CHAPTER 9

Relations

9.1 A relation is a set

The examples considered so far have been limited to one basic type which means that the specifications cannot develop to be any more sophisticated. What is needed is some way of relating sets to one another. This can be done by means of a *relation*, which is based on the idea of a *Cartesian product*.

9.2 Cartesian product

A *Cartesian product*, named after the French mathematician Descartes, is a pairing of values of two or more sets. The Cartesian product of the sets or types X, Y and Z would be written:

$$X \times Y \times Z$$

and pronounced 'the Cartesian product of X, Y and Z' or 'X cross Y cross Z'. Values drawn from this combination of sets are called *tuples* and are written:

$$(x, y, z)$$

where x is of type X, y of type Y and z of type Z.

For example:

$$\{red, green\} \times \{hardtop, softtop\} =$$
$$\{(red, hardtop), (red, softtop), (green, hardtop), (green, softtop)\}$$

Such a tuple is called *ordered* since the order of writing the components is important.

A tuple formed from two types is called a *pair* and a tuple formed from three types is called a *triple*. A tuple of n types is sometimes called an *n-tuple*.

An example of a tuple (a 4-tuple) connects information about a person:

$$NAME \times ADDRESS \times \mathbb{N} \times TELEPHONE$$

to record name, address, age and telephone-number.

9.3 Relations

A special case of a Cartesian product is a *pair*. A *binary* relation is a set of pairs, of related values.

For example, a relation called *speaks* between countries and languages spoken in those countries can be thought of as a set of pairs. Given:

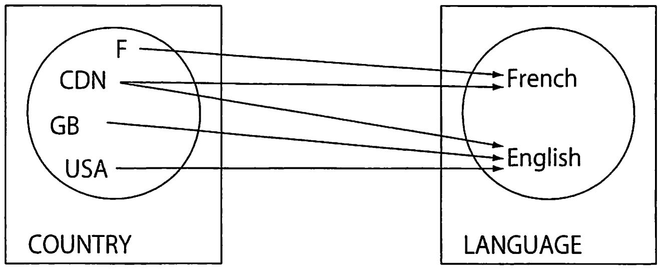
[COUNTRY]	the set of all countries
[LANGUAGE]	the set of all languages

part of the value of this set might be

{(France, French), (Canada, French),
(Canada, English), (GB, English), (USA, English)}

Figure 9.1

speaks



Such a set could be declared:

speaks: $\mathbb{P}$(COUNTRY × LANGUAGE)

Note that there are no restrictions requiring values of the two sets to be paired only one to one; relations pair *many* to *many*. For example, it would be quite acceptable to have the relation:

{(France, French), (Germany, German), (Austria, German),
(Switzerland, French), (Switzerland, German),
(Switzerland, Italian), (Switzerland, Romansch)}

Furthermore, there is no particular reason to choose to relate the values in this direction; one could just as well relate language to country, by declaring:

spoken: $\mathbb{P}$(LANGUAGE × COUNTRY)

9.4 Declaring a relation

The idea of a relation is reinforced by the use of the two-headed arrow in the alternative conventional style of declaration:

R: X ↔ Y
speaks: COUNTRY ↔ LANGUAGE

These can be pronounced: '*R* relates *X* to *Y*' and '*speaks* relates country to language'.

Note the following equivalences:

$X \leftrightarrow Y == \mathbb{P}(X \times Y)$

speaks: COUNTRY $\leftrightarrow$ LANGUAGE
==
speaks: $\mathbb{P}$(COUNTRY $\times$ LANGUAGE)

9.5 Maplets

The idea of a related pair is reinforced by the conventional notation for one pair in a relation, a *maplet*:

$x \mapsto y == (x, y)$

pronounced 'x is related to y' or 'x maps to y'

GB $\mapsto$ English $\in$ speaks
== (GB, English) $\in$ speaks

The set of pairs given above could alternatively be written as a set of maplets:

{France $\mapsto$ French, Germany $\mapsto$ German, Austria $\mapsto$ German,
Switzerland $\mapsto$ French, Switzerland $\mapsto$ German,
Switzerland $\mapsto$ Italian, Switzerland $\mapsto$ Romansch}

9.6 Membership

To discover if a certain pair of values are related it is sufficient to see if the pair or maplet is an element of the relation:

