Chapter 8

Relations

As a reader of English prose, the word 'relation' might cause you to think about your relatives — parents, children, spouse, etc. – because you are *related* to them biologically or legally. This mental association is useful in this chapter, too, because we can represent such connections, and many others, with *relations*.

Before reading on, we need to ask: How confident do you feel about sets in general, and Cartesian Products in particular? The Math Review appendix and the Additional Set Concepts chapter can help prepare you for this chapter's content. For that matter, reviewing the Matrices chapter wouldn't hurt, either.

8.1 (Binary) Relations

The simplest variety of relation is a *binary relation* (a.k.a. *dyadic*, *2–adic* or *2–ary* relation), and is so named because such relations are sets of pairs of elements. If you're guessing that this is where Cartesian Products come in to the story, you're right.

(binary) relation

Definition 53: (Binary) Relation

A *(binary) relation* from a set X (the *domain*) to a set Y (the *codomain*) is a subset of the Cartesian Product of X and Y ($X \times Y$).

In general, a binary relation has two *base sets*, the 'from' base set (the domain) and the 'to' base set (the codomain). In many relations, the domain

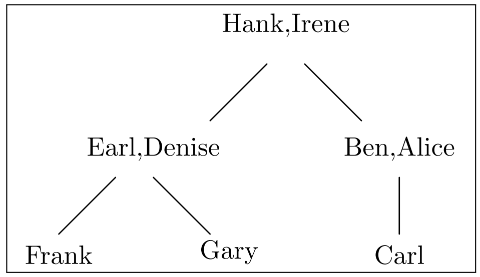


Figure 8.1: A Family Tree

and codomain are the same set, simply called the base set of the relation. In that situation, the wording "on set W" is usually used instead of "from set W to set W." The phrases have the same meaning; the former is just less awkward to write or speak.

Example 151:

Figure 8.1 shows a simple family tree. Hank and Irene have two children, Denise and Ben. Denise and Earl also have two children, Frank and Gary, while Ben and Alice have one child, Carl. The set of family members is $M = \{\text{Alice}, \text{Ben}, \text{Carl}, \text{Denise}, \text{Earl}, \text{Frank}, \text{Gary}, \text{Hank}, \text{Irene}\}$.

We can define a binary parent relation, P, on the set M (that is, from M to M) by creating a set of ordered pairs such that the left member of the pair is a parent and the right member is a child of that parent. That is, P will contain the ordered pair $(\text{Hank}, \text{Denise})$ because Hank is a parent of Denise. Here is the complete relation P:

$$\begin{aligned} P = \{&(\text{Hank}, \text{Denise}), (\text{Hank}, \text{Ben}), (\text{Irene}, \text{Denise}), (\text{Irene}, \text{Ben}), \\ &(\text{Earl}, \text{Frank}), (\text{Earl}, \text{Gary}), (\text{Denise}, \text{Frank}), (\text{Denise}, \text{Gary}), \\ &(\text{Ben}, \text{Carl}), (\text{Alice}, \text{Carl})\}. \end{aligned}$$

How do we know that P is a relation on M? Easy! P is a subset of the Cartesian Product $M \times M$ — that's all we need for P to be a binary relation. There are many other subsets of $M \times M$ that are also binary relations,[1] but none of them represent the family tree of Figure 8.1 using (parent,child) ordered pairs.

When elements of domains and codomains are paired in a relation, we say that they are *related* within the context of the relation.

related

Definition 54: Related

When L is a relation and $(x, y) \in L$, x is said to be *related* to y and is denoted "$x \; L \; y$".

If you think that this "$x \; L \; y$" notation looks strange now, wait until you see it in an example!

Example 152:

In Example 151, Hank is the parent of Denise. One way to represent this fact is by using set membership notation: $(\text{Hank}, \text{Denise}) \in P$. This is perfectly fine, as P is a set and $(\text{Hank}, \text{Denise})$ is an element of P. Another way is to use the notation introduced by the definition of related: Hank P Denise. (Told you it would look strange!)

The "$x \; L \; y$" notation looks a little less odd with numbers.

Example 153:

Let $R = \{(x, y) \mid x \,\%\, y = 2\}$, where $x, y \in \{2, 4, 6, 8, 10\}$. For which pairs of values is $x \; R \; y$ true?

We need values x and y where $y > 2$ and x is two more than a multiple of y. (When $y = 2$, two more than a multiple of y is another multiple of y, and so $x \,\%\, y = 0$.) For the given domain, $6 \; R \; 4$, $10 \; R \; 4$, $8 \; R \; 6$, and $10 \; R \; 8$ are all true, and so $R = \{(6, 4), (10, 4), (8, 6), (10, 8)\}$.

[1] How many others? The total number of possible ordered pairs formed from our set M of family members is $|M \times M| = |M| \cdot |M| = 9 \cdot 9 = 81$. Next, think back to power sets, which is where we learned that there are $2^{|S|}$ possible subsets of a set S. Here, there are 2^{81} possible subsets of $M \times M$, and thus 2^{81} possible relations, of which P is just one. So, there are $2^{81} - 1 = 2,417,851,639,229,258,349,412,351$ other relations.

Let's consider a slight variation on Example 153:

Example 154:

Let $R = \{(x, y) \mid x \,\%\, y = 1\}$, where $x, y \in \{2, 4, 6, 8, 10\}$. For which pairs of values is $x\ R\ y$ true?

None of them! Our domain has only multiples of two, meaning that we can produce only remainders that are multiples of two,[2] and so a remainder of one is impossible. Thus, $R = \varnothing$.

To take the example a step further: When $R = \varnothing$, is R still a relation? Yes! The empty set is a subset of any set, which means that R is a subset of the Cartesian Product of $\{2, 4, 6, 8, 10\}$ with itself, meeting the conditions of the definition of 'relation.'

This book will not use the "$x\ L\ y$" notation very often, but other authors do use it, which is why you need to be familiar with it.

8.2 Creating Relations from Relations

We have seen that the idea of a relation is rather straight–forward, and that we can use set–builder notation to describe the Cartesian Product subsets that represent our situations. Another approach to the creation of relations is to build new relations from existing ones, rather than defining them directly from one or more base sets.

There are a variety of ways to build relations from other relations. One way is through the application of set operators such as union, intersection, and difference. Relations are sets, after all. But are the results of these operators actually relations on the same base set(s)? Are there other ways to build relations from existing relations? And, why would we want to build relations in these ways? This section will provide some answers to these questions.

[2] Yes, zero is a multiple of two. The product of any integer with x produces a multiple of x. $0 \cdot 2 = 0$, so zero is a multiple of two.

8.2.1 Relations from Relations using Set Operators

If you don't remember how the set operators union ($\cup$), intersection ($\cap$), difference ($-$), and complement ($\overline{\square}$) operate, we suggest you review the set sub–section of Appendix A (Math Review) before reading further.

Example 155:

Consider the relation $C = D \times D$, where $D = \{4, 5, 6\}$. That is, $C = \{(4, 4), (4, 5), (4, 6), (5, 4), (5, 5), (5, 6), (6, 4), (6, 5), (6, 6)\}$. Let:

$EQ = \{(x, y) \mid x = y\} = \{(4, 4), (5, 5), (6, 6)\}$, and
$LT = \{(x, y) \mid x < y\} = \{(4, 5), (4, 6), (5, 6)\}$,

where $x, y \in D$ for both. (Yes, normally we use single–letter set identifiers, but EQ for 'equals' and LT for 'less than' are more meaningful names.)

Questions: Is $EQ \cup LT$ a relation on D? If so, what does it represent?

Yes, the union of two sets on D must also be a relation on D. Here's why. All that the union of two relations does is collect the ordered pairs of each into a new third relation (ignoring any duplicates, of course). The ordered pairs are not changed in this process. As a consequence, the resulting relation's content must be a collection of ordered pairs on D, and thus, as a subset of $D \times D$, a relation on D.

As for what $EQ \cup LT$ represents, we hope you have already figured it out. If we collect together all of the pairs representing 'equal to' and those representing 'less than,' the result is the set of ordered pairs representing 'less than or equal to.' We'll label it as LE:

$LE = \{(x, y) \mid x \leq y\} = \{(4, 4), (4, 5), (4, 6), (5, 5), (5, 6), (6, 6)\}$

because we'll use it in the next example.

The reasoning for why the union of two relations on a set is also a relation on the same set holds for intersection and difference, too: We're forming relations from ordered pairs provided by the given relations. Example 156 demonstrates the utility of those two operators.

Example 156:

We'll continue where Example 155 stopped. Let's define one more relation, GE:

$GE = \{(x, y) \mid x \geq y\} = \{(4, 4), (5, 4), (5, 5), (6, 4), (6, 5), (6, 6)\}$

What do the relations $LE \cap GE$ and $LE - GE$ represent?

Intersection retains the ordered pairs that are in both of the given relations. The common concept between $\leq$ and $\geq$ is equality, which is why $LE \cap GE = \{(4, 4), (5, 5), (6, 6)\} = EQ$.

We can think of $LE - GE$ as starting with the content of LE, from which we 'remove' the ordered pairs that are also found in GE. That is, we remove the (x, x) pairs. Removing the pairs that represent equality from LE leaves only those that represent 'less than': $LE - GE = LT$.

Set complement needs a little more justification, because it doesn't directly re–use the content of the input relations as union, intersection, and difference do. Recall that the complement of a set S is the set of all of the elements of the universe that are not in S. When S is a relation, $\overline{S}$ is all of the ordered pairs from the universe (for our running example, the universe is C, the Cartesian Product) that are not in S. Stated in terms of our example: $\overline{S} = C - S = (D \times D) - S$. This expression shows that, because complement is defined in terms of set difference, we are actually still re–using ordered pairs; we just had to dig a little bit to uncover that fact.

