Relations and Functions

Michael Butler

4 March 2019

Sets, relations, functions

- Powerset is the type constructor for sets of elements
- Cartesian product is the type constructor for pairs of elements
- A relation is a set of pairs
  - Domain and range of a relation
  - Relational image
  - Restriction and subtraction
- A function is a special case of a relation
  - Many-to-one: each domain element mapped to a unique range element
  - Partial function, function application
  - Function override
  - Total functions

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:

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

Relational Image

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

Relational image examples:

$$
\begin{aligned}
directory[\; \{mary\} \;] \quad &= \quad \{\; 287573, \;\; 398620 \;\} \\
directory[\; \{\; john, jim \;\} \;] \quad &= \quad \{\; 829483, \;\; 398620 \;\}
\end{aligned}
$$

Relational Image Definition

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

The relational image of set A under relation R is written $\boxed{R[A]}$

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

Modelling Queries using Relational Image

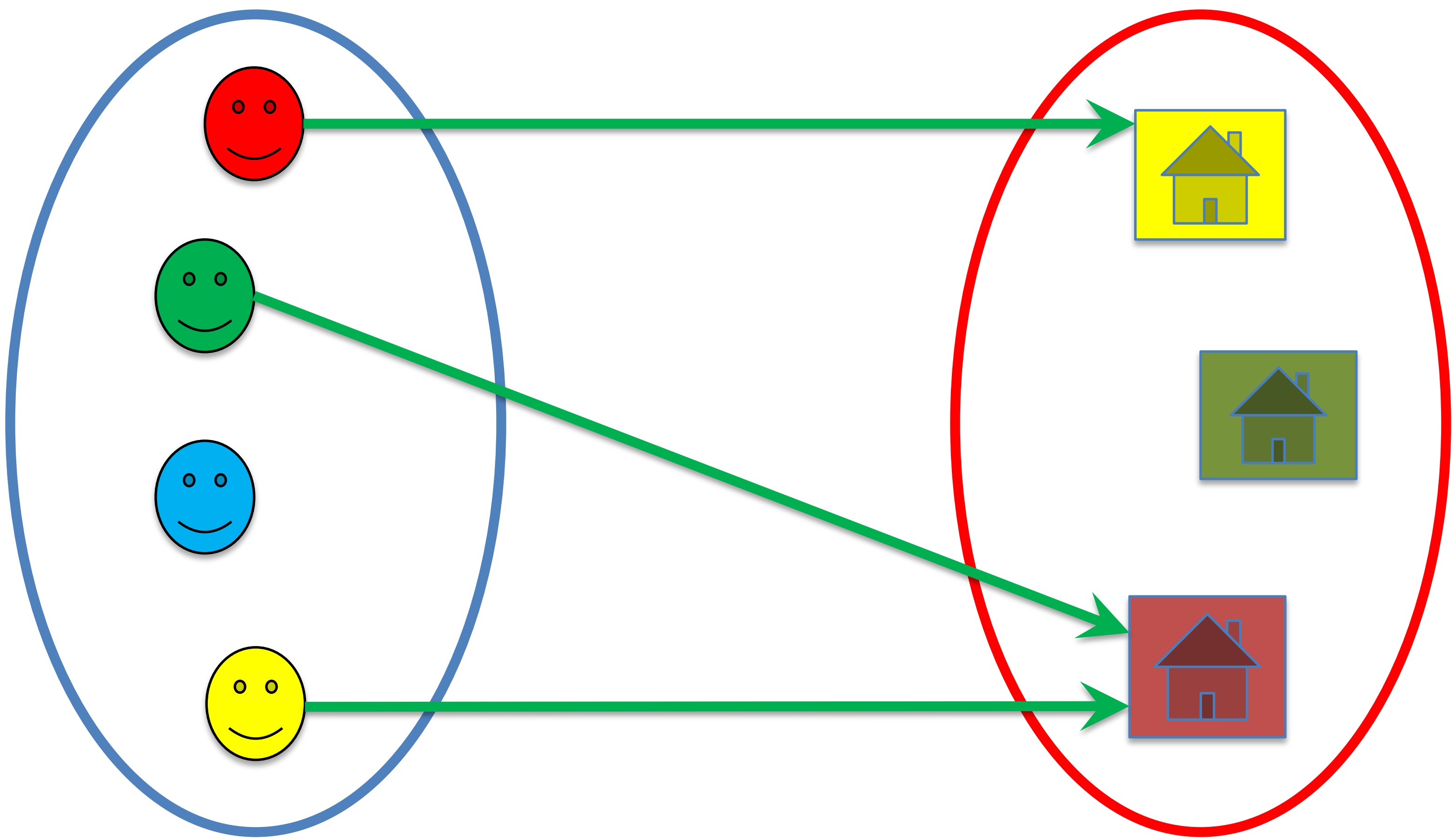
Determine all the numbers associated with a person in the directory:

