Introduction

The study of sets so far has been limited by an inability to relate sets to one another. (Merely forming their union or intersection is not relating them).

The concept of a relation is fundamental to set theory and to Z.

An ordered pair is a couple of objects, say (Jim, Joe), in which the order of the objects is significant (Jim, Joe) ≠ (Joe, Jim).

What a Relation is

A relation is a set of ordered pairs. It may be finite or infinite but its most important feature is that it is a <u>set</u>.

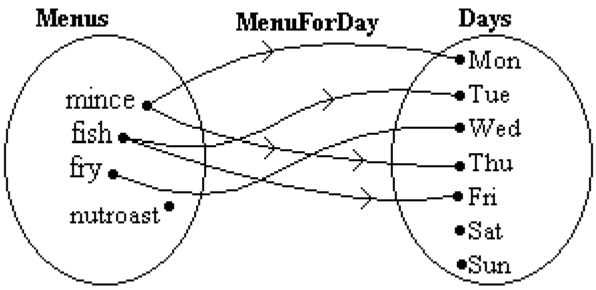
In particular one relation can be a subset of another.

The set {(mince, Mon), (fish, Tue), (fry, Wed), (mince, Thur), (fish, Fri)} is a relation

(We would still have a relation, but a different one if another element such as (Aldridge, bagpipe) were added).

Venn Diagram

Arrows may be used with Venn diagrams to denote relations. Call the above relation MenuForDay. It maps elements of the set Menus to elements of the set Days :



Maplet

Each ordered pair can be represented by a maplet, for which the symbol is $\mapsto$.
Thus (x, y) and $x \mapsto y$ are the same thing.

Cartesian Product
The cartesian product of two sets X and Y, written X **x**Y is the set of all ordered pairs whose first components come from X and second components from Y.
(The infinite set of (x, y) coordinates in a plane is a cartesian product).

$$X \times Y = \{x:X;\ y:Y \bullet (x,y)\ \}$$

Clearly a relation R from X to Y is a subset of $X \times Y$.

The cartesian product $X \times Y \times Z = \{x:X, y:Y, z:Z) \bullet (x,y,z)\}$
is a set of ordered triples.

Sets of ordered n-tuples are defined similarly.

Declaring a Relation
If $R \subseteq X \times Y$ then $R \in \mathbb{P}(X \times Y)$

Since every powerset in Z is by convention a type
R: $\mathbb{P}(X \times Y)$
or R has type $\mathbb{P}(X \times Y)$

Another notation(syntactic sugar) for $\mathbb{P}(X \times Y)$ is $X \leftrightarrow Y$.
Hence R: $X \leftrightarrow Y$
This is the notation we usually use.

In the example on Menus and Days,
MenuForDay : Menus $\leftrightarrow$ Days.

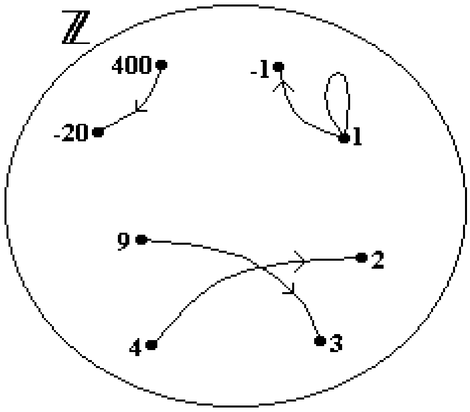
Naming Relations

It is good practice to use a concatenated word suggesting the kinds of entities involved in the relation.

Example:

IsSquareOf = {(1, 1), (1, -1), (4, 2), (9, 3), (400, -20)}
IsSquareOf: $\mathbb{P}(\mathbb{Z} \times \mathbb{Z})$

A relation such as IsSquareOf is called a relation on a set. Its Venn diagram is



Infix Notation

Any relation name may be used as our infix symbol provided it is underlined.

9 <u>IsSquareof</u> 3
Fish <u>MenuForDay</u> Tuesday

Mathematical symbols are not underlined.