Example 157:

Recall that $EQ = \{(4, 4), (5, 5), (6, 6)\}$ (from Example 155). What is $\overline{EQ}$?

EQ contains all of the ordered pairs representing equality. If we remove equality from the universe, what remains is inequality: $\overline{EQ} = \mathcal{U} - EQ =$

$\{(4,5),(4,6),(5,4),(5,6),(6,4),(6,5)\}$.

There's one more set operator that we haven't discussed in the context of creating relations from relations: Cartesian Product. We left it for the end for a good reason: The Cartesian Product of two binary relations is not another binary relation; it's a *4–ary* , or *tetradic*, relation. That is, its content isn't ordered pairs, it's ordered quadruples. Because the result isn't a binary relation, we have no use for it . . . at the moment. The idea of computing Cartesian Products of relations has definite value. We'll revisit this idea in Section 8.2.4.

4–ary relation

8.2.2 Relations from Relations using Inversion

Return with us now to those thrilling days of [3] . . . Chapter 1, when we learned that the converse of the implication $p \to q$ is the implication $q \to p$ — we simply exchanged the positions of the antecedent and the consequent. Applying this idea of exchanging positions to the elements within the ordered pairs of a relation creates the *converse relation*, also known as the *inverse relation*.[4]

inverse relation

Definition 55: Inverse Relation

The inverse (a.k.a. converse) of a relation R from set A to set B is the relation $R^{-1} = \{(b,a) \mid (a,b) \in R\}$ from B to A, where $a \in A$ and $b \in B$.

Conversationally, R^{-1} has R's ordered pairs, but the content of each pair is in the opposite order.

[3] . . . yesteryear! That was part of the usual opening to "The Lone Ranger" radio program.

[4] Given what we know of the terms 'inverse' and 'converse' in logic, this naming reuse is unfortunate, but only a little. In logic, inverse and converse are different adjustments to $p \to q$, but the inverse ($\neg p \to \neg q$) and converse ($q \to p$) are logically equivalent, giving some support to the idea that the terms 'inverse relation' and 'converse relation' can have the same meaning. As if two names for one idea isn't enough, there are others, including 'reciprocal,' 'opposite,' 'dual,' and 'transpose.' We'll stick with 'inverse.' As for other notations . . . don't ask.

Example 158:

Example 151 defined the parent relation, P, that matched the content of the family tree in Figure 8.1. What does the inverse relation P^{-1} represent?

Recall that:

$$\begin{aligned} P = \{&(\text{Hank}, \text{Denise}), (\text{Hank}, \text{Ben}), (\text{Irene}, \text{Denise}), (\text{Irene}, \text{Ben}), \\ &(\text{Earl}, \text{Frank}), (\text{Earl}, \text{Gary}), (\text{Denise}, \text{Frank}), (\text{Denise}, \text{Gary}), \\ &(\text{Ben}, \text{Carl}), (\text{Alice}, \text{Carl})\} \end{aligned}$$

on the set $M = \{\text{Alice}, \text{Ben}, \text{Carl}, \text{Denise}, \text{Earl}, \text{Frank}, \text{Gary}, \text{Hank}, \text{Irene}\}$. Swapping the elements of the ordered pairs creates a relation that defines the child relation $C = \{(x, y) \mid x \text{ is a child of } y\}$ on M. That is:

$$\begin{aligned} P^{-1} = C = \{&(\text{Denise}, \text{Hank}), (\text{Ben}, \text{Hank}), (\text{Denise}, \text{Irene}), (\text{Ben}, \text{Irene}), \\ &(\text{Frank}, \text{Earl}), (\text{Gary}, \text{Earl}), (\text{Frank}, \text{Denise}), (\text{Gary}, \text{Denise}), \\ &(\text{Carl}, \text{Ben}), (\text{Carl}, \text{Alice})\} \end{aligned}$$

In Example 158, the relation is on a set. Having the domain and codomain be the same set is necessary for R^{-1} to also be a relation on the same base set, but is not a requirement for the result to be called an inverse relation.

8.2.3 Relations from Relations using Relational Composition

The definition of relational composition is a bit mind–numbing. So, rather than start this third way to build relations from relations with the definition, we'll start with an example. Don't worry; we'll bore you with the definition soon!

Example 159:

In Miss Othmar's class, she records the traditional A–B–C–D–F grades of her students, but highlights them in her grade book with different colors. This makes for a two–step grading process: Record a letter–grade for each student, and highlight each student's name with the letter–grade's color. Each step has a corresponding relation.

The first step, that of recording the grade for each student, creates the relation G. Here is the content of G, for a subset of four of Miss Othmar's students:

G = {(Amos,D),(Jane,B),(Levi,D),(Ruth,A)}

For the colors, she uses a red–yellow–blue diverging palette, which we will label with K to avoid confusion with the grade 'C':

K = {(A,blue),(B,sky),(C,yellow),(D,orange),(F,red)}

('Sky' is a light blue color.) When she is done, Miss Othmar looks at the highlighted names and realizes that her two–step process has created a new relation M from a merging of G and K that pairs student names with colors:

M = {(Amos,orange),(Jane,sky),(Levi,orange),(Ruth,blue)}

A diagram can help make sense of the connections between the ordered pairs in G and K. Consider Figure 8.2, starting with the (Amos,D) pair, and note the arrow pointing at Amos. Follow that arrow as it leaves (Amos,D) from the D and enters the (D,orange) pair at its D. Finally, it leaves from 'orange' (as does another arrow). This sequence of arrows shows the connection that forms the ordered pair (Amos,orange) in the new relation. All of the other ordered pair matches have their own sequences of arrows. Note that reusing ordered pairs, as this example does with (D,orange), is fine.

The 'merging' of two relations to create a third, as demonstrated in Example 159, is called *relational composition*. Besides needing two existing relations, we need the 'to' set of one relation to be the same as the 'from' set of the other. This enables an overlapping of the ordered pairs that makes the composition possible. In Example 159, the shared set is the set of letter grades.

Now that you have seen a practical example of relational composition, the definition should be significantly less mind–numbing than it would have otherwise been. Even so, you'll still want to read it carefully!

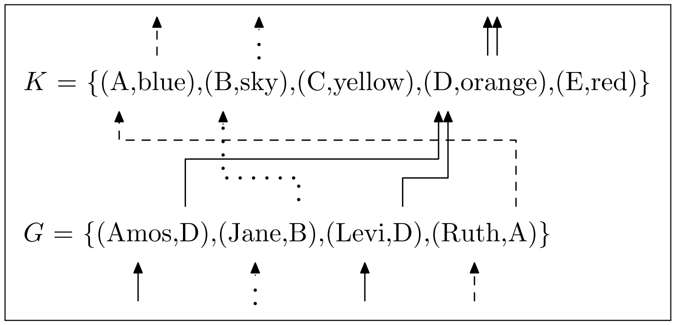


Figure 8.2: Relational Composition: Names to Grades to Colors.

relational composition

Definition 56: Relational Composition

Let G be a relation from set A to set B, and let F be a relation from the same set B to set C. The relational composition of relations F and G, denoted $F \circ G$ (LaTeX: `\circ`)[5], is the relation of ordered pairs (a, c) such that $(a, b) \in G$ and $(b, c) \in F$, where $a \in A, b \in B$, and $c \in C$.

Example 160:

At Watson–Deer Elementary School's Mid–Morning Recess games, the playground monitors award gold, silver, bronze, and pewter medals (well, certificates; they're cheaper) for first through fourth places, respectively.[6] In this month's jumprope endurance challenge, first place went to Lua, second was shared by Eve and Abe after an unfortunate collision, and fourth went to Kai. How can we create a relation that pairs the 'medals' with the students?

If you haven't already finished sarcastically muttering "Duh, using relational composition, maybe?", we're disappointed. Let's do it. We have two relations, M for the 'medals' and P for the places:

M = {(gold,first),(silver,second),(bronze,third),(pewter,fourth)}

functional composition

[5] Does this notation look familiar? It's also used for *functional composition* , which you've probably encountered before. We will cover it in the next chapter, where we will compare it to relational composition.

$P = \{$(first,Lua),(second,Eve),(second,Abe),(fourth,Kai)$\}$

Next, we need to order the relations correctly. Is it $M \circ P$ or $P \circ M$? According to the definition, the shared element must be the second member of the ordered pairs of the second relation listed. (In the definition, G is listed second in "$F \circ G$," and G's ordered pairs have the shared values — the b's — second.) Here, the place names are shared by both relations, and they appear second in M's ordered pairs. Thus, M needs to be second in the notation, making the correct ordering $P \circ M$.

Now to create $P \circ M$'s ordered pairs. We take each of the ordered pairs of the second relation (M) in turn, matching them with the corresponding ordering pair(s) from P to form the (medal,student) pairs we want:

$P \circ M = \{$(gold,Lua),(silver,Eve),(silver,Abe),(pewter,Kai)$\}$.

Figure 8.3 shows the visualization. Note that in this example, we use (silver,second) twice, because we have two elements of P that start with 'second.' Also note that we do not have any arrows running through (bronze,third) at all, because there wasn't a third place finisher (due to the tie for second).

Are you curious as to why our diagrams are constructed to have the arrows pointing bottom–to–top instead of top–to–bottom? We did this to better match how people are likely to write the relations. Given a composition $F \circ G$, it's natural to write the content of F first and that of G second. Doing this means that the arrows will go bottom–up.

8.2.4 Other Means of Creating Relations from Relations

Time to pick up where Section 8.2.1 feared to tread: Using Cartesian Product on relations. In that section we mentioned that the result of the Cartesian Product of two binary relations creates a 4–ary relation. While such a relation isn't useful in a discussion of binary relations, it is more generally useful. Here's how.

