.
- ▶ Relational image, relational inverse.
- ▶ Restriction and subtraction.
- ▶ Relational composition.