$(50, x) \in\ >$ may be written as $50 > x$ (but not as $50 \geqslant x$)

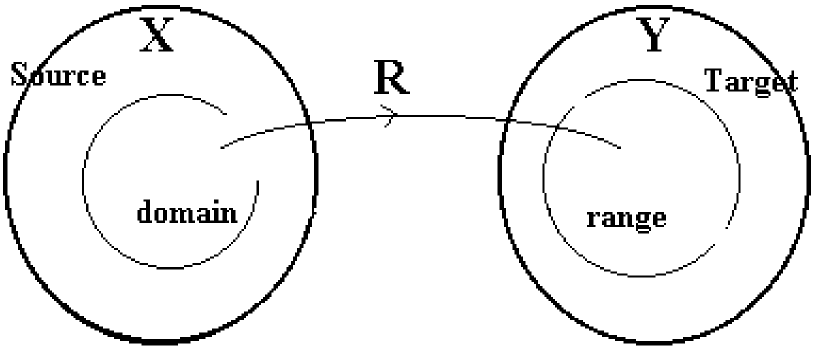
Constant Relations

Some relations such as $\geqslant$ have constant effect, which can be described symbolically. This makes possible an axiomatic definition of the relation, using low-line characters _ for the components, for example

$$\begin{array}{|l}
_ \geqslant _ : \mathbb{N} \leftrightarrow \mathbb{N} \\
\hline
\forall\, m, n : \mathbb{N} \bullet m \geqslant n \Leftrightarrow \exists\, k : \mathbb{N} \bullet m = n + k
\end{array}$$

Domain and Range, Target and Source

Diagrammatically the above four terms mean for a relation R: X $\longleftrightarrow$Y



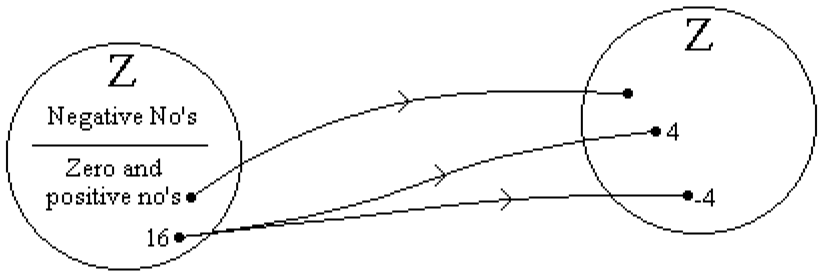
X is the source set or from-set, and Y is the target or to-set.

R acts only on a subset (the **Domain**) of the Source and maps it to a subset (the **Range**) of the Target.

Domain

The domain is the set of first components in the ordered pairs of R.

Consider the relation HasAsSquareRoot : $\mathbb{Z}\longleftrightarrow\mathbb{Z}$



The domain of HasAsSquareRoot is $\mathbb{N}$ not $\mathbb{Z}$.

The domain of the relation R is written dom R.

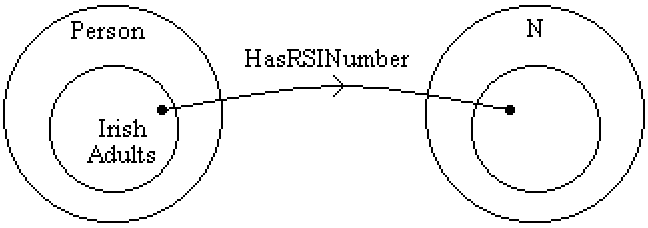
In general the definition of dom is as follows:

R: X$\longleftrightarrow$Y

dom R = { x:X; y:Y | (x,y) $\in$R $\bullet$x}

Range
The range of R written
ran R
is the set of second components in R.

Consider HasRSINumber : Person$\leftrightarrow\mathbb{N}$



The range is a subset of the target N and has at most 2 million elements.