[6] "Pewter?" Yes, pewter! It's an alloy of tin, copper, and other metals. The U.S. Figure Skating Championships started awarding pewter medals to fourth place finishers in 1959. It goes without saying that the idea hasn't caught on with other sports, but it works well with our example.

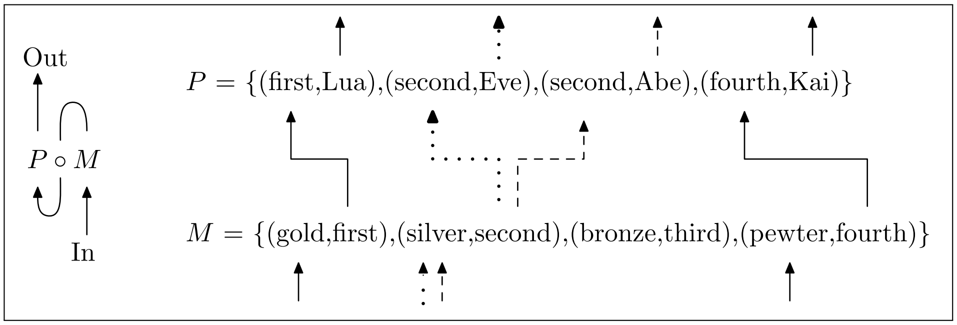


Figure 8.3: Relational Composition: Medals to Places to Students.

dbms

relational model

When people and organizations need to store a lot of information about an endeavor, such a business, they are likely to use a software product known as a *DBMS* (short for Database Management System) to both store the information and provide convenient access to it. Many DBMSes organize the information entrusted to them using the *relational model* , which is so named because the stored information is arranged into relations. Relations are good at storing basic types of information, such as numbers and words, but not so good with more complex types, such as songs and movies. Many current relational DBMSes have been augmented to support such types, but other, non–relational, types of DBMSes exist, too.

Example 161 demonstrates the utility of Cartesian Product to a relational DBMS.

Example 161:

Professor Plum has two relations that describe his small graduate seminar. The first, the class roster, has the ID, name, and email of each of the three registered students. The second, a seating chart, pairs each student with a chair in the room. As sets, we would write them like this:

$R = \{(21,\text{Boddy},\text{core}),(52,\text{Peacock},\text{peahen}),(68,\text{Mustard},\text{mustum})\}$
$S = \{(\text{A05},52),(\text{B19},21)\}$

. . . but when viewed as relations in a DBMS, they would be structured as tables with meaningful names and labels:

Roster

SID	Name	Email
21	Boddy	core
52	Peacock	peahen
68	Mustard	mustum

Seating

Chair	SID
A05	52
B19	21

Apparently, Mustard was absent the day the professor made the seating chart, but the DBMS won't be bothered by this.

The professor would like to confirm with each student their chosen seat, but the seat info is in one relation and the email addresses are in another. Cartesian Product to the rescue! Using that operator, the DBMS can pair up all of the roster information with all of the seating information in a 5–ary relation:

Roster $\times$ Seating

SID_1	Name	Email	Chair	SID_2
21	Boddy	core	A05	52
21	**Boddy**	**core**	**B19**	**21**
52	**Peacock**	**peahen**	**A05**	**52**
52	Peacock	peahen	B19	21
68	Mustard	mustum	A05	52
68	Mustard	mustum	B19	21

The result contains all possible associations of the ordered triples from Roster and the ordered pairs from Seating. The table rows that the professor needs, shown in bold in the table, are the two in which the SIDs are equal (because they are the associations that combine the information belonging to the same students).

Example 161 stopped before Professor Plum was satisfied — how is the professor to get just the subset of rows with equal SIDs from the Cartesian Product? And how can Mustard be identified as needing to receive a "see me" email to get a seat? The basic group of set operators does not include operators for these purposes, but relational DBMSes provide them. The *select operator* can answer the first question. The combination of Cartesian Product with *select operator*

join operator
outer join

select is so commonly used in relational DBMSes that another operator, the *join operator* , was created to perform both actions with a single command. A variation known as *outer join* can be used to help answer the second question.

Curious to know more about relational DBMSes, their connections to sets, and what else they can do? Database classes are commonly offered by Computer Science departments and business schools. Talk with an academic advisor . . . today!

8.3 Graph Representations of Relations

So far, our only way to represent a relation is as a set of ordered pairs. That fits the definition of 'relation' neatly, but it isn't always a convenient representation to use when we need to identify characteristics of relations, which we will need to do soon.

A representation of relations that is far more visual is one based on a data structure known as a *graph*. You may have already learned a few things about graphs; if so, this sub–section will be (mostly?) review. If not, don't worry; although graphs is a large topic, we only need a small subset for our purposes.

8.3.1 Graphs

The connection between graphs and relations is made clear by the definition.

graph

Definition 57: Graph

A graph $G = (V, E)$, where V is a finite, non–empty set of vertices and E is a binary relation on V.

By this definition, a graph isn't so much a way to visualize relations as it is a slight variation on the definition of binary relations. So much for graphs being deep and mysterious![7]

edge

Nothing in our definition of graphs tells us what a graph looks like. Easily rectified: Think of a vertex as a dot on a piece of paper. An ordered pair is a pair of dots with an arrow (in graph terminology, an *edge*) drawn from the left member of the pair to the right member of the pair. When we use

[7]Yeah, so, OK: *Basic* graphs aren't deep and mysterious. Dig a little deeper, and they can get pretty complex. Happily, basic graphs are all that we need in this chapter.

an arrow to indicate ordering of a pair of vertices, the graph is known as a *directed graph*, or *digraph* for short. *digraph*

You might be wondering: May a vertex have an edge that points back to itself? Yes! This type of edge is known as a *self–loop* . We will soon see that self–loops are very helpful for recognizing one of our upcoming relation properties. *self–loop*

You might also be wondering: The definition explicitly says that the relation is on V. Does this mean that "from — to" relations cannot have graph representations? No! We can handle such relations with a simple adjustment: Allow V to be the union of the relation's 'from' and 'to' sets.

This is all we need to know about graphs to create diagrams that visualize relations!

8.3.2 Using Graphs to Represent Relations

In the near future (the next section!), we will use graphs to help us understand a variety of properties of relations. For the graphs to be most useful, they need more than vertices and edges — they also need clarity, and, well, a touch of aesthetics. Aesthetics is the study and appreciation of beauty, and is applicable here because the 'attractiveness' of our graphical representations of relations will add to their clarity.[8] Rather than trying to explain, we will demonstrate with a couple of examples.

Example 162:

In Example 153, our base set was $\{2, 4, 6, 8, 10\}$. When creating a graph of an "on" relation, the base set is also our collection of vertices, meaning that the visualization of the graph has five dots. As a set, the relation R consists of four ordered pairs: $\{(6, 4), (10, 4), (8, 6), (10, 8)\}$. Thus, our visualization also has four edges (arrows).

Figure 8.4 shows three different graphs, all of which represent R. The first version, 8.4(a), has two sets of vertices, one for the domain and one for the codomain. This isn't a good representation for this example, because relation R is an "on" relation, not a "from one set, to a second

[8] "Mathematics, rightly viewed, possesses not only truth, but supreme beauty." – Bertrand Russell. The standard for our figures in this book is less lofty: "Well, my eyes ain't bleedin'!"

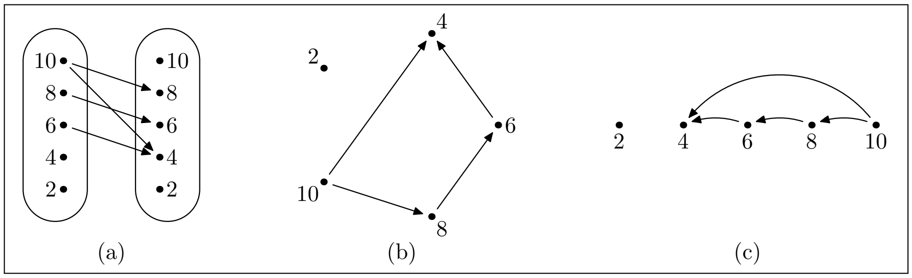


Figure 8.4: Three Graphs of the Relation of Example 153

set” relation. By representing the single base set $\{2, 4, 6, 8, 10\}$ of R as two separate groups, we are improperly implying that the domain and codomain sets are different sets.

Figure 8.4(b) has just one set of vertices, arranged in a circle. This addresses our concern about Figure 8.4(a), and also ‘de–clutters’ the visualization. Figure 8.4(c) opts for a linear arrangement of the vertices, providing a more compact graph.

For relation R, either (b) or (c) is an acceptable graph; which one is ‘better’ is mostly a matter of taste.

Example 163:

In the English alphabet, the distinction between consonants and vowels is based on whether or not the human vocal tract is restricted when producing the sound.

Of the first five letters of the English (a.k.a. Basic Latin) alphabet, ‘A’ and ‘E’ are vowels; the rest (‘B’, ‘C’, and ‘D’) are consonants. The set S of the (letter,sound–type) pairs is a relation from the set of the first five letters to the set of speech sound types: $S = \{$(A,vowel),(B,consonant), (C,consonant),(D,consonant),(E,vowel)$\}$.

We could just scatter the seven vertices (the five letters plus ‘consonant’ and ‘vowel’) and start drawing arrows, but, as Figure 8.5(a) shows, the result can be disorganized.