(GB, English) $\in$ speaks

or

GB $\mapsto$ English $\in$ speaks

9.7 Infix relations

If we declare a relation using low-line characters as place-holders to show the fact that it is *infix*:

speaks: COUNTRY $\leftrightarrow$ LANGUAGE

then we can use the name of the relation as an *infix* operator:

GB speaks English

In general:

$x\ R\ y == x \mapsto y \in R == (x, y) \in R$

9.8 Domain and range

A relation relates values of a set called the *source* or *from-set* to values of a set called the *target* or *to-set*. In the example:

$$R: X \leftrightarrow Y$$

$$R = \{(x_1, y_2), (x_2, y_2), (x_4, y_3), (x_6, y_3), (x_4, y_2)\}$$

the source is *X* and the target is *Y*. In the example:

speaks: COUNTRY $\leftrightarrow$ LANGUAGE

the source is *COUNTRY* and the target is *LANGUAGE*.

9.8.1 Domain

In most cases only a subset of the source is involved in the relation. This subset is called the *domain* and is written *dom*.

In the example the domain of *R* is

$$\text{dom } R = \{x_1, x_2, x_4, x_6\}$$

It is the subset containing those values of *X* which are related by *R* to values of *Y*.

9.8.2 Range

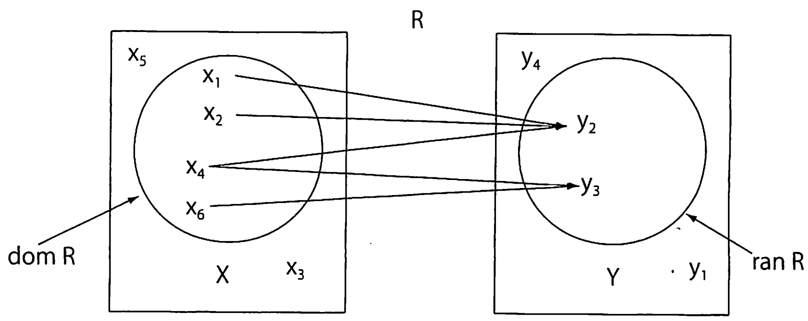
Usually only a subset of the target is involved in a relation. This subset is called the *range* and written *ran*.

In the example the range of a relation *R* is

$$\text{ran } R = \{y_2, y_3\}$$

It is the subset containing those values of *Y* to which *R* relates at least one value of *X*.

Figure 9.2



9.8.3 Examples

The domain of the relation *speaks*

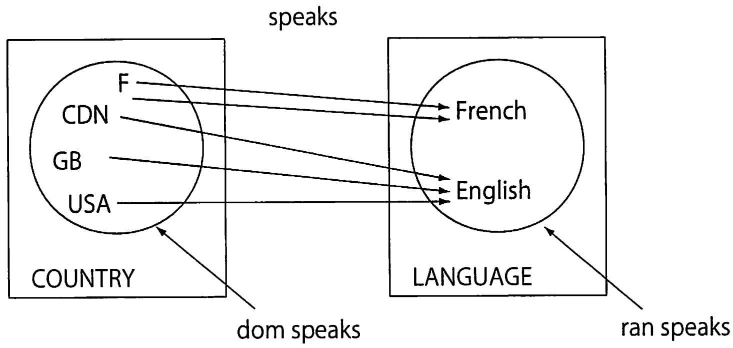
dom speaks

is the set of those countries where at least one language is spoken.
The range of *speaks*

ran speaks

is that set of languages which are spoken in at least one country.

Figure 9.3



9.9 Relational image

To discover the set of values from the range of a relation related to a set of values from the domain of the relation one can use the *relational image*:

$R ⦇S⦈$

pronounced 'the relational image of *S* in *R*'. For example, the languages spoken in France and Switzerland would be:

speaks ⦇{France, Switzerland}⦈

which is the set

{French, German, Italian, Romansch}

9.10 Constant value for a relation

Some relations have constant values. If the value of the relation is known, it can be given by an *axiomatic definition*. For example, to define the infix relation greater-than-or-equal-to (≥):

$$\_\geq\_ : \mathbb{Z} \leftrightarrow \mathbb{Z}$$

$$\forall i, j : \mathbb{Z} \bullet i \geq j \Leftrightarrow \exists n : \mathbb{N} \bullet i = j + n$$

Where convenient several relations can be combined in one definition:

$$
\begin{array}{|l}
\_\geq\_ : \mathbb{Z} \leftrightarrow \mathbb{Z} \\
\_>\_ : \mathbb{Z} \leftrightarrow \mathbb{Z} \\
\hline
\forall i, j : \mathbb{Z} \bullet \\
(i \geq j \Leftrightarrow \exists\, n : \mathbb{N} \bullet i = j + n \wedge \\
i > j \Leftrightarrow \exists\, n : \mathbb{N}_1 \bullet i = j + n)
\end{array}
$$

The low-line (_) characters are used here to indicate that the names (≥) and (>) are infix, that is, used between two values.