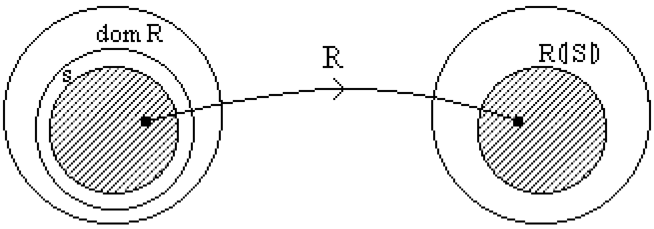
In general the definition of ran is as follows:

R: $X\leftrightarrow Y$
ran R = $\{ x:X;\ y:Y \mid (x,y) \in R \bullet y\}$

Relational Image

We frequently need to know what a certain subset of the domain of R is mapped to. The image of such a subset S is called its relational image, written $R(\!|\ S\ |\!)$

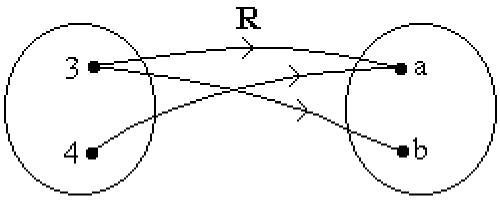
By definition $R(\!|\ S\ |\!) = \{x:X;\ y:Y \mid (x,y) \in R \wedge x \in S \bullet y\}$



This is a very useful operation.

Many - Many relation

Relations in general are many to many mappings. A Venn diagram for the relation R may show for example,



Here many arrows leave a point and many arrows arrive at a point.

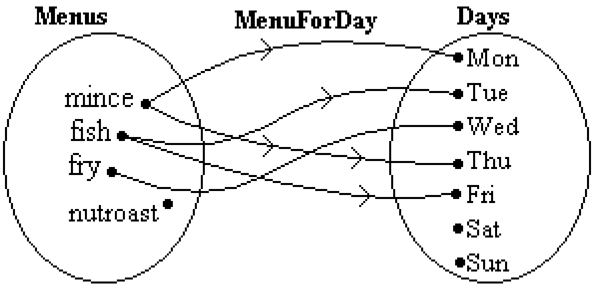
R={(3,a),(3,b),(4,a)}

Restrictions and Anti-Restrictions

There are four important operators on a relation in Z which restrict its sphere of influence.
(For the definitions, we will use the following
R: X $\leftrightarrow$ Y and
S : $\mathbb{P}$X and
T: $\mathbb{P}$Y)

1. Domain Restriction $\lhd$

Consider MenuForDay : Menus$\leftrightarrow$Days



Let Meat ={mince,fry) be a subset of dom MenuForDay
Then Meat $\lhd$ MenuForDay =
{(mince,Mon),(mince,Thurs),(fry,Wed)}
i.e. that subset of MenuForDay where the first components are all from the set Meat.

In general the definition of domain restriction is :
$S \lhd R = \{ x:X;\ y:Y \mid (x,y) \in R \land x \in S \bullet (x,y) \}$

2. Domain Anti Restriction $⩤$

If $S \subseteq \mathrm{dom} R$ then $S ⩤ R$ is the complement of $S \lhd R$ in R.

Thus Meat $\lhd$ MenuForDays is the complement of Meat$⩤$ MenuForDays
Meat$⩤$ MenuForDays
= { (fish,Tue) , (fish,Fri) }
= that subset of MenuForDays where the first components are <u>not</u> in Meat$\lhd$R.

Anti-restriction is useful when a new relation can be defined by specifying the elements of the domain to be excluded.

In general the definition of domain anti-restriction is :
$S ⩤ R = \{ x:X;\ y:Y \mid (x,y) \in R \land x \notin S \bullet (x,y) \}$

3.Range Restriction $\rhd$

If

MenuForDays={(mince,Mon),(fish,Tue),(fry,Wed),(mince,Thu),(fish,Fri)}

and if SpecialDays={Mon,Wed,Fri} then

MenuForDays $\rhd$SpecialDays = {(mince,Mon),(fry,Wed),(fish,Fri)}

i.e. that subset of MenuForDays where the second components are from the set SpecialDays.

In general the definition of range restriction is :

$\mathbf{R \rhd T = \{ x{:}X;\ y{:}Y \mid (x,y) \in R \wedge y \in T \bullet (x,y) \}}$

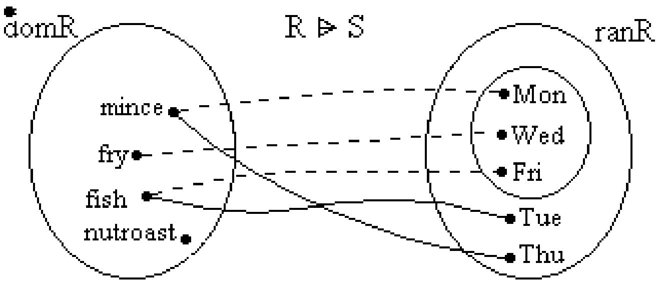
4.Range Anti-restriction $\unrhd$

With MenuForDays and SpecialDays as above

MenuForDays $\unrhd$ SpecialDays is the complement of

MenuForDays $\rhd$ SpecialDays in MenuForDays

So, MenuForDays $\unrhd${Mon.Wed,Fri}={(fish,Tue),(mince,Thu)}



Range anti-restriction is also called range subtraction.