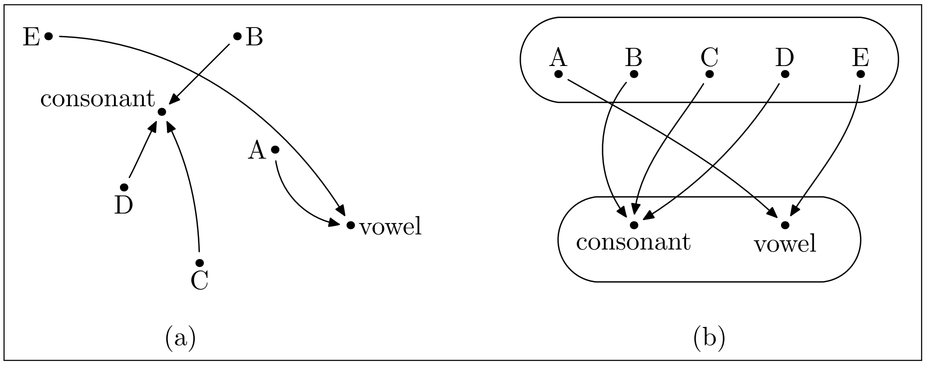


Figure 8.5: Two Graphs of the Relation of Example 163.

When the domain and codomain are separate sets, in the interest of clarity, we will group the domain and codomain elements separately. Doing so will make it easier for us to answer basic questions at a glance, such as "Do all of the letters have an associated sound type?", and "How many of the letters are vowels?". One such graph is shown in Figure 8.5(b).

Example 164:

The family tree diagram (Figure 8.1) can be re–drawn as a digraph, as shown in Figure 8.6. Typically, family trees do not include explicit arrows, because the orderings are well–understood. Including the extra lines and arrowheads only adds clutter, not clarity.

8.4 Four Properties of Binary Relations

Fair warning: This section will present a few concepts that are likely to be confusing and maybe even headache–inducing, at which point you may wonder, "Why do I have to learn this???". Hopefully, at the same time you ask that question, you will remember this paragraph. Individually, most of these properties may not appear to be helpful. Together, though, they are used to define additional properties that have practical applications in computer science. Please stay focused; your patience will be rewarded.

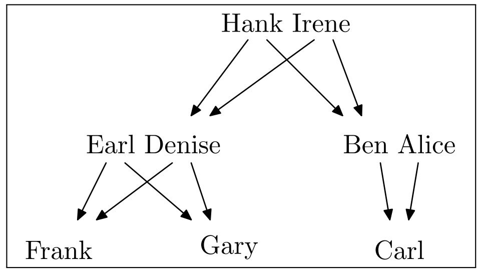


Figure 8.6: The family tree of Figure 8.1 re–drawn as a digraph.

Something else to know before we dive in: All four of the following properties apply only to relations defined "on" a set. This means, for example, that there is little value in asking if relation S from Example 163 possesses any of these properties, because the answer will always be 'no.'

8.4.1 Reflexivity

Reflexivity is the easiest of the four, both to define and to understand, making it a good property to examine first. This will not be a short sub–section, though, as we will take advantage of reflexivity's relative simplicity to make several points that have value for the following properties, too.

In English, 'reflexive' is an adjective used to describe something that is pointed back toward its own beginning. For example, a song about songs is reflexive. Another: A dog chasing its own tail is playing reflexively. With those examples in mind, the definition of reflexivity as it applies to relations is easy to understand and remember.

Reflexive Relations

In conversational English, when a relation is reflexive, each member of the base set is related to itself. More formally:

reflexivity

Definition 58: Reflexivity

A (binary) relation R on a set A is *reflexive* if $(a, a) \in R$, $\forall\, a \in A$.

Example 165:

Consider the relation $N = \{(x, y) \mid x \text{ and } y \text{ have the same first name}\}$ on the set $M = \{\text{Michael Douglas}\}$. Is N even a relation on M? If it is, is it a reflexive relation on M?

Let's start by showing all of the ordered pairs in N — this won't take long:

$$N = \{(\text{Michael Douglas, Michael Douglas})\}$$

For N to be a relation on M, it must be a subset of $M \times M$. As defined above, $N = M \times M$. Because every set is a subset of itself, yes, N is a subset of $M \times M$, and so N is a relation on M.

Now for reflexivity: For N to be a reflexive relation on M, it must pass the test supplied by the definition of reflexivity. The definition says that N must contain the ordered pair (x, x) for each $x \in M$. Because M contains only one name, the ordered pair that N must contain to be reflexive is (Michael Douglas, Michael Douglas), and it does. Thus, N is reflexive.

Example 165 was about as small an example of a reflexive relation as we can create, but we can go smaller. Even better, going smaller is worth our time.

Example 166:

Somewhere around the end of the movie "The Ant-Man and the Wasp," Ant-Man's support team, well, takes a powder.[9] We can represent members of the non–existent support team with the set $T = \varnothing$. Is the Cartesian Product $T \times T$ a reflexive relation?

Was your first thought a quick 'no'? We don't blame you for thinking that, but the answer is actually 'yes.' $T \times T$ is a relation on T. It's also true that $T \times T = \varnothing$, making it the empty relation. (Remember, any set is a subset of itself, and the empty set *is* a set.)

How can the empty relation be reflexive? Vacuously! The definition is conditional: A (binary) relation R is reflexive **IF** $(a, a) \in R$, $\forall a \in A$. For every element t of T, we must have the ordered pair (t, t) in the reflexive relation. There are no elements in T, so there can be no ordered pairs. Thus, there are no ordered pairs that must appear in the relation in order for it to be reflexive. This is why our empty relation vacuously satisfies the definition.

We defined vacuous truth in Chapter 1 and explained vacuous proof in Chapter 4, but this was our first opportunity to use the concept. More are coming!

Example 167:

Two–letter words are really ordered pairs. For example, "to" is an English word, but "ot" is not — order matters.

Let W be a relation on the set $L = \{a, e, h\}$ such that $(x, y) \in W$ when xy (the concatenation of x and y) is an English word. According to the 6th edition of "The Official Scrabble Players Dictionary," six two–letter words can be formed from the letters in L: 'aa', 'ae', 'ah', 'eh', 'ha', and 'he'. Thus, $W = \{(a, a), (a, e), (a, h), (e, h), (h, a), (h, e)\}$. Is W reflexive?

$(a, a) \in W$, but $(e, e) \notin W$, and so W is not reflexive. There is no reason to check (h, h) — as soon as we know that one reflexive ordered pair is missing, we know that the relation cannot be reflexive.

Recognizing Reflexivity in Graphs of Relations

Checking for reflexivity in a set of ordered pairs isn't very difficult, so long as the base set of the relation isn't too large. Checking for reflexivity in a graph is even easier.

Figure 8.7 contains graphs of Examples 165 and 167. Take a few moments to examine them, keeping in mind that the first is reflexive and the second is not. The relevant difference is found in the self–loops: Reflexive relations

[9]The phase "to take a powder" is slang for leaving in a hurry. If you don't already see the connection to the movie, go watch it . . . after you've also watched a pile of prequels!

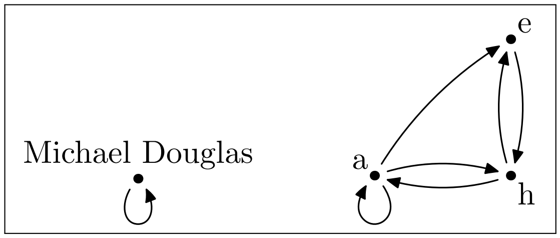


Figure 8.7: Graphs of the Relations of Examples 165 (left) and 167.

have self–loops on all vertices. Or, relations that are not reflexive are missing at least one self–loop.

The self-loops, at least how we have drawn them, look a bit like earrings. This mental image is a nice way to remember reflexivity in graphs. People usually feel that a missing earring is bad news. Similarly, a missing earring is bad news for reflexivity.

8.4.2 Symmetry

You may be familiar with the word *symmetry* in the context of bilateral (a.k.a. reflection) symmetry or radial (a.k.a. rotational) symmetry in biology. When discussing relations, symmetry has a very different meaning.

Symmetric Relations

symmetry

Definition 59: Symmetry

A (binary) relation S on set B is *symmetric* when $(b, c) \in S$ iff $(c, b) \in S$.

In English, this definition tells us that when an ordered pair (x, y) is in a symmetric relation, the ordered pair (y, x) is also in the relation.

Appropriately, relational symmetry is defined in terms of a logical operator (biimplication) whose truth table column is horizontally symmetric: The top half of the column $\binom{\mathrm{T}}{\mathrm{F}}$ is a mirror–image of the bottom half $\binom{\mathrm{F}}{\mathrm{T}}$.

Examples are just as helpful with symmetry as they are with reflexivity.

Example 168:

Vivian, Willa, and Ximena have registered as a team for an intramural 3–on–3 basketball tournament. Let $T = \{(x, y) \mid x \text{ and } y \text{ are teammates}\}$ on the set of players $P = \{\text{Vivian}, \text{Willa}, \text{Ximena}\}$. Is T symmetric?

To be symmetric, $(x, y) \in T$ if and only if $(y, x) \in T$. It is clear that that logical statement holds in this example, because any two people in the set P are teammates, regardless of the order in which we list them. For instance, if $(\text{Vivian}, \text{Willa}) \in T$ then $(\text{Willa}, \text{Vivian}) \in T$, and if $(\text{Willa}, \text{Vivian}) \in T$ then $(\text{Vivian}, \text{Willa}) \in T$. This is the case for any pair of teammates drawn from P. Therefore, yes, T is symmetric.

As long as this example is fresh, let's ask (and answer) a review question about it: Is T reflexive?

The answer depends on the definition of "teammate." Is a person his or her own teammate? We hope you'll agree that the answer is 'no.' That is, $(\text{Vivian}, \text{Vivian}) \notin T$, which is enough to prove that T is not reflexive.

Example 169:

We used a two–letter word example (Example 167) to help explain reflexivity. Recall that the relation is $W = \{(a, a), (a, e), (a, h), (e, h), (h, a), (h, e)\}$ on the set $W = \{a, e, h\}$. Is W symmetric?

We can answer this via inspection of the ordered pairs in W. All that we need to do is consider each ordered pair $(x, y) \in W$ and verify that $(y, x) \in W$. If that is always true, W is symmetric.