9.11 Example of a relation

Public holidays around the world can be described as follows:

[COUNTRY]	the set of all the countries of the world
[DATE]	the dates of a given year

The relationship between countries and the dates of the countries' public holidays is the relation *holidays*:

$$
\begin{array}{|l}
\textsf{Hols} \\
\hline
\text{holidays} : \text{COUNTRY} \leftrightarrow \text{DATE}
\end{array}
$$

An operation to discover whether a date *d?* is a public holiday in country *c?* is:

REPLY ::= yes | no

$$
\begin{array}{|ll}
\multicolumn{2}{l}{\textsf{Enquire}} \\
\Xi\text{Hols} & \\
c? : & \text{COUNTRY} \\
d? : & \text{DATE} \\
rep! : & \text{REPLY} \\
\hline
\multicolumn{2}{|l}{(c? \mapsto d? \in \text{holidays} \wedge rep! = yes)} \\
\multicolumn{2}{|l}{\vee} \\
\multicolumn{2}{|l}{(c? \mapsto d? \notin \text{holidays} \wedge rep! = no)}
\end{array}
$$

An operation to decree a public holiday in country *c?* on date *d?* is:

$$\begin{array}{|ll}
\multicolumn{2}{l}{\text{Decree}} \\
\Delta Hols & \\
c?: & COUNTRY \\
d?: & DATE \\
\hline
holidays' = holidays \cup \{c? \mapsto d?\} & \\
\hline
\end{array}$$

Note that *Decree* ignores the possibility of the date already being a public holiday in that country.

An operation to abolish a public holiday in country *c?* on date *d?* is:

$$\begin{array}{|ll}
\multicolumn{2}{l}{\text{Abolish}} \\
\Delta Hols & \\
c?: & COUNTRY \\
d?: & DATE \\
\hline
holidays' = holidays \setminus \{c? \mapsto d?\} & \\
\hline
\end{array}$$

Note that *Abolish* ignores the possibility of the date not already being a public holiday in that country.

An operation to find the dates *ds!* of all the public holidays in country *c?* is:

$$\begin{array}{|ll}
\multicolumn{2}{l}{\text{Dates}} \\
\Xi Hols & \\
c?: & COUNTRY \\
ds!: & \mathbb{P}DATE \\
\hline
ds! = holidays(\{c?\}) & \\
\hline
\end{array}$$

9.12 Restriction

Further operators are available to *restrict* a relation.

9.12.1 Domain restriction

The *domain restriction* operator restricts a relation to that part where the domain is contained in a particular set:

$$S \lhd R$$

'the relation *R* domain restricted to *S*'.

9.12.2 Range restriction

The *range restriction* operator restricts a relation to that part where the range is contained in a set:

$R \rhd S$

'the relation *R* range restricted to *S*'.

The point of the operator can be thought of as pointing right to the range.

9.12.3 Domain subtraction

The domain subtraction operator restricts a relation to that part where the domain is *not* contained in a set:

$S \ntriangleleft R$

'the relation *R* domain subtracted to *S*'.

The point of the operator can be thought of as pointing left to the domain.

9.12.4 Range subtraction

The range subtraction operator restricts a relation to that part where the range is *not* contained in a set:

$R \ntriangleright S$

'the relation *R* range subtracted to *S*'.

9.13 Example of restriction

Given the set of all countries and the set of dates and the relation *holidays* as before and the set of countries in the European Union *EU*

$EU: \mathbb{P}COUNTRY$

the relation mapping only EU countries to their public holidays is:

$EU \lhd holidays$

and the relation mapping *non*-EU countries to their public holidays is:

$EU \ntriangleleft holidays$

Given a subset of dates *summer*

$summer: \mathbb{P}DATE$

the relation of countries to public holidays in the summer is holidays $\rhd$ summer

and the relation of countries to public holidays *not* in the summer is

holidays $\ntriangleright$ summer

9.14 Composition

Relations can be joined together by an operation called *composition*. Given a relation *R* which relates *X* to *Y*

R: X $\leftrightarrow$ Y

and a relation *Q* which relates *Y* to *Z*

Q: Y $\leftrightarrow$ Z

the following compositions are possible.