In general the definition of range anti-restriction is :

$\mathbf{R \unrhd T = \{ x{:}X;\ y{:}Y \mid (x,y) \in R \wedge y \notin T \bullet (x,y) \}}$

Inverse of a relation.

If $R : X \leftrightarrow Y$

then $R^{\sim}$ is the set of inverse ordered pairs

$R^{\sim} = \{x{:}X;\ y{:}Y \mid (x,y) \in R \bullet (y,x) \}$

Example

R = { (a,1), (b,2), (a, 4) }

then

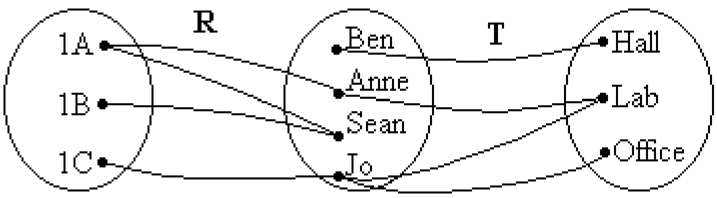
$R^{\sim}$ = {(1, a), (2, b), (4, a) }

Composition

To compose two relations R and T means to activate one, and then the other using the result of the first. Certain interfacing conditions must be met.

Forward Composition ⨾

R⨾T means "do R first then T". T uses ranR, or a subset of ranR as input.
The result is a relation, RÀT from domR to ranT.



R⨾T={(1A,Lab), (1C,Lab), (1C,Office)}

For each pair in R⨾T there is some "stepping-stone" y in ranR » domT.
The types of ranR , domT must be the same.
The type of R⨾T is type of domR x type of ranT.

Repeated Composition
A homogeneous relation is one where the domain and range have the same type. It may be possible to compose such a function with itself :

$R : X \leftrightarrow X$
$R ⨾ R : X \leftrightarrow X$

For example if the relation is

$addOne : \mathbb{Z} \leftrightarrow \mathbb{Z}$
$addOne == \{ (x:\mathbb{Z} \bullet (x, x+1) \}$

so that 6 <u>addOne</u> 7 and 7 <u>addOne</u> 8 etc.

then $addOne^2 == addOne ⨾ addOne$
$addOne^3 == addOne^2 ⨾ addOne$
and so on.

As an example,
6 $addOne^2$ 8 , . . . , 6 $addOne^2$ 13 , ...

In general
$R^+ = R \cup R^2 \cup R^3 \cup \ . \ . \ . \cup R^n \cup \ . \ . \ . \ .$
These may be empty for n⩾2.

$addOne^+ = addOne \cup addOne^2 \cup addOne^3 \cup addOne^4 \cup \ldots.$
means that there exists a repeated composition of addOne which maps x to y.

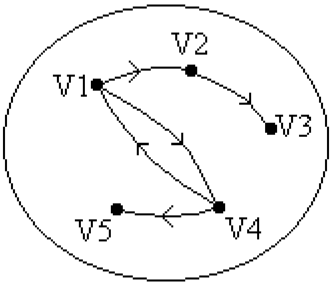
so x $\underline{addOne}^+$ y
means that there exists a repeated composition of addOne which maps x to y.

which means that addOne is another name for the relation ‘<’.

Identity Relation Id
Id X == {x:X ● (x,x) }
We can apply Id to any type,
e.g. id PERSON = {p:PERSON ● (p,p) }

Transitive Closure = R^+
As an example of Transitive Closure consider the relation
P = IsCableConnectedTo on a set of villages $V_1, \ldots, V_5$.
Assume that the relation is one way.