Let's start with (a, a). If $(a, a) \in W$, then it's clear that the ordered pair with the content in reverse order (that is, (a, a)) will also be in the relation, because it's the same ordered pair! This means that we can ignore all such ordered pairs (that is, the reflexive ordered pairs) when testing for symmetry.

The next ordered pair is (a, e). The reversed pair, (e, a), is not in W. This means that "if $(a, e) \in W$ then $(e, a) \in W$" is false, which also means that "$(a, e) \in W$ iff $(e, a) \in W$" is false. Because the iff is always true in a symmetric relation, we know that W is not symmetric, without having to test any more pairs of ordered pairs.

Note that W exhibits symmetry with the pairs (a, h) and (h, a), as well as with the pairs (e, h) and (h, e). That's two of the three that we need for symmetry, but that's not good enough – we need three out of three.

Example 170:

Example 165 introduced the relation $N = \{$(Michael Douglas, Michael Douglas)$\}$ on the set $\{$(Michael Douglas$\}$. N is reflexive, but is it symmetric?

Let's conduct the same test that we used in Example 169: For each ordered pair $(x, y) \in N$, is $(y, x) \in N$? If that's always true, then N is symmetric. N only has one ordered pair, making this test easy to perform. (Michael Douglas, Michael Douglas) $\in N$, so we need the reversed ordered pair, (Michael Douglas, Michael Douglas), to also be in N. Because it is the same ordered pair, yes, it is in N. As that is the only ordered pair, and that ordered pair passed the test, then yes, N is symmetric.

Here's another way to see that N is symmetric. N's only ordered pair is a reflexive ordered pair — N has no ordered pairs of the form (x, y) where $x \neq y$. Without at least one such ordered pair, we cannot violate the definition of symmetry. If a relation cannot violate the definition, the relation must satisfy the definition. By that reasoning, N is (vacuously) symmetric.

Recognizing Symmetry in Graphs of Relations

Detecting reflexivity in the graph of a relation was pretty easy. Happily, detecting symmetry in graphs is almost as easy ... if we draw aesthetically pleasing graphs!

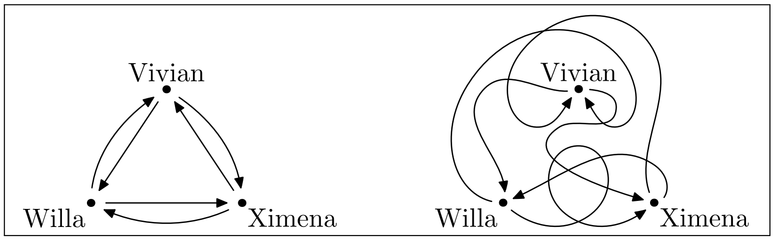


Figure 8.8: Two Graphs of the Relation of Example 168.

When a relation is symmetric, its graph will have only "back and forth" edges. That is, when a symmetric relation has an edge from vertex v to vertex w, it will also have an edge from w back to v. In a clearly–drawn graph, a pair of vertices with only one of the two edges will be easily identified. Note that self–loops are inherently "back and forth" all by themselves, so we can ignore them whether they exist or not.

Example 171:

Figure 8.8 shows two versions of graphs of the 3–on–3 basketball team example (Example 168). The graph on the left shows the pairs of edges plainly, making the symmetry of the relation easy to recognize. The graph on the right . . . doesn't. Remember, good visualizations make the important details clear.

8.4.3 Antisymmetry

This is the most important detail to know about antisymmetry:

Antisymmetry is **not** the opposite of symmetry!

Puzzled? We sympathize. In English, *anti* is a prefix meaning "opposite," which leads people to believe that an *anti*symmetric relation is one that is *not* symmetric. As we will see, there is a lot of middle ground between a symmetric relation and an antisymmetric relation, but we will also see that some relations are both symmetric and antisymmetric.

Can you handle a little more confusion? There's another property of relations called *asymmetry* that is also not the opposite of symmetry, even though the prefix *a* means "not."[10] Good news! We won't be defining asymmetry for a simple reason: This book doesn't need it.

Antisymmetric Relations

If antisymmetry isn't the opposite of symmetry, then what does it mean? Thanks for asking!

Definition 60: Antisymmetry

antisymmetry

A (binary) relation A on set D is antisymmetric if $(d, e) \in A$ and $d \neq e$, then $(e, d) \notin A$, $\forall d, e \in D$.

Conversationally: In an antisymmetric relation, distinct elements are paired together only once. As with symmetry, the self–loop (a.k.a. reflexive) edges don't matter to antisymmetry; they can exist or not.

Example 172:

Drew and Faustina keep track of the relative ages of their offspring with the relation $O = \{(x, y) \mid x \text{ is older than } y\}$ on their set of children $C = \{\text{Poseidon}, \text{Zoe}, \text{Nikita}\}$. Assuming that Poseidon is 7, newborn Zoe is 0, and future child Nikita is -2 (what can we say; Drew and Faustina have a plan), is the relation O antisymmetric?

As a set of ordered pairs, $O = \{(\text{Poseidon}, \text{Zoe}), (\text{Poseidon}, \text{Nikita}),$ $(\text{Zoe}, \text{Nikita})\}$. For a relation to be antisymmetric, it cannot contain an ordered pair (y, x) if it already contains the ordered pair (x, y), when x and y are distinct. Here, that means O cannot have $(\text{Zoe}, \text{Poseidon})$, cannot have $(\text{Nikita}, \text{Poseidon})$, and also cannot have $(\text{Nikita}, \text{Zoe})$. O contains none of these pairs, thus O is antisymmetric.

[10] Makes you wonder what people were ~~dr~~thinking when they named these properties, doesn't it? They have to be related to whomever decided that *flammable* and *inflammable* should have the same meaning. (Before you write us: We already know why they have the same meaning. Save your word–origin wisdom for someone on Reddit who really needs it.)

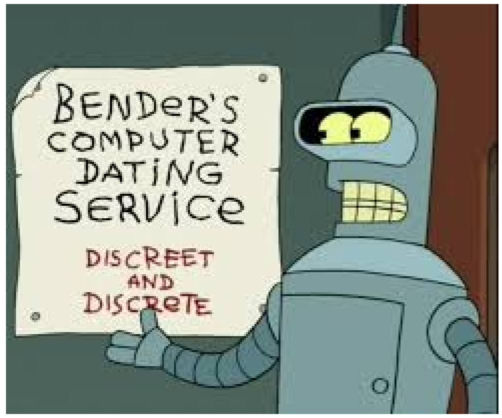


Figure 8.9: How Drew and Faustina met. Credit: "Futurama" episode 2ACV07 ("Put Your Head on My Shoulders").

Example 173:

In Example 167 and Example 169, we learned that the two–letter word relation is not reflexive and also not symmetric. Is it or is it not antisymmetric?

Recall that $W = \{(a,a),(a,e),(a,h),(e,h),(h,a),(h,e)\}$ on $L = \{a,e,h\}$. We can test W an ordered pair at a time, checking each for the 'matching' (reversed) ordered pair that would destroy W's dream of antisymmetry. (a,a) is a self–loop, which the definition of antisymmetry tells us to ignore. The next pair is (a,e). If $(e,a) \in W$, we know W is not antisymmetric and can stop. But, $(e,a) \notin W$, so we continue. Next is (a,h). This time the reversed ordered pair (h,a) is in W. Because a and h are paired together twice in the relation, W is not antisymmetric.

Combined, Examples 169 and 173 demonstrate that relations can easily be both not symmetric and not antisymmetric. We mentioned at the start of this sub–section that a relation can be both symmetric and antisymmetric. The next example demonstrates that we've already seen one such relation.

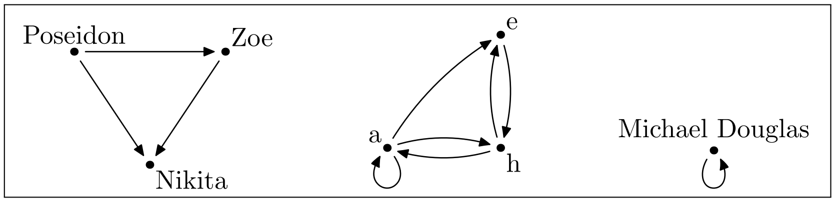


Figure 8.10: Graphs of the Relations of Examples 172, 173, and 174.

Example 174:

Back in Example 165, we introduced the relation $N = \{$(Michael Douglas, Michael Douglas)$\}$ on the set $M = \{$Michael Douglas$\}$ and showed that it is reflexive. In Example 170, we showed that it was also symmetric, albeit vacuously. Is N also antisymmetric?

We already spoiled the answer to that question: Yes, N is also antisymmetric. Why? For the same reason that N is symmetric: N doesn't violate the definition of antisymmetry, which, like the definition of symmetry, is conditional. The key part of the definition: "[...] if $(d, e) \in A$ and $d \neq e$, then $(e, d) \notin A$ [...]" Relation N does not contain any such (d, e) ordered pairs, meaning that the antecedent of the 'if' is false, which by definition makes the conditional expression true. No ordered pairs in N violate the definition of antisymmetry, and so, vacuously, N is antisymmetric.

Recognizing Antisymmetry in Graphs of Relations

Antisymmetry is just as easily detected in a well–drawn graph as is symmetry. The difference is that, instead of making sure we have only "back and forth" pairs of edges, we need to make certain that there are no "back and forth" edges. That is, we want to see only single edges between pairs of vertices.

Figure 8.10 contains the graphs of all three of the relations used in this sub–section. As with reflexivity and symmetry, only a quick glance is required to verify the presence or absence of antisymmetry. The left graph clearly has no "back and forth" arrows (and so its relation is antisymmetric), the center graph has two such pairs of edges (either of which violates the definition of antisymmetry), and the right graph obviously has no worrisome edges (meaning that its relation is antisymmetric because the relation does not even try to violate the definition of antisymmetry).

