UNIVERSITY OF WATERLOO

Faculty of Mathematics
Waterloo, Ontario N2L 3G1

Centre for Education in
Mathematics and Computing

Grade 6 Math Circles

November 6 & 7 2018

Relations, Functions, and Morphisms

Relations

Let's talk about relations!

A ***relation*** is: A general way to relate things from different sets.

A ***set*** is: A collection of things: any collection can be called a set.

The ***elements*** of a set are: The things inside a set.

As with understanding anything, it's important to develop an *intuition* for how these work. A helpful way to think about a relation is as a set of *paths* that tell you where you'll end up <u>from</u> where you started. Up to now, you should be most comfortable dealing with *operators*, like addition and multiplication. These operators manipulate the elements that are in one set, like the set of all numbers. A relation on the other hand connects elements that can be from different sets.

Example

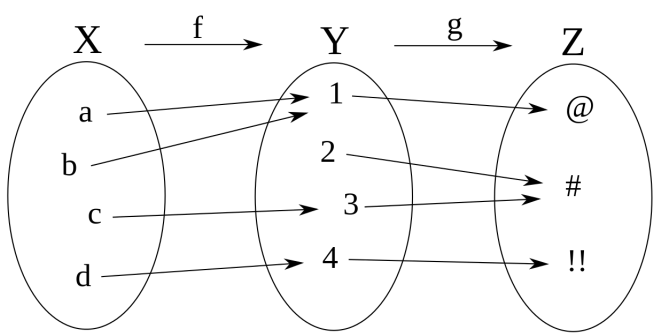

Figure 1: There are 3 sets here: X, Y, and Z. In this diagram, f is a relation that goes from X to Y, and g is a relation that goes from Y to Z.

*Note: Usually sets are written by putting the objects inside a pair of curly brackets { }. We'll draw them as ovals in our diagrams.

This example is just connecting the things in the different sets, creating a relation. A relation is very general. Things in sets just need to be connected in some directed way to have a fully defined relation.

Parts of a Relation

binary relation is: A relation that only goes between two sets. The word *binary* just means that there are two things, in this case two sets.

The ***domain*** binary relation is: The set that the paths of the relation start from.

The ***codomain*** binary relation is: The set that the paths of the relation end on.

A binary relation is still really general. Relations that connect more than 2 sets can usually be split into more than one binary relation instead. Domain and codomain are only used to describe the sets of starting and ending points of binary relations, since that's where the words make sense. In a relation that is not binary, you wouldn't know which set I'm referring to when I use these words.

In the example above, the relation labelled "f" on its own is a binary relation, with domain X and codomain Y. The relation "g" is also a binary relation with domain Y and codomain Z

Types of Relations

So far we've just talked about general relations. These very general relations are not actually very useful yet. To have relations that are useful, we need to add more restrictions, more rules that we want relations to follow. One way of doing this separating relations into types. Relations can be classified into different types depending on how they connect elements from sets together.

One-to-One (Injective)

A one-to-one relation does two things:

1. Connects each element in its domain with only one element in its codomain
2. Makes sure each element in its codomain comes from at most one element in its domain

Notice that this means that every element in the domain does connect to something. In particular, every element in the domain connects to one and only one thing. **BUT**, every element in the codomain does not need to connect to something. Every element in the codomain that does connect to something, though, can only be connected to one thing. This is also called being *injective*.

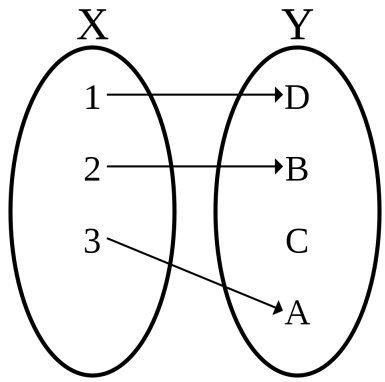

Figure 2: This is an injectyive relation. Every element that has a line reaching it only has one line, and every element in the domain X has one line leaving from it.

Onto (Surjective)

Relations that are onto have that every element in the codomain is connected to *something* in the domain. In other words, any element in the codomain could be traced back to an element in the domain. Onto relations are also called *surjective*.

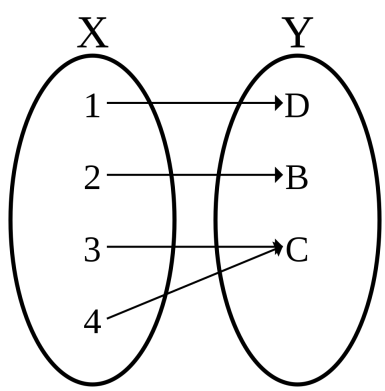

Figure 3: This is a surjective relation. Every element in the codomain comes from something in the domain.

