QUT

CAB203
Tutorial 6

1 Relations

1. Write out the following sets in full:

   (a) $\{0,1\} \times \{0,1\}$

   Solution:
   $$\{(0,0),(0,1),(1,0),(1,1)\}$$

   (b) $\{a,b\} \times \{0,1\}$

   Solution:
   $$\{(a,0),(a,1),(b,0),(b,1)\}$$

   (c) $\{\alpha,\beta,\gamma\} \times \{a,b\}$

   Solution:
   $$\{(\alpha,a),(\alpha,b),(\beta,a),(\beta,b),(\gamma,a),(\gamma,b)\}$$

2. For each of the following relations determine if it is symmetric, reflexive, transitive, anti-symmetric or irreflexive, and whether it is a equivalence relation, partial ordering or total ordering. If it is an equivalence relation, describe the equivalence classes.

   (a) $R = \{(0,0),(0,1),(1,1),(1,2),(0,2),(2,2)\}$ on the set $\{0,1,2\}$

   Solution: Reflexive, anti-symmetric, transitive. Total ordering.

   (b) $R = \{(0,0),(2,2),(1,1),(3,3),(0,1),(1,0),(2,3),(3,2)\}$ on the set $\{0,1,2,3\}$

   Solution: Reflexive, symmetric, transitive. Equivalence relation. Equivalence classes $\{0,1\}$ and $\{2,3\}$.

   (c) $R = \{(0,0),(1,0),(2,0),(1,1),(2,2)\}$ on the set $\{0,1,2\}$

   Solution: Reflexive, antisymmetric, transitive. Partial ordering.

   (d) $R = \{(a,b) \in \mathbb{N}^2 : \exists c \in \mathbb{N}\ ac = b\}$ on the set $\mathbb{N}$

   Solution: aRa for all a by setting $c = 1$, so it is reflexive. If $ac = b$ (aRb) and $bc' = a$ (bRa) then $bcc' = b$ so $cc' = 1$. This can only happen if $c = c' = 1$, so we get $a = b$. Thus R is anti-symmetric. If $ac = b$ (aRb) and $bc' = d$ (bRd) then $a(cc') = d$ so aRd and the relation is transitive. Thus this is a partial ordering. It is not a total ordering since we have neither $3R5$ or $5R3$, for example.

(e) R defined on the set of all cars where aRb if a and b have the same size of tyres

Solution: R will inherit most of the properties of the underlying relation on tyre sizes, i.e. equality. If a and b have same tyre size, as do b and c, then a and c will also have the same tyre size. a always has the same tyre size as itself. And if a and b have the same tyre size, then b and a have the same tyre size. So this is an equivalence relation The equivalence classes are the sets of cars with a particular tyre size.

We can generalise, and say that if we have some function f that defines a property of elements of a set s, then if we define a relation where two elements are related if that property f is equal for both elements, the relation will be an equivalence relation. The equivalence classes will be the sets of elements which have the same property. It is often easier to specify the equivalence classes: they are $S_p = \{x \in S : f(x) = p\}$ where p is in the range of f.

(f) R defined on the set of all people, where aRb if b has lived in all the cities that a has lived in.

Solution: Again, R will inherit many properties from the underlying relation. In this case we are looking at sets of cities. Let S_a be the set of cities that a has lived in. Then aRb when $S_a \subseteq S_b$. From this we can straightforwardly see that R will be reflexive, and transitive. However, R will not by anti-symmetric. For example, we might have two people a and b who have only lived in Brisbane. Then we get both aRb and bRa. But R is not symmetric either, since there is at least one person b who have lived in both Brisbane and Sydney, and at least one person a who has only lived in Brisbane. Then aRb but not bRa.

So R is not an ordering of any type, or an equivalence relation.

(g) R defined on the set of cities, where aRb if there is a regularly scheduled direct flight from a to b or from b to a

Solution: There are no direct flights from a city to itself (maybe in strange circumstances, but not scheduled!) so $(a, a) \notin R$ for all a. Thus R is irreflexive. From the definition we get symmetry straightforwardly: if there is a direct flight from a to b then this gets us both aRb and bRa. R is not transitive, though. There are direct flights from Sydney to Brisbane, and from Brisbane to Dunedin, but not from Sydney to Dunedin.

This type of relation (irreflexive and symmetric) is called a *graph*.

(h) *(Stretch question)* Define S by the set of formulas of the form $f(x) = ax + b$ where $a, b \in \mathbb{R}$. Then define the relation R on S by fRg if $f = g$ or if there does not exist $x \in \mathbb{R}$ such that $f(x) = g(x)$.

Solution: Clearly from the definition R is reflexive. Also, the definition doesn't care about the order of f and g, so R is symmetric.

Suppose $f \neq g$. If $f(x) = ax+b$ and $g(x) = cx+d$ then $f(x) = g(x)$ means $(a-c)x = b-d$. This has a solution as long as $a - c \neq 0$. So if $a = c$ then there is no solution (unless $b = d$ but then $f = g$) and fRG. The converse also applies: if there is no solution then $a = c$. If $a \neq c$ then there is a solution and $\neg fRg$. So we can say fRg if and only if the a parameters of f and g are the same, i.e. the lines have the same slope. Note this characterisation also works when $f = g$.