8.4.4 Transitivity

The bad news: While our fourth relation property, transitivity, is easy to explain, it is harder to detect within relations than are the other three properties. The good news: Transitivity is also the property that you are most likely to have seen previously. For example, if I tell you that $a = b$ and that $b = c$, you know that a and c must also be equal. That's transitivity of equality. What you probably did not know before reading this sentence is that even the transitivity of equality is based on ... you guessed it, relations.[11]

Transitive Relations

As promised, transitivity's definition is straight–forward.

transitivity

Definition 61: Transitivity

A relation T on set F is transitive if $(f, g) \in T$ and $(g, h) \in T$, then $(f, h) \in T$, $\forall f, g, h \in F$.

Example 175:

Consider the stack of blocks in Figure 8.11, and the concept of "above," as in blocks P and I are both above block L.

Let relation $A = \{(b, c) \mid b \text{ is above } c \text{ in Figure 8.11}\}$ on the set of blocks $B = \{\mathrm{E}, \mathrm{I}, \mathrm{L}, \mathrm{P}\}$. Is A transitive?

To answer this question, having A's set of ordered pairs will help. In alphabetical order, $A = \{(\mathrm{I}, \mathrm{E}), (\mathrm{I}, \mathrm{L}), (\mathrm{L}, \mathrm{E}), (\mathrm{P}, \mathrm{E}), (\mathrm{P}, \mathrm{I}), (\mathrm{P}, \mathrm{L})\}$.

The definition of transitivity tells that we only need to worry about pairs

[11] Transitivity in grammars is distinct from transitivity in relations. A transitive verb requires a direct object to be grammatically correct (e.g., *She aced the class.*), while an intransitive verb does not (*He slept.*).

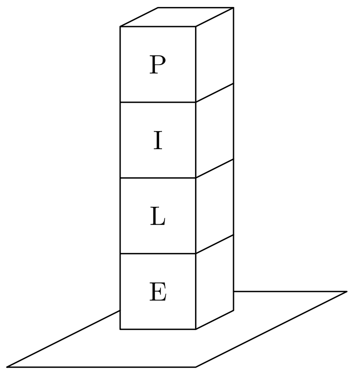


Figure 8.11: Example 175's stack of blocks.

of ordered pairs that share the 'overlapping' value (e.g., g in (f,g) and (g,h)). In a transitive relation, the pairs of the form (f,h) must also be in the relation. Our task is to identify all of the $(f,g)-(g,h)$ pairs in the relation and check that the pair (f,h) is also in the relation.

Helpful advice: **Be systematic!** If you randomly look around for overlapping pairs, you are likely to both find some pairs twice and skip some pairs entirely. The former is just wasted work, but the latter could cause you to report an incorrect answer.

Here's our system: Moving left to right, consider each ordered pair in the relation as our leading (f,g) pair. Scan (also left to right) through the list of ordered pairs, finding all that start with g, and create all of the $(f,g)-(g,h)$ pairs. For each such pair of pairs, search for the pair (f,h) in the relation. If an (f,h) is not in the relation, we can stop, because the relation cannot be transitive when an (f,h) ordered pair is missing. But if all of the (f,h) ordered pairs are found, the relation is transitive.

Making a table to keep track of all of the pairs of pairs is worthwhile. The first (leftmost) ordered pair in A is (I, E), so we start our table's first row with it:

(f,g)	(g,h)	(f,h)	Present?
(I, E)			

Now we check A for ordered pairs that have E as the left value. In our diagram, the E block is not above any other blocks. Thus, there can be no ordered pairs in A of the form $(\mathrm{E}, \square)$. If you're hoping that this means we're done …sorry. Remember, the definition is conditional: **IF** (f, g) and (g, h) are in T, then so must (f, h). Having just (f, g) doesn't allow that condition to be tested, and an untested condition cannot fail. We must continue our search.

The next ordered pair in A is (I, L). This time, scanning A reveals an overlapping (g, h) pair: (L, E). For A to be transitive, it must also contain the (f, h) pair, which is (I, E). A quick search reveals that, yes, $(\mathrm{I}, \mathrm{E}) \in A$. (Or we can remember that we just examined it in the first row …) This allows us to complete the second row of the table:

(f, g)	(g, h)	(f, h)	Present?
(I, E)	—	—	—
(I, L)	(L, E)	(I, E)	Yes!

Got the idea? We hope so, because if we continue at this pace, this example will require many more pages to finish. Let's skip ahead to this state of our table:

(f, g)	(g, h)	(f, h)	Present?
(I, E)	—	—	—
(I, L)	(L, E)	(I, E)	Yes!
(L, E)	—	—	—
(P, E)	—	—	—
(P, I)	(I, E)	(P, E)	Yes!

We're pausing here to make this point: We found one $(f, g) - (g, h)$ pair that begins with (P, I), but there is another! Block I is above both blocks E and L; we need to check both of their corresponding ordered pairs. The final table:

(f,g)	(g,h)	(f,h)	Present?
(I, E)	—	—	—
(I, L)	(L, E)	(I, E)	Yes!
(L, E)	—	—	—
(P, E)	—	—	—
(P, I)	(I, E)	(P, E)	Yes!
(P, I)	(I, L)	(P, L)	Yes!
(P, L)	(L, E)	(P, E)	Yes!

Thanks to systematism,[12] we have found all four pairs of overlapping ordered pairs, and found that A contains all of the necessary (f,h) ordered pairs that need to accompany them. A is transitive!

Example 175 required more work than did any of our previous relation property examples. As we warned, checking transitivity is straight–forward but often requires more effort than does checking the three relation properties we've already covered.

Example 176:

Time for another visit to the two–letter word example (Example 167). The relation, as a set of ordered pairs, is $W = \{(a,a),(a,e),(a,h),(e,h),(h,a),(h,e)\}$ on the set $W = \{a,e,h\}$. Is W transitive?

Using the table approach from Example 175, we begin with the ordered pair (a,a), which is a reflexive ordered pair. We don't need to spend any time finding $(f,g)-(g,h)$ matches for such ordered pairs; here's why. Whether a reflexive ordered pair is the (f,g) pair or the (g,h) pair, the other pair will also be the (f,h) pair. For example, $(a,a)-(a,e)$ needs (a,e) to be in the relation, and $(1,2)-(2,2)$ would need $(1,2)$. In both cases, the relation needs an ordered pair that it obviously already possesses. We can't totally ignore such pairs, though; stay tuned.

In this example, (a,a) is the only ordered pair that doesn't need a row in our table, but every little bit helps. So, we start our table with (a,e):

[12]It is so a word! *Systematism* is a noun meaning "following a method or system." On this we can agree: It's an uninspiring superpower.

Figure 8.12: “Sati! Come here, darling. Leave the poor man in peace.” “But, Papa, he’s Keanu Reeves, not Michael Douglas!” “Even more reason . . . ” Credit: Warner Bros, “The Matrix Revolutions.”

(f, g)	(g, h)	(f, h)	Present?
(a, e)	(e, h)	(a, h)	Yes!
(a, h)	(h, a)	(a, a)	Yes!

This is why we couldn’t totally ignore (a, a): We needed it to test the definition’s condition. Onward:

(f, g)	(g, h)	(f, h)	Present?
(a, e)	(e, h)	(a, h)	Yes!
(a, h)	(h, a)	(a, a)	Yes!
(a, h)	(h, e)	(a, e)	Yes!
(e, h)	(h, a)	(e, a)	No!

W was looking good for transitivity for a while, but we soon discovered a missing (f, h) ordered pair. One is all that we need to show that W is not transitive. (If you complete the table, you’ll find that there are more missing ordered pairs.)

We bet that you can guess what our last transitivity example is going to be.

Example 177:

Ah, Michael Douglas; we can’t seem to leave the poor man in peace.

Example 165's relation is $N = \{(\text{Michael Douglas}, \text{Michael Douglas})\}$ on $\{(\text{Michael Douglas})\}$. We've explained that N is reflexive, symmetric, and antisymmetric. Is it also transitive?

Having seen our explanations for symmetry and antisymmetry on this example, you're probably expecting the answer to be, "Yes, vacuously!" We'll give partial credit for that answer, for being both right and wrong. Read on!

As we covered in Example 175, we need a $(f, g) - (g, h)$ pairing of ordered pairs to define a (f, h) ordered pair that needs to be missing for the relation to be declared 'not transitive.' N technically does have such a pairing: $(\text{Michael Douglas}, \text{Michael Douglas}) - (\text{Michael Douglas}, \text{Michael Douglas})$. We already know that pairings using reflexive ordered pairs cannot cause the definition to be violated. Still, if we were to complete the table anyway (and abbreviate in the interest of space) ...

(f, g)	(g, h)	(f, h)	Present?
(M. D., M. D.)	(M. D., M. D.)	(M. D., M. D.)	Yes!

... we would confirm that N does not violate the definition, and so is transitive. But, because we were actually able to test the condition using N's ordered pair, saying that N's satisfaction of the definition is vacuous is incorrect.

Recognizing Transitivity in Graphs of Relations

As you read the transitivity examples, you were probably wishing that we would hurry up and get to this sub–section, where the secret to visualizing transitivity would be revealed. Sorry to say, there is no such secret.

We'll get straight to the point: Graphs are not nearly as helpful for visualizing transitivity as they are for visualizing reflexivity, symmetry, and antisymmetry. For very simple relations, yes, a graph can clearly show transitivity. But, even for graphs with just a handful of edges, identifying transitivity can be tricky.