9.14.1 Forward composition

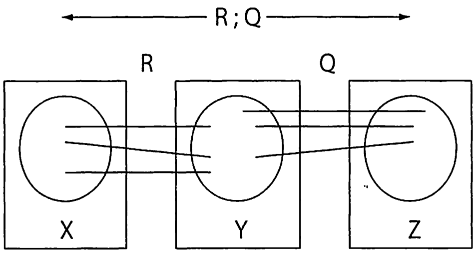
The relation formed by the relation *R*, then the relation Q, is called the *forward composition* of *R* with *Q*:

R: X $\leftrightarrow$ Y

Q: Y $\leftrightarrow$ Z

R ; Q: X $\leftrightarrow$ Z

Figure 9.4



For any pair (x, z) related by *R* forward composed with *Q*

x R ; Q z

there is a *y* where *R* relates *x* to *y* and Q relates *y* to *z*:

$\exists$ y: Y • x R y $\wedge$ y Q z

9.14.2 Backward composition

The *backward composition* of Q with *R*. It is the same as the forward composition of *R* with *Q*:

Q $\circ$ R == R ; Q

It is similar to the mathematical notion of *functional composition*.

9.14.3 Repeated composition

A *homogeneous* relation is one which relates values from a type to values of *the same type* (the source and the target are the same). Such a relation can be composed with itself:

$R: X \leftrightarrow X$
$R ; R: X \leftrightarrow X$

9.15 Example

Countries are related by the relation *borders* if they share a border

borders: COUNTRY $\leftrightarrow$ COUNTRY

For example:

France borders Switzerland
Switzerland borders Austria

Countries are related by *borders* composed with *borders* if they each share a border with a third country

France borders ; borders Austria

since France borders Germany and Germany borders Austria. Note that it is also relates any continental country to itself. (For example, France borders Germany, which borders France.)

The expression:

borders ; borders

can also be written:

France borders^2 Austria

Furthermore:

Spain borders^3 Denmark

means

Spain borders ; borders ; borders Denmark

9.16 Transitive closure

In general, the transitive closure R^+, as used in:

$x \, R^+ \, y$

means that there is a repeated composition of *R* which relates *x* to *y*.

For example:

France $borders^+$ India   which is true

means that France is on the same land-mass as India.

9.17 Identity relation

The *identity relation*

$$\mathrm{id}\ X$$

is the relation which maps all *x*'s on to themselves:

$$\mathrm{id}\ X == \{x: X \bullet x \mapsto x\}$$

9.18 Reflexive transitive closure

The repeated composition

$$= R^+ \cup \mathrm{id}\ X$$

includes the identity relation. The reflexive transitive closure is similar to the transitive closure except that it includes the identity relation.

$x\ R^*\ x$	this is always true, even if x R x is false
France $borders^*$ France	this is true

9.19 Inverse of a relation

The inverse of a relation *R* from *X* to *Y*

$$R: X \leftrightarrow Y$$

is written

$$R^\sim$$

and is the 'mirror image', that is, it relates the same values of *Y* to *X* as *R* relates from *X* to *Y*, so if

$$x\ R\ y$$

then

$$y\ R^\sim\ x$$

9.20 Examples

Family relationships can be defined by means of the notation introduced in this chapter.

9.20.1 Definitions

[PERSON] the set of all persons
father, mother: PERSON $\leftrightarrow$ PERSON

with suitable values and with interpretations:

x father y

and

v mother w

meaning '*x* has *y* as father' and '*v* has *w* as mother'

9.20.2 Parent

The relation *parent* (mother or father)

parent: PERSON $\leftrightarrow$ PERSON

can be defined as the union of the relations *father* and *mother*:

parent = father $\cup$ mother

9.20.3 Sibling

The relation *sibling* (brother or sister)

sibling: PERSON $\leftrightarrow$ PERSON

can be defined as the relation *parent* composed with its own inverse. In other words, the set of persons with the same parents. A person is not usually counted as their own sibling so the identity relation for *PERSON* is excluded:

sibling = (parent ; parent$^{\sim}$) \ id PERSON

9.20.4 Ancestor

The relation *ancestor* can be defined as the repeated composition of *parent*:

ancestor: PERSON $\leftrightarrow$ PERSON
ancestor = parent^{+}

9.21 Summary of notation

X, Y and Z are sets and
x: X;
y: Y;