Now if we have f, g, h such that fRg and gRh, then their a parameters (i.e. slope) must all be the same, so fRh as well, and R is transitive.

A less formal way to look at this is that the formulas specify lines in the 2-dimensional plane (except for vertical lines). fRg when the lines for f and g do not intersect, or when

f and g are the same line. That is to say, f and g are parallel. From this it becomes clear that R partitions the lines into sets which are all parallel to each other, and hence this is an equivalence relation.

So we know that R is an equivalence relation. The equivalence classes are the sets of lines which are parallel to each other.

(i) *(Stretch question)* Define S by the set of formulas $f(x) = ax^2 + bx + c$, and the relation R on S by fRg if $\forall x \in \mathbb{R}\ f(x) \leq g(x)$.

Solution: From the definition it is easy to see that R is reflexive as $f(x) \leq f(x)$ for all x. For anti-symmetry, suppose that $f \neq g$, fRg and gRf. Then for all x we have both $f(x) \leq g(x)$ and $g(x) \leq f(x)$, so $f(x) = g(x)$. This only happens when $f = g$. Finally, for transitivity, if for all x we have $f(x) \leq g(x) \leq h(x)$ then we have for all x $f(x) \leq h(x)$ so fRh.

So we know that R is at least a partial ordering. It is not a total ordering, because we can set $f(x) = x$ and $g(x) = -x$. Then for $x = 1$ we have $f(x) > g(x)$ and for $x = -1$ we have $f(x) < g(x)$, so neither fRg nor gRf.

(j) *(Stretch question)* Let S be a non-empty set and let $f : S \to \mathbb{Z}$ where for each $x \in \mathbb{Z}$ there is at most one $s \in S$ such that $f(s) = x$. To put it differently, if $f(s) = f(t)$ then $s = t$. Define a relation R on S by sRt when $f(s) \leq f(t)$.

Solution: Here R will again inherit many properties of the underlying relation $\leq$. First, R is clearly reflexive since $f(s) \leq f(s)$. It is also transitive since $f(s) \leq f(t) \leq f(u)$ implies $f(s) \leq f(u)$. If $f(s) \leq f(t)$ and $f(t) \leq f(s)$ then $f(s) = f(t)$ and $s = t$. So R is antisymmetric. Thus R is at least a partial ordering. It is also a total ordering, since for any integers $f(s)$ and $f(t)$, we have either $f(s) \leq f(t)$ or $f(t) \leq f(s)$.

3. *(Stretch question)* There is at least one relation on any set A that is symmetric, antisymmetric, transitive, and irreflexive. What is it? Is it the only one? Is there any relation that is both reflexive and irreflexive?

Solution: Recall that the empty set $\emptyset$ is a relation over any set A. Trivially it is irreflexive since $(a, a) \notin \emptyset$ for any a. Also trivially it is symmetric, anti-symmetric and transitive. For example, trasitivity requires that for any a, b, c if $a\emptyset b$ and $b\emptyset c$ then $a\emptyset c$. But since the condition is never satisfied this is always the case. (Recall that $p \to q$ is always true when p is false.)

How about reflexivity? If $A \neq \emptyset$ then $\emptyset$ is not refelxive since we don't have $a\emptyset a$ for any a in A. However, if $A = \emptyset$ then trivially $\emptyset$ is reflexive, and also irreflexive!

4. In Python, we might encode a relation over a set A as a Python set containing Python tuples, each of which is length 2 where the both elements are in A. For example, a relation might look like:

```
R = { (1, 3), (2, 1), (3, 4) }
```

Write a short Python function that takes a relation as above and returns the set A that R is over. Note that relations do not have to refer to every element of the set that they are over, (i.e. they do not need to have some (a, b) or (b, a) for every $a \in A$) so the set you return will just be the set of elements that R refers to.

Solution:

```
def underlyingSet(R):
   return { a for (a,b) in R } | { b for (a,b) in R }
```

5. Write a Python function which, given a relation in the format described in the previous question, and returns a pair (a, b) where a is True if the relation is symmetric and False otherwise, and b is True if the relation is anti-symmetric and False otherwise.

Solution:

```
def antisymmetric(R):
   # looks for pairs (a,b) in R that have a matching (b,a), but only if a != b.
   # Note ((b,a) in R) will be True or False
   symmetricpair = { ((b,a) in R) for (a,b) in R if (a != b)}
   # symmetric means the matching pair is always there: no Falses
   # anti-symmetric means the matching pair is never there: no Trues
   return ( False not in symmetricpair, True not in symmetricpair)
```

6. Write a Python function which, given a relation in the format used in the previous questions, returns True if the relation is transitive, otherwise False.

Solution:

```
def transitive(R):
   # look for cases (a,b),(b2,c) in R, but only where b == b2
   # we'll check whether the required (a,c) is present
   cases = { (a, c) in R for (a,b) in R for (b2, c) in R if b == b2 }
   return False not in cases
```

2 Functions

1. For each of the following, determine whether the given relation is a function on the given set. If it is a function, determine whether it has an inverse, and the range.

