Formal Specification F28FS2, Lecture 7
Relations

Jamie Gabbay

February 17, 2014

Ordered pairs

An ordered pair $a \mapsto b : A \times B$ is a pair of $a : A$ and $b : B$, in order.

For example, $Jack \mapsto Jill : PERSON \times PERSON$.

For comparison, $\{a, a'\} : \mathbb{P}A$ is an unordered pair.

If A and B are types, then $A \times B$ is the type of ordered pairs of terms of type A and terms of type B.

Relations

A relation R is an element of the powerset of a product type.

In symbols:

$$R : \mathbb{P}(A \times B)$$

or more briefly (just a shorthand)

$$R : A \leftrightarrow B.$$

For example $\{Jack \mapsto Jill\} : PERSON \leftrightarrow PERSON$.

Also $\{Jack \mapsto Jill, Jill \mapsto Jack\} : PERSON \leftrightarrow PERSON$.

Which one is suitable for modelling ‘loves’?

Homogeneity

A relation R of type $A \leftrightarrow A$ is homogeneous.

$\{Jack \mapsto Jill, Jill \mapsto Jack\} : PERSON \leftrightarrow PERSON$ is homogeneous.

$\{2 \mapsto d\} : \mathbb{Z} \leftrightarrow Door$ is not homogeneous, where $[Door]$.

Example: doors

Declare free types $[PERSON, MODULE]$.

We can use $PERSON \leftrightarrow MODULE$ to describe who is taking what course.

Let $taking : PERSON \leftrightarrow MODULE$.

Then we say 'p is taking module m' when $p \mapsto m \in taking$.

Note that
$PERSON \times MODULE = \{p : PERSON, m : MODULE \bullet p \mapsto m\}$.
This is the total relation.

Example

Suppose a set $\mathit{firstYear} : \mathbb{P}\mathit{PERSON}$.

Here's an expression for 'the first years taking $\mathit{programming} : \mathit{MODULE}$':

$$\{x : \mathit{PERSON} \mid x \in \mathit{firstYear} \land x \mapsto \mathit{programming} \in \mathit{taking} \bullet x\}.$$

Here's an expression for 'the first years taking some module':

$$\{x : \mathit{PERSON} \mid x \in \mathit{firstYear} \land (\exists\, m : \mathit{MODULE} \bullet x \mapsto m \in \mathit{taking}) \bullet x\}.$$

Is this set equal to $\mathit{firstYear}$?

Difference between relations and binary predicates

A binary predicate $P(x, y)$ where $x : \mathbb{X}$ and $y : \mathbb{Y}$ and can be 'identified' with its graph

$$\{x : \mathbb{X}, y : \mathbb{Y} \mid P(x, y) \bullet x \mapsto y\} : \mathbb{X} \leftrightarrow \mathbb{Y}.$$

However, relations are sets so we can use the full vocabulary of Z to manipulate them.

That is, relations internalise predicates; we can manipulate predicates by manipulating relations.

Terminology that you need to know: Source, target, domain, range

The source of $R : A \leftrightarrow B$ is A.

The target of $R : A \leftrightarrow B$ is B.

The domain dom(R) of $R : A \leftrightarrow B$ is $\{a : A \mid (\exists\, b : B \bullet a \mapsto b \in R) \bullet a\}$.

The range range(R) of $R : A \leftrightarrow B$ is $\{b : B \mid (\exists\, a : A \bullet a \mapsto b \in R) \bullet b\}$.

$\text{dom}(\{\text{Jack} \mapsto \text{Jill}\}) = \{\text{Jack}\}$

$\text{range}(\{\text{Jack} \mapsto \text{Jill}\}) = \{\text{Jill}\}$

A schema to make somebody love somebody else

LovePotion

$loves, loves' : PERSON \leftrightarrow PERSON$

$person1?, person2? : PERSON$

$loves' = loves \cup \{person1? \mapsto person2?\}$

Relational image

If $R : A \leftrightarrow B$ and $S \subseteq A$ then $R(S)$ is those elements of B that are related to by some $a \in S$.

In symbols:

$$R(S) = \{b : B \mid (\exists\, a \in S \bullet a \mapsto b \in R)\}$$

For example $loves(\{Jack\}) = Jill$.

If $ego = \{p : PERSON \bullet p \mapsto p\}$ then $ego(S) = S$.

Fix some $saint : PERSON$. If $saintlylove = \{q : PERSON \bullet saint \mapsto q\}$ then $saintlylove(S) = PERSON$ if and only if $saint \in S$.

We may write $R(\{s\})$ as just $R(s)$.

Inverse of a relation

Just turn it round. $\{Jack \mapsto Jill\}^{-1} = \{Jill \mapsto Jack\}$.

Who can answer the question "write a schema to input a relation and return the inverse of that relation?".

Who can answer the question "write a schema to input a relation and return the symmetric closure of that relation, which is the union of the relation with its inverse?".

Inverse of a relation

Invert

$R : A \leftrightarrow B$

$R' : B \leftrightarrow A$

$R' = \{a : A, b : B \mid a \mapsto b \in R \bullet b \mapsto a\}$

SymmetricClosure

$R, R' : A \leftrightarrow A$

$R' = \{a : A, a' : A \mid a \mapsto a' \in R \bullet a' \mapsto a\} \cup R$

Domain and range restriction

$$S \lhd R = \{a \mapsto b : A \times B \mid a \mapsto b \in R \land a \in S\}.$$

(Notice the notation $a \mapsto b$ on the left. Can you expand it out?)

For example if

$$loves = \{Jack \mapsto Jill, Jill \mapsto Jack, Sally \mapsto Suzie, Tony \mapsto Tony\}$$

and

$$men = \{Jack, Tony\} \qquad women = \{Jill, Sally, Suzie\}$$

then $men \lhd loves = \{Jack \mapsto Jill, Tony \mapsto Tony\}$ and $women \lhd loves = \{Jill \mapsto Jack, Sally \mapsto Suzie\}$.

Range restriction, subtraction

There is also range restriction $R \rhd S$.

Exercise: what should that be?

There are domain and range anti-restriction

$$S \lhd R = \{a \mapsto b : A \times B \mid a \mapsto b \in R \land a \notin S\}$$
$$R \unrhd T = \{a \mapsto b : A \times B \mid a \mapsto b \in R \land b \notin T\}.$$

loves $\unrhd \{Tony\}$ is ‘everybody who does not love Tony’.

loves $\unrhd$ *men* is ‘everybody who loves only women’.

Composition

Aha! But do you love somebody who loves Tony?

Suppose $R : A \leftrightarrow B$ and $S : B \leftrightarrow C$. R composed with S is:

$R;\ S = \{a \mapsto c : A \times C \mid (\exists\, b : B \bullet (a \mapsto b \in R \land b \mapsto c \in S)) \bullet a \mapsto c\}$.

Take $R = \mathit{loves} = S$. Then *loves*; *loves* is the relation '*a* loves somebody (the *b* above) who loves *c*'.

Composition

More general if R is homogeneous, so $R : A \leftrightarrow A$, then

$$R^n = \overbrace{R;\ \ldots;\ R}^{n \text{ times}}.$$

In the case $n = 0$ take $R^0 = \{a : A \mid a \mapsto a\}$.

The transitive closure $R^+ = \bigcup\{n : \mathbb{N} \mid n > 0 \bullet R^n\}$.

The reflexive transitive closure $R^* = R^+ \cup R^0$.