$R: X \leftrightarrow Y$:

$X \times Y$	the set of ordered pairs of X's and Y's
$X \leftrightarrow Y$	the set of relations from X to Y: $== \mathbb{P}(X \times Y)$
$x \; R \; y$	x is related by R to y: $== (x, y) \in R$
$x \mapsto y$	$== (x, y)$
$\{ x_1 \mapsto y_1, x_2 \mapsto y_2, \ldots, x_n \mapsto y_n \}$	
	$==$ the relation $\{ (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \}$ relating x_1 to y_1, x_2 to y_2, …, x_n to y_n
dom R	the domain of a relation $== \{x: X \mid (\exists y: Y \bullet x \; R \; y) \bullet x\}$
ran R	the range of a relation $== \{y: Y \mid (\exists x: X \bullet x \; R \; y) \bullet y\}$
$R (\!\| S \|\!)$	the relational image of S in R
$S \lhd R$	the relation R domain restricted to S
$R \rhd S$	the relation R range restricted to S
$S ⩤ R$	the relation R domain anti-restricted to S
$R ⩥ S$	the relation R range anti-restricted to S
$R \; ; Q$	the forward composition of R with Q
$Q \circ R$	the backward composition of Q with R
id X	$\{x: X \bullet x \mapsto x\}$
R^+	the repeated self-composition of R
R^*	the repeated self-composition of R, with identity $== R^+ \cup \mathrm{id}\, X$
$R^\sim$	the inverse of R

EXERCISES

In all cases use definitions from this chapter.

1. Express the fact that the language Latin is not spoken in any country (as the official language).
2. Express the fact that Switzerland has four official languages.
3. Give a value to the relation *speaksInEU* which relates countries which are in the set *EU* to their languages.
4. Give a value to the relation *grandparent*.
5. A person's first cousin (or full cousin or cousin-german) is defined as a child of the person's aunt or uncle.

   Give a value for the relation *firstCousin*.

The following questions use these declarations:

[PERSON]	the set of all possible uniquely identified persons
[MODULE]	the set of all module numbers at a university

students, teachers,
EU, inter: $\mathbb{P}$PERSON
offered: $\mathbb{P}$MODULE
studies: PERSON $\leftrightarrow$ MODULE
teaches: PERSON $\leftrightarrow$ MODULE

and the predicates:

EU $\cap$ inter = $\varnothing$
EU $\cup$ inter = students
dom studies $\subseteq$ students
dom teaches $\subseteq$ teachers
ran studies $\subseteq$ offered
ran teaches = ran studies

6. Explain the *effect* of each of the predicates.
7. Give an expression for the set of modules studied by person p.
8. Give an expression for the *number* of modules taught by person q.
9. Explain what is meant by the *inverse* of the relation *studies*:

   studies$^{\sim}$

10. Explain what is meant by the *composition* of relations:

    studies ; teaches$^{\sim}$

11. Give an expression for the set of persons who teach person p.
12. Give an expression for the number of persons who teach both person p and person q.
13. Give an expression for that part of the relation *studies* that pertains to international students (those in the set *inter*).
14. Give an expression that states that p and q teach some of the same international students.
15. The following declarations are part of the description of an international conference:

    [PERSON] the set of all possible uniquely identified persons
    [LANGUAGE] the set of languages of the world

    official: $\mathbb{P}$LANGUAGE

    CONFERENCE
    delegates: $\mathbb{P}$PERSON
    official: $\mathbb{P}$LANGUAGE
    speaks: PERSON $\leftrightarrow$ LANGUAGE

    Write expressions for each of the following (each expression to be independent of the effect of the others):

(a) Every delegate speaks at least one language.
(b) Every delegate speaks at least one official language of the conference.
(c) There is at least one language that every delegate speaks.
(d) There is a delegate who speaks a language that no other delegate speaks.
(e) Write an operation schema to register a new delegate and the set of languages he or she speaks. Include preconditions but do not deal with error conditions.