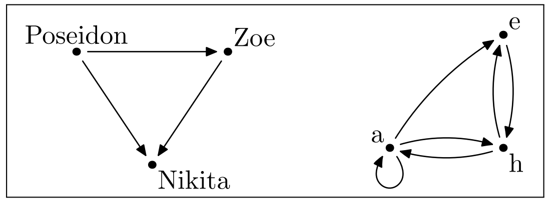


Figure 8.13: Repeats the Graphs of the Relations of Examples 172 and 167.

Example 178:

Example 172 introduced Drew and Faustina's family and the relation O that contains ordered pairs of their children by age. The graph of O is reproduced in Figure 8.13 (left figure). We can tell at a glance that O is transitive. No need to exhaustively examine a list of ordered pairs this time!

Example 179:

The graph of our two–letter word example (the right figure in Figure 8.13) contains just six edges, yet it is not easy to see that the relation the graph depicts is not transitive. Yes, the fact that we would need the (e, a) edge to accompany the edges (e, h) and (h, a) is fairly clear. But how clear is it that we also need (h, h) and (e, e)? Sure, we could sit down and work through all of the pairs of edges, but that's no different than working through all of the pairs of ordered pairs; remember, the edges are just representations of the ordered pairs.

The take–away message: The systematic approach is still the approach to take to verify that a relation is or is not transitive. Well ... unless there's another relation representation that is more helpful?

8.5 Matrix Representations of Relations

We've just seen (last section!) that digraph representations of graphs are a nice way to visualize all of our four graph properties except transitivity. Two–dimensional matrices (Chapter 7) are another useful representation for relations, with a couple of new benefits. One, nearly all programming languages have matrix operation libraries available. This means that, if we can figure out how to identify relation properties within a matrix representation, we can have a computer do the dirty work of property–checking for us. Two, we can detect transitivity within matrices using matrix multiplication. True, multiplying matrices is a pain — but not if we politely ask a computer to do it for us!

There is a small catch: While we can use matrices to represent any binary relation, matrix representations are most useful for 'on' relations. Happily, our four relation properties require 'on' relations.

8.5.1 Representing Binary Relations Using Two–Dimensional Matrices

Old news: Binary relations are sets of ordered pairs. The first element of each ordered pair is found in the 'from' base set, while the second element is found in the 'to' base set. To create a corresponding 2–D matrix, we line up all of the elements of the 'from' set, and use them to label the rows of the matrix. Separately, we do the same with the elements of the 'to' set, and use them to label the columns.

The content of the matrix M representing the relation R consists of the values 0 and 1. Element m_{rc} is set to 1 if and only if $(r, c) \in R$, and element m_{rc} is set to 0 if and only if $(r, c) \notin R$.

Example 180:

Consider again Example 163, which presented a 'from–to' relation example of letters and two sound types (consonants and vowels). The 'from' base set was {A,B,C,D,E}, the 'to' base set was {consonant,vowel}, and the relation was $S = \{$(A,vowel),(B,consonant),(C,consonant),(D,consonant), (E,vowel)$\}$. Using those orderings of the elements in the 'from' and 'to' sets, our matrix representation's content is:

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{cc} \textit{consonant} & \textit{vowel} \end{array} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array}$$

The matrix representation of S is quite useful. Of course, we can easily see that, for example, (A,vowel) $\in S$ and that (A,consonant) is not, but we can do more. If we add up the numbers in each row, we learn that all of the sums are one, meaning that all of the letters must be vowels or consonants, but not both. The sums of the columns tell us that there are three consonants and two vowels.

Relation S in Example 180 uses different base sets for its domain and codomain, making it what we've been calling a 'from–to' relation rather than an 'on' relation. Our relation properties (reflexivity, etc.) can be used only with 'on' relations. One implication of this restriction is that our matrices will be square. Another is that we will label our rows and columns with the same set of values. We need to follow an important rule when doing that labeling: *List the values in the same order for both the row labeling and the column labeling!* You can choose any ordering of the base set values that you like; you just have to like it enough to use it for both row and column labeling. Use that ordering left to right on the columns, and top–down on the rows.

8.5.2 Recognizing Reflexivity in Matrices of Relations

A quick review of Definition 58: Relation R on a set A is reflexive if $(a, a) \in R,\ \forall\, a \in A$. In order for a matrix representation to be useful in helping us (or a computer) recognize reflexivity, the collection of ones in the matrix that represent the (a, a) ordered pairs need to stand out relative to all of the other ones in the matrix. Happily, they do.

Example 181:

Back in Example 155, we constructed the relation $LE = \{(4, 4), (4, 5), (4, 6), (5, 5), (5, 6), (6, 6)\}$ on $D = \{4, 5, 6\}$, but haven't yet used it as an example for any of our relation properties. Time to change that! Is LE reflexive?

Here are two possible matrix representations of LE:

$$M_1 = \begin{array}{c c} & \begin{array}{ccc} 4 & 5 & 6 \end{array} \\ \begin{array}{c} 4 \\ 5 \\ 6 \end{array} & \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \qquad M_2 = \begin{array}{c c} & \begin{array}{ccc} 6 & 4 & 5 \end{array} \\ \begin{array}{c} 6 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \end{array}$$

$M_1 \neq M_2$ because we used different label orderings for each. But notice where the (a, a) ordered pairs are found, regardless of the ordering: Down the main diagonal (upper–left to lower–right) of the matrix. In LE, all three of the reflexive ordered pairs are present, which is why all three of the main diagonal elements are ones. These matrices demonstrate that when the main diagonal of the matrix representation is all ones, the relation is reflexive.

A small detail of Example 181: The elements not on the main diagonal are irrelevant as far as reflexivity is concerned.

8.5.3 Recognizing Symmetry in Matrices of Relations

In Chapter 7, we learned the definition of matrix symmetry. Informally: When we swap the columns and rows of a square matrix, if the original and resulting matrices are equal, the matrix is symmetric.

Swapping the rows and columns means swapping the row and column indices of the matrix elements. That is, element m_{rc} in the original matrix will become element m_{cr} in the new matrix, and vice–versa. Push this fact on your mental stack; we'll come back to in very soon.

Back to relations. A relation is symmetric when every ordered pair (s, t) in the relation is accompanied by its reversed ordered pair (t, s). In matrix form, elements m_{st} and m_{ts} are both one.

Time to pop your stack, put these ideas side–by–side, and ask the big question: If the matrix representation of a relation is a symmetric matrix, must that relation be a symmetric relation? Sounds like it's time for . . . a proof![13]

[13] We apologize profusely for the number of pages that have passed between proofs. We will work hard to increase the proof–to–page ratio going forward because . . . OK, so no one asked for it. But we'll try anyway.

Example 182:

Problem: Prove or disprove: If the matrix representation M of relation R is a symmetric matrix, then R is a symmetric relation.

Solution: Thanks to our definitions of matrix and relation symmetry, the proof is straight–forward. We need to show the connection between matrix and relation symmetry.

Proof (Direct): Recall that, by the definition of a symmetric matrix, all of the m_{rc} – m_{cr} pairs must have matching values.

Assume that M is the matrix representation of a relation R on a base set S, and assume that M is symmetric. Consider the elements m_{st} and m_{ts} of M, where $s, t \in S$. Because M is symmetric, $m_{st} = m_{ts} = 1$ or $m_{st} = m_{ts} = 0$. If the former, both of the ordered pairs (s, t) and (t, s) are elements of R, satisfying the definition of a symmetric relation. If the latter, neither ordered pair is in R, vacuously satisfying the definition.

Therefore, if the matrix representation M of relation R is a symmetric matrix, then R is a symmetric relation.

The converse is also true, making this an "if and only if" theorem.

Because relation symmetry corresponds to matrix symmetry so nicely, all of our knowledge of symmetry from matrices can be applied to relations to determine their symmetry, as the next example demonstrates.

Example 183:

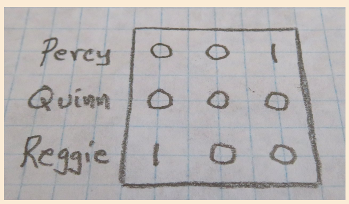
Brothers Percival, Quinn, and Reginald are being quiet . . . too quiet for their mother's taste.

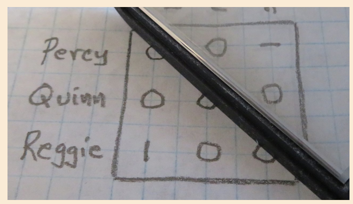
Mom: "Boys, what are you doing?"
Reggie: "I'm playing Global Thermonuclear War with Percy."
Percy: "And I'm playing Global Thermonuclear War with Reggie."

Quinn: “I’m reading ‘Slaughterhouse-Five’ again.”
Mom: “What lovely symmetry!”

Is their mother right? Are the boys’ current interactions symmetric?

We can check by (a) inspecting ordered pairs in the interaction relation, (b) drawing a graph of the relation and looking for the “back and forth” edges, or (c) creating the relation’s matrix representation and perching a mirror on the main diagonal. Let’s risk seven years’ bad luck and go with (c)!





The content of the upper–right corner of the matrix, as shown in the left picture, matches that of the reflected mirror–universe version, demonstrating that the interaction relation is symmetric. As usual, Mom’s right.

A thought: Should we have (Quinn,Quinn) in the relation? Probably not, unless Quinn reads aloud to himself. But, as far as symmetry is concerned, either is fine — remember that in symmetry, the reflexive ordered pairs don’t matter.

8.5.4 Recognizing Antisymmetry in Matrices of Relations

If you understood how to recognize symmetry in a matrix representation of a relation, and you haven’t already forgotten what antisymmetry is, you probably already know how to detect antisymmetry: None of the corresponding lower–left and upper–right values can match.

Example 184:

Question: Is the ‘PILE’ relation from Example 175 antisymmetric?