$$
\begin{array}{lcl}
\mathit{GetNumbers} & \triangleq & \textbf{any } p, \mathit{result} \textbf{ where} \\
& & \quad p \in \mathit{Person} \\
& & \quad \mathit{result} = \mathit{dir}[\ \{p\}\ ] \\
& & \textbf{end}
\end{array}
$$

Determine all the numbers associated with a set of people:

$$
\begin{array}{lcl}
\mathit{GetMultiNumbers} & \triangleq & \textbf{any } ps, \mathit{result} \textbf{ where} \\
& & \quad ps \subseteq \mathit{Person} \\
& & \quad \mathit{result} = \mathit{dir}[\ ps\ ] \\
& & \textbf{end}
\end{array}
$$

Location



Many-to-one relation

Partial Functions

Special kind of relation: each domain element has at most one range element associated with it.

To declare f as a partial function:

$$\boxed{f \ \in \ X \mathrel{+\!\!\!\rightarrow} Y}$$

This says that f is a many-to-one relation

Each domain element is mapped to exactly one range element:

$$x \mapsto y \ \in \ f \quad \wedge \quad y' \neq y \qquad \Longrightarrow \qquad x \mapsto y' \ \notin \ f$$

If x is mapped to y, then x cannot be mapped to another value y'.

Function Application

We can use function application for partial functions.

If $x \in \mathrm{dom}(f)$, then we write $\boxed{f(x)}$ for the unique range element associated with x in f.

If $x \notin \mathrm{dom}(f)$, then $f(x)$ is undefined.

Examples

$$
\begin{aligned}
dir1 \;=\; \{\; & mary \mapsto 398620, \\
& jim \mapsto 493028, \\
& jane \mapsto 493028 \;\}
\end{aligned}
\qquad
\begin{aligned}
dir2 \;=\; \{\; & mary \mapsto 287573, \\
& mary \mapsto 398620, \\
& jane \mapsto 493028 \;\}
\end{aligned}
$$

$$
\begin{aligned}
dir1 &\in Person \mathrel{+\!\!\!\rightarrow} Phone \\
dir1(jim) &= 493028 \\
dir1(sarah) &\quad \text{is undefined}
\end{aligned}
$$

$$
dir2 \notin Person \mathrel{+\!\!\!\rightarrow} Phone
$$

Well-definedness and application definitions

Expression	Well-definedness condition
$f(x)$	$x \in dom(f) \ \wedge \ f \in X \nrightarrow Y$

The following definition of function application assumes that $f(x)$ is well-defined:

Predicate	Definition
$y = f(x)$	$x \mapsto y \in f$

Birthday Book Example

Birthday book relates people to their birthday.

Each person has one birthday.

People can share birthdays.

sets $\quad PERSON \quad DATE$

variables $\quad birthday$
invariants $\quad birthday \ \in \ PERSON \rightarrow\!\!\!\!\!+ \, DATE$

initialisation $\quad birthday \ := \ \{\}$

Adding and checking birthdays

Add an entry to the directory:

$$
\begin{array}{lcl}
AddEntry & \hat{=} & \textbf{any}\ p, d\ \textbf{where} \\
 & & \quad p \in Person \\
 & & \quad p \notin dom(birthday) \\
 & & \quad d \in Date \\
 & & \textbf{then} \\
 & & \quad birthday \ := \ birthday \cup \{p \mapsto d\} \\
 & & \textbf{end}
\end{array}
$$

Check a person's birthday:

$$
\begin{array}{lcl}
Check & \hat{=} & \textbf{any}\ p, result\ \textbf{where} \\
 & & \quad p \in dom(birthday) \\
 & & \quad result = birthday(p) \\
 & & \textbf{end}
\end{array}
$$