All the specific elements that a relation connects to in a codomain can theselves also be placed into a set, called the ***range*** of the relation. This means that any relation where the codomain the range is onto, so any relation can be made onto just by removing the parts of the codomain that don't "come from" the domain.

Bijective

A bijective relation is a both one-to-one **AND** onto. This means the relation would look something like this:

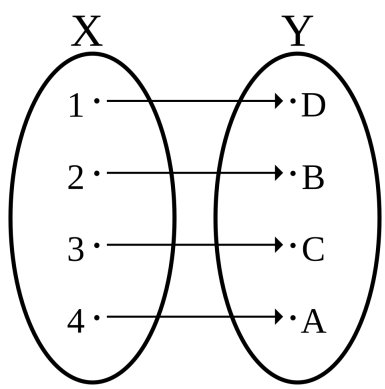

Figure 4: This is a bijective relation. It is both injective and surjective.

You can see that for a bijective relation, its domain and codomain will always have the same size.

Relations and Patterns

The most useful relations are often defined by patterns. Instead of telling you individually where each starting point would take you, relations can tell you instead what the road will look like from where you start. For example, we can define a "+2 relation" which takes whatever number you start with and adds 2 to it.

If we decide that our domain is any integer, and our relation adds 2 to anything we start with, then the codomain will also be any integer. This is because no matter what integer you pick, once you add 2 to it, your answer will definitely be an integer. This relation would usually be defined by its domain and codomain like this:

$$+2 \text{ relation} : (Integers) \longrightarrow (Integers)$$

Variables

Let's say the relation we want to define has a domain with an infinite number of elements. This would be like the "+2 relation" above, with the domain as any integer. Drawing out this relation like we have so far would be impossible, since we can't write out *all* the integers. The integers go on forever. This relation only makes sense because it follows a pattern that tells you where you will end up from your starting place, no matter where you started. To be able to talk about these sorts of relations, *variables* become important tools. Throughout the last few lessons we've used letters to talk about numbers or quantities that we don't know, either because we haven't measured them yet, or because we're talking about any specific thing from a group of things. These letters are called variables. Variables are used in the same ways we've used them so far, and are most often single letters or symbols.

Using a variable then, we can write the pattern that the "+2 relation" follows as:

$$x \longmapsto x + 2$$

Where x is an element of the domain of the relation. We can also write this pattern using 2 variables as:

$$y = x + 2$$

Where x is an element of the domain of the relation, and y is an element of the codomain. From this you can see that any equation that you can right is a relation of some sort. Relations are useful indeed!

This shows the importance of knowing the domain and codomain of a relation as well. If the domain was any number at all, then the possible values and the codomain would be very different from if the domain was only any integer.

Relations vs. Operations

Operations are defined between the elements of a set, and are entirely internal. They tell you what you can do with the elements you have. Relations are defined *between* sets, so in a way are external. They just connect elements from possibly different sets together.

Like with relations, the concept of a set is so general, though, that it isn't really useful unless you restrict it in some way. Applying restrictions by coming up with precise definitions and conditions that sets have to follow makes sets much more useful. Depending on the restrictions, you can be looking at different *mathematical structures*. Different types of

mathematical structures use different operators.

An example of this that we've already looked at is with vectors. We talked about vectors before, and how we can do vector addition and subtraction. Vectors are just the elements of a type of mathematical structure called a vector space. The way addition and subtraction works for vectors is different from what it's like for numbers because the operators have different definitions. They are a different type of structure.

One more thing to understand here is the difference between a mathematical structure, and a *type* of mathematical structure. All vectors belong to some vector space, and all vector spaces are the same *type* of mathematical structure. However, the set of all 2D vectors is **not** the same structure as the set of all 3D vectors. They are the same *type* of structure with operations defined the same way, but different and independent structures in themselves.

Idea: Manipulating Structures

Just like the elements inside a set can be manipulated with operations depending on what kind of structure the set is, we're interested in seeing how we can manipulate the structures themselves. One way we can do this is using relations. We're going to work towards looking at how *two structures can be equal*.

Functions

An important type of relation is a function. What makes a relation a function? A function is a relation where each element in the domain of the relation is connected to one element in the codomain. Functions are the most useful type of relation, and are used everywhere in science and math.

Inverse of a Function

Functions only care about the domain being a certain way. What if the condition of being a funciton was true both ways? This way, you would also be able to define an inverse function. An inverse function is a function that *undoes* the original function. This would go from the codomain to the domain of the original function. You can see based on the conditions that this would need to be a *bijective relation*.