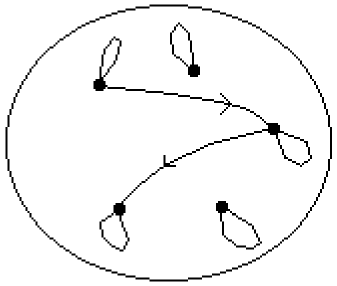
If P = $\{ (V_1,V_2), (V_1,V_4), (V_2,V_3), (V_4,V_1), (V_4,V_5) \}$, as shown



then the transitive closure of P is P $\cup$ $\{ (V_1,V_3), (V_4,V_3), (V_4,V_2), (V_1,V_5) \}$
$= P \cup P^2 \cup P^3$.

Reflexive Relation

A relation R is reflexive if $(a,a) \in R$ for every a in R



R reflexive shows a loop at each element of dom R

Reflexive-transitive Closure R^*

The reflexive-transistive closure of R is the smallest extension of R which is both transitive and reflexive. Clearly If we add a loop to each element in domR^+ we have the reflexive-transitive closure R^*

Since Id maps a to a

$$Id \cup R^+ = R^*$$

Finally, since $Id = R^o$ and $R^+ = \bigcup_{K \geqslant 1} R^K$

$$R^* = Id \cup R^+ = R^o \cup \bigcup_{K \geqslant 1} R^K = \bigcup_{K \geqslant 0} R^K$$

Hence $R^* = \bigcup_{K \geqslant 0} R^K$

addOne* = addOne $\cup$ id $\mathbb{Z}$

so this is another name for $\geqslant$

Examples

1. **Borders**

Countries are related by the relation borders if they share a border:
[COUNTRY]
borders : COUNTRY $\longleftrightarrow$COUNTRY

e.g. ***borders*** = {… (france, switzerland), (switzerland, austria), (austria, hungary), (hungary, romania), (romania, bulgaria), (bulgaria, turkey), (turkey, iran) …… , (peru, brazil), (brazil, paraguay),(paraguay, chile)…..}

So, you can see that

france $\underline{\textbf{\textit{borders}}^{+}}$ iran

This is another way of stating that france is directly bordering iran or is bordering a country which is bordering a country.. which is bordering iran. So, we can reach iran from france (i.e. it is on the same landmass).

We can also see that

peru $\underline{\textbf{\textit{borders}}^{+}}$ chile,

so peru and chile are on the same landmass.

but

(france,peru) $\notin \underline{\textbf{\textit{borders}}^{+}}$

which means that they are not on the same landmass.

This concept of connectivity is central to the idea of transitive closure. We can model clusters of connected elements by using transitive closure.

2. **Airports**

[AIRPORT]

connected = {… (lhr, dublin), (dublin, jfk), (jfk, rome) …}

If we wish to state that we can get to rome from lhr, then we state that

lhr $\underline{\text{connected}^{+}}$ rome

This does not give us any hint as to how many hops we need, just that it is possible to get there.

We will revisit this example later when we look at building a route from one airport to another.

3. Citing papers

Given
[PAPER]
and cites: PAPER $\longleftrightarrow$PAPER
and (paper1, paper2) $\in$cites has the meaning that paper1 cites paper2 .

Write the following

a. Write a Z expression for the set of all papers cited directly or indirectly by paper x.

---

b. Write a Z expression for the set of all papers which cite other papers(directly or indirectly) but themselves are not cites (directly or indirectly).
c. Write a Z expression which states that if any paper cites another (directly or indirectly) the second one must not cite the first (directly or indirectly).
d. Write a Z expression for the number of papers cited directly by paper x.
e. Write a Z expression for the number of papers cited directly or indirectly by paper x.

4. Family relations

Given
[PERSON]
and
parent : PERSON $\longleftrightarrow$PERSON
male, female : $\mathbb{P}$PERSON
and that (abe, homer) $\in$ parent means that abe is homer's parent.

A.Write a Z expression for the following:

a. The parents of person x.
b. The grandparents of person x.
c. The grandchildren of person x.
d. The descendants of person x.
e. The siblings of person x.
f. The aunts of person x.

B. Give a Z expression for the set of all people in the database who have no relatives in the database.
C. Write an invariant to say that no person can have more than 2 parents.