Domain Restriction

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

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

Example:

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

$$\begin{aligned} \{\mathit{john}, \mathit{jim}, \mathit{jane}\} \lhd \mathit{directory} \quad = \quad \{ \ & \mathit{john} \mapsto 829483, \\ & \mathit{jim} \mapsto 398620 \ \} \end{aligned}$$

Domain Subtraction

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

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

Example:

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

$$\begin{aligned} \{\mathit{john}, \mathit{jim}, \mathit{jane}\} \trianglelefteq \mathit{directory} \quad = \quad \{ \; & \mathit{mary} \mapsto 287573, \\ & \mathit{mary} \mapsto 398620 \; \} \end{aligned}$$

Domain and Range, Restriction and Substraction

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

Predicate	Definition	
$x \mapsto y \in A \lhd R$	$x \mapsto y \in R \wedge x \in A$	domain restriction
$x \mapsto y \in A ⩤ R$	$x \mapsto y \in R \wedge x \notin A$	domain subtraction
$x \mapsto y \in R \rhd B$	$x \mapsto y \in R \wedge y \in B$	range restriction
$x \mapsto y \in R ⩥ B$	$x \mapsto y \in R \wedge y \notin B$	range subtraction

Removing Entries from the Directory

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

$$
\begin{array}{lcl}
RemovePerson & \hat{=} & \textbf{any } p \textbf{ where} \\
& & \quad p \in Person \\
& & \textbf{then} \\
& & \quad dir \ := \ \{p\} \lhd dir \\
& & \textbf{end}
\end{array}
$$

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

$$
\begin{array}{lcl}
RemoveNumber & \hat{=} & \textbf{any } n \textbf{ where} \\
& & \quad n \in PhoneNum \\
& & \textbf{then} \\
& & \quad dir \ := \ dir \rhd \{n\} \\
& & \textbf{end}
\end{array}
$$

Function Overriding

Override f by g $\boxed{f \oplus g}$

f and g must be partial functions of the same type

Override: replace existing mappings with new ones

Examples:

$$dir1 \;=\; \{\ mary \mapsto 398620,\ jim \mapsto 493028,\ jane \mapsto 493028\ \}$$

$$\begin{aligned} & dir1 \oplus \{\ mary \mapsto 674321\} \\ & \quad = \{\ mary \mapsto 674321,\ jim \mapsto 493028,\ jane \mapsto 493028\ \} \end{aligned}$$

$$\begin{aligned} & dir1 \oplus \{\ mary \mapsto 674321,\ jane \mapsto 829483\} \\ & \quad = \{\ mary \mapsto 674321,\ jim \mapsto 493028,\ jane \mapsto 829483\ \} \end{aligned}$$

Function Overriding Definition

Definition in terms of function override and set union:

$$
\begin{aligned}
f \mathbin{\lhd\!\!\!+} \{a \mapsto b\} &= (\{a\} \mathbin{⩤} f) \cup \{a \mapsto b\} \\
f \mathbin{\lhd\!\!\!+} g &= (\mathrm{dom}(g) \mathbin{⩤} f) \cup g
\end{aligned}
$$

Modifying a birthday

Modify an entry in the directory:

$$
\begin{array}{lcl}
\mathit{ModifyEntry} & \hat{=} & \textbf{any } p, d \textbf{ where} \\
& & \quad p \in \mathit{dom}(\mathit{birthday}) \\
& & \quad d \in \mathit{Date} \\
& & \textbf{then} \\
& & \quad \mathit{birthday} \ := \ \mathit{birthday} \lhd\!\!\!- \{p \mapsto d\} \\
& & \textbf{end}
\end{array}
$$

Syntactic shorthand:

$$
\begin{array}{lcl}
\mathit{ModifyEntry} & \hat{=} & \textbf{any } p, d \textbf{ where} \\
& & \quad p \in \mathit{Person} \\
& & \quad d \in \mathit{Date} \\
& & \textbf{then} \\
& & \quad \mathit{birthday}(p) \ := \ d \\
& & \textbf{end}
\end{array}
$$

Event-B Lecture Notes

- For overview of modelling with sets in Event-B see Notes:
- http://eprints.soton.ac.uk/402239/
- (also linked from COMP1216 web page)

- Read Sections 1-7