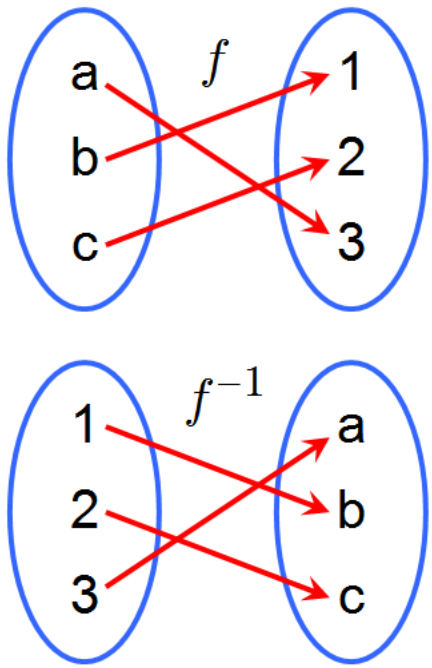

Figure 5: A function and its inverse. You can see that both are bijective.

Morphisms

a ***morphism*** is: a function that preserves the structure between its domain and codomain.

The way this works is: take elements a, b, and a+b from Set 1; and take elements c, d, and c+d from Set 2. Also take a "*R relation" that follows the pattern of doing *R to whatever in the domain you start with (*R can be anything, so we're using another variable to talk about it). The domain of the "*R relation" is Set 1, and its codomain is Set 2. For the "*R relation" to be a **morphism**, it needs to be true that:

IF $c = a * R$ **AND** $d = b * R$

THEN $(c + d) * R = (a + b) * R$

This needs to be true for any a and b you choose from the domain. We're going to use this as our one needed condition.

Isomorphism

An ***isomorphism*** is: A morphism that has an inverse function that reverses it.

Let's think about what this means for us.

- A morphism needs to be a function to begin with
- If a function has an inverse, then we know (in general) that the function must be a bijective relation
- If you have a bijective relation between two sets, then those sets must have the same size.

This means an isomorphism is (in general) a function that preserves structure between two sets that have the same size. Sets that have an isomorphism between them are called ***isomorphic***.

Structures that are isomorphic are *equivalent* in every way that the isomorphism preserves. The structures may contain different types of objects, and not everything about the way they work with things outside the structure may be the same. But, in terms of the operations of the structures the isomorphism preserves, the structures can be considered mathematically identical.

Note: The topic of isomorphisms is complicated and has a lot more to it than just discussed, with a lot of uses in complicated mathematics. This lesson is just meant to give you a feel for how it works!

Problems

1. What are the new terms that you've learned in this lesson? Try to give a definition for each of them.
   The new terms this lesson are: Set, Element, Relation, Domain, Codomain, Variable, Mathematical Structure, Function, Morphism, Isomorphism. See the lesson for definitions of each of these terms.

2. Give some examples of relations. Include relations that are functions, one-to-one, onto, and bijective
   See the lesson above for some examples of these types of relations. remember that relations, most generally, don't have any rules. Try getting creative!

3. Give some examples of relations that follow a pattern. Write the relations formally, giving the domain, codomain, and range of each relation.
   The example I will give here is an "x^2 relation". This relation has:

   $$x^2 \text{ relation} : \textit{(Real Numbers)} \longrightarrow \textit{(Real Numbers)}$$

   The domain and codomain of the function is the set of all real numbers (any number you could find on a number line). The *range* of the relation is actually just the positive real numbers (so any number on a positive number line). This is because the square of any real number is never a negative number.

4. For a "$\times 2$ relation" that multiplies whatever input you give it by 2, what would the relation's range be if the domain is all real numbers (any number on a number line)? What is the range if the domain is any integer?
   If the domain is any real number, then the range would both also be any real number. If the domain is any integer, then the range would be any *even* integer, because double any integer is an even number.

CHALLENGE

5. Show using an example with specific numbers that the "$\times 2$ relation" is a morphism that preserves addition when the domain is all the integers. Is it also an isomorphism? If so, give the inverse function to prove that it is. If not, explain why not. Give a description of what this means for the domain and codomain of the relation.

If the domain is any integer, then the range would be any *even* integer, because double any integer is an even number. Taking the range as the codomain, the relation becomes onto/surjective. We also know that any one integer $\times 2$ only gives one number as its answer, and that any integer *can* be multiplied by 2. This means that the relation is also one-to-one/injective. That makes this a bijective relation. Any bijective relation is also a function (since fuctions have less rules to follow), and any bijective function has an inverse function. So, this relation should be an isomorphism from the set of all integers to the set of all even integers. We can use specific numbers to show that the addition is preserved:

$$(3 + 4) \times 2 = 14 = (3 \times 2) + (4 \times 2)$$

To show it really is an isomorphism, we need to find its inverse. The inverse of the "$\times 2$ relation" should be the "$\div 2$ relation".

Note that this doesn't mean that the set of all integers and the set of all even integers are exactly the same. We know for a fact that they aren't. But their *addition structures* are equivalent, and that is all we've shown here.