   (a) $\{(0,1),(1,2),(2,0)\}$ on the set $\{0,1,2\}$

   Solution: It is a function, and has an inverse. The range is $\{0,1,2\}$

   (b) $\{(1,0),(2,1),(1,2),(0,1)\}$ on $\{0,1,2\}$

   Solution: It is not a function, since 1 appears twice on the left.

   (c) The set $R \subseteq V \times C$, where V is the set of voters in some election, and C is the set of candidates, so that (v, c) is in R if v voted for c. (In this election, voters can vote for at most one candidate. To be considered a voter, a person must vote for some candidate.) Assume $|V| > |C|$.

Solution: For every $v \in V$ we have at least one (v, c) in R, otherwise v wouldn't be considered a voter. Now, since v can vote for at most one candidate, there are no other (v, c') in R. So R is a function. The range of R is the set of candidates who received at least one vote.

Since $|V| > |C|$ there must be at least some $c \in C$ which received at least two votes (this is called the pigeon-hole principle.) Then there are two distinct $u, v \in V$ such that $(u, c), (v, c) \in R$. This means that R cannot have an inverse.

(d) With the above definition, does anything change if we allow a voter to vote for more than one candidate? How about if we include people who don't vote for anyone?

Solution: If we allow a voter to vote for more than one candidate then we no longer have a function. There will not be a unique c so that $(v, c) \in R$. If we include people who don't vote at all, then there may not be any c such that $(v, c) \in R$. In this case we may still have a function if we strict the domain to the set of people who cast a vote.

(e) The set $\{(x, y) \in \mathbb{R}^2 : 2x + y = 3\}$ on the set $\mathbb{R}$

Solution: Let us suppose that (x, y) and (x, z) are in the set. Then we can write

$$\begin{aligned} y &= 3 - 2x \\ z &= 3 - 2x \end{aligned}$$

so $y = z$. Thus this is a function. We can rearrange the equation to look like

$$x = \frac{3 - y}{2}.$$

This gives us two pieces of information. First, we can put any $y \in \mathbb{R}$ into this formula to find an x such that (x, y) is in the set. Thus the range of our function is $\mathbb{R}$. Also, the formula gives us a unique x for each y, so this function has an inverse.

(f) The set $\{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 4\}$ on the set $\mathbb{R}$

Solution: Straightforwardly, $(\sqrt{2}, \sqrt{2})$ is in the set. But so is $(\sqrt{2}, -\sqrt{2})$. So this is not a function.

2. Write a Python function which, given a relation in the format used in the previous section, a set A, a set B returns (a, b) where a is True if the relation is a function from A to B, otherwise False, and b is True if the relation is a function from A to B and has an inverse, otherwise False.

Solution:

```
def numimages(R, a):
   """Returns the size of the set of all b such that (a,b) is in R.
   This will be 1 for all a in A if R is a function."""
   return len( { b for (a2,b) in R if a2 == a } )

def isfunction(R, A, B):
   isf = { 1 } == { numimages(R,a) for a in A }
   R2 = { (b,a) for (a,b) in R }
   hasinv = isf and { 1 } == { numimages(R2,b) for b in B}
```

```
    return (isf, hasinv)
```

3 Sequences

1. For each of the following, which is the better choice, a list or a tuple?

   (a) x,y,z-coordinates for vertices of a 3D model

   Solution: Tuple. This is fixed length, and the three coordinates are part of a meaningful whole.

   (b) Bus numbers for busses arriving at a bus stop.

   Solution: List. The length might change depending on the buses and the stop. Also, the list is likely to change and updating is probably more convenient than replacing the entire list. The busses don't really relate to each other.

   (c) First name, Last name, student number

   Solution: Tuple. These items all relate to each other, and the length will not change. Also, changes to the data is unlikely, so immutable is fine.

2. Write a collection of functions that implement a mini student database. The functions are:

   (a) `createD()` Return an empty database

   (b) `addStudent(D, student)`: Add a student to the database

   (c) `studentName(D, number)`: retrieve the students first and last name from the database by student number

   `D` is some data structure (you get to decide what it should be) representing the database. `student` is a data structure representing an individual student, and should include student number, first name and last name.

   To keep things simple, your functions are allowed to have undefined behaviour when given inappropriate inputs. You can assume, for example, that a student number querried will always be in the database, student numbers are unique, etc..

Solution: We will use a dictionary to store the database. This will make it easy to implement `studentName` and there are no other functions that require any other kind of functionality. Other reasonable choices would be a set or a list. Set is the best fit since there is no inherent ordering, but it does require more work than a dictionary for our purposes.

```
# A database will be a dictionary where keys are student numbers
# and values are (firstname, lastname) tuples
def createD():
    return dict()

# A student will be a tuple: (number, firstname, lastname)
def addStudent(D, student):
```

```
    number, firstname, lastname = student
    D[number] = (firstname, lastname)

def studentName(D, number):
    return D[number]
```