Answer: Seems like it ought to be; how can two cubes be above each

Figure 8.14: Logically, the mirror universe matrix should have a sinister mustache and goatee. Credit: Star Trek, "Mirror, Mirror," Season 2, Episode 4.

other? But let's be sure. The relation in that example was named A, so let's call our matrix representation M_A:

$$M_A = \begin{array}{c} \\ E \\ I \\ L \\ P \end{array} \begin{array}{c} \begin{array}{cccc} E & I & L & P \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

We're going to make you imagine the mirror this time. There are six corresponding pairs of values above and below the main diagonal to check, and they are all different. For antisymmetry, this is good; A is antisymmetric.

8.5.5 Recognizing Transitivity in Matrices of Relations

We promised that representing relations with matrices would solve our transitivity recognition problem, and it will ... if you're a computer, or can somehow multiply two matrices rapidly in your head. No matter who or what does the math, you should understand why matrix multiplication can be used to check for transitivity in a relation.

Before we get to 'why it works,' let's cover the 'how to do it:' Start by creating a matrix representation M_R of the relation R. Next, multiply M_R by itself via matrix multiplication. If the same set of elements that were zero in M_R are still zero in M_R^2, then R is transitive. Otherwise, R is not transitive.

Example 185:

Question: Is $R = \{(\mathrm{A,A}),(\mathrm{A,B}),(\mathrm{B,B}),(\mathrm{C,A}),(\mathrm{C,C})\}$ a transitive relation?

Answer: R is not transitive. We can verify this using the exhaustive "if (f, g) and (g, h), then (f, h)" pairs of ordered pairs approach we demonstrated in Example 175. Specifically, R is not transitive because it contains (C,A) and (A,B), but not (C,B).

Of course, this example exists to show how to get that answer using M_R, which is:

$$M_R = \begin{array}{c c} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \end{array}$$

Note that there are four elements that are zero in M_R. Now take a look at M_R^2:

$$M_R^2 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{array}{c c} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 2 & \mathbf{1} & 1 \end{bmatrix} \end{array}$$

M_R^2 has only three elements that are zero. The one in red, the one that is no longer zero, is — you guessed it — (C,B)'s.

But ... why is the element of the missing ordered pair no longer zero? And, why *only* that element? To reach the answers to those questions, we will ask another question: Why is the ordered pair (C,B) needed to make R transitive? Phrased another way: What is sufficient to make (C,B) necessary for transitivity?

You might remember that we answered that question just ahead of this example: Having both (C,A) and (A,B) in R is sufficient to make (C,B) necessary for transitivity to hold. But other pairs of ordered pairs are also sufficient: (C,B) and (B,B), as well as (C,C) and (C,B). All three fit the $(f, g) - (g, h)$ pattern that defines transitivity.

Another question before we can answer the original questions: How does matrix multiplication 'find' those potentially useful pairs of ordered pairs? Exhaustively! Consider how the '1' at row C and column B came to be. We evaluated this expression (we've dropped the relation label for clarity):

$$\begin{aligned} m^2_{CB} &= m_{CA} \cdot m_{AB} + m_{CB} \cdot m_{BB} + m_{CC} \cdot m_{CB} \\ &= 1 \cdot 1 + 0 \cdot 1 + 1 \cdot 0 \\ &= 1 + 0 + 0 \\ &= 1 \end{aligned}$$

(This expression is just the expansion of the general summation that defines matrix multiplication. If you don't remember that summation, hop back to Section 7.3.3 for an explanation.)

There are a couple of things to notice about this expression, within the context of transitivity. First, the subscripts of the elements in the products in the first line correspond exactly to the collection of sufficient pairs of ordered pairs that we identified earlier. In effect, the products are testing whether or not the three $(f, g) - (g, h)$ pairs exist in R. If one does, its product is one; otherwise, it is zero. Second, should more than one pair of the ordered pairs exist in R, the sum will be greater than one. This is why we talk about zero and non–zero elements rather than zero and one. (More on this later!)

With those additional questions answered, we can return to the original questions (finally!). We now know why m_{CB} is no longer zero. Only m_{CB} became non–zero because, of all of the potentially 'sufficient' pairs of ordered pairs in R, only the (C,A) — (A,B) pair of ordered pairs exists. Each zero in M^2_R represents a potentially 'necessary' ordered pair that turned out to not be necessary, due to the lack 'sufficient' pairs of ordered pairs in R. The matrix multiplication, simply because of the way matrix multiplication is defined, conveniently tested all of them for us and reported the 'nothing to see here' results as zeros.

Bonus information: You probably noticed that two values of two appear in M_R^2. You might be wondering if there's a difference between a value of one and a value of two. There is! The value represents the number of 'sufficients' that exist in R to make that ordered pair necessary. For example, consider m_{AB}. There are three possible pairs of ordered pairs that are sufficient to make (A,B) necessary for transitivity, and R has two of them: (A,A) — (A,B) and (A,B) — (B,B).

Continuing that thought: M_R always consists of just zeros and ones, meaning that we can view it as a logical (i.e., $(0,1)$) matrix. You might see where this observation is leading, and the answer is: Yes, we can use logical matrix product in place of matrix multiplication to test for transitivity. Of course, unlike $M_R \cdot M_R$, the result of $M_R \odot M_R$ will contain only zeros and ones.

Example 186:

We already know that relation A from Example 175 (the 'PILE' example) is transitive. Let's verify it using a logical matrix product.

From Example 184, we know the content of M_A. We just need to compute $M_A^{[2]}$:

$$M_A^{[2]} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \odot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \begin{array}{c} \\ E \\ I \\ L \\ P \end{array} \begin{array}{c} \begin{array}{cccc} E & I & L & P \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

M_A has 10 zeros. Those same elements are all still zero in $M_A^{[2]}$, verifying that A is transitive.

Do the 'extra' zeros in $M_A^{[2]}$ bother you? Don't let them. $M_A^{[2]}$ is not an accumulation of values of both M_A and $M_A \odot M_A$. Instead, think of it as a collection of all of the (f, h) pairs produced by the exhaustive $(f, g) - (g, h)$ testing for transitivity. Some of the ordered pairs in A ((I,L), for instance) aren't required to be in A by any $(f, g) - (g, h)$ pairs of ordered pairs, and so they aren't represented in $M_A^{[2]}$. $M_A^{[2]}$ tells us what must be present in A for it to be transitive, and (I,L) isn't needed. That's why extra zeros aren't a problem, but missing zeros are.

8.6 Equivalence Relations

Feeling a little burnt out on relation properties? Sorry, but we have a few more to go. There is a potential bright side to that news: The remaining properties are (mostly) defined in terms of properties that we already know. All you need to do is remember which of the old properties are needed by each of the new ones. More good news: We'll have a system to help you remember them!

8.6.1 Motivating Equivalence Relations

Do you know an object–oriented (OO) programming language, such as Java, C++, or Python? Such languages are used by programmers to create software systems in which objects encapsulate the data describing, and operations that manipulate, the concepts upon which such systems are built. For example, if we wanted to create a program that plays Klondike solitaire, we would likely represent each of the 52 playing cards with an object, and each of the seven piles of the tableau with seven additional objects.

To ensure that the human player follows the rules of the game, our program would need to check that cards are played legally. In Klondike solitaire, when a black–suited eight is atop a pile, only a red–suited seven can be played upon it. To perform this check, the program needs to be able to compare two card objects to verify that the new card is one less than the top card, and that the colors of the suits are different.

To help programmers test objects against each other, the language Java allows programmers to mark them as being *comparable*. In Java 12's documentation for comparability[14] are these words:

[14] Specifically, this is in the Java 12 API for the `Comparable` interface. See: https://docs.oracle.com/en/java/javase/12/docs/api/java.base/java/lang/Comparable.html

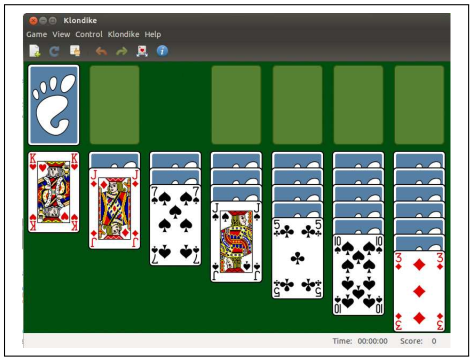


Figure 8.15: Gnome AisleRiot's Klondike Solitaire interface.

For the mathematically inclined,[15] the *relation* that defines the natural ordering on a given class C is:

{`(x, y) such that x.compareTo(y) <= 0`}.

The *quotient* for this total order is:

{`(x, y) such that x.compareTo(y) == 0`}.

It follows immediately from the contract for `compareTo` that the quotient is an *equivalence relation* on C, and that the natural ordering is a *total order* on C.

(In Java, the 'factories' that create objects are called *classes*.) Note that comparability is defined in terms of relations, and that the concept of a quotient is an example of an equivalence relation. Just another reminder that the material we're covering is useful in computer science.

If you're wondering what *quotients* and *total orders* are, keep reading. Quotients of relations will be explained soon, in Section 8.6.3, while total orders are the last of our relation properties and will be covered in Section 8.8.

[15] That's us! Just by reading this, you are henceforth "mathematically inclined." Congratulations! Knowledge of the secret handshake will be revealed in due time.

8.6.2 Equivalence Relations

After all of that setup, the actual definition is likely to be a bit anticlimactic:

equivalence relation

Definition 62: Equivalence Relation

A relation E on set A is an equivalence relation if E is reflexive, symmetric, and transitive.

That is the definition, but there is more to the story. Some implications of that combination of relation properties will be raised by the following examples. Going into them, you need to know that the first three letters of the Greek alphabet are *alpha* (upper case: A, lower case: α), *beta* (B, β), and *gamma* (Γ, γ).

Example 187:

Question: Let $S = \{$(x,y) $\mid x$ and y are the same letter, regardless of case$\}$ on the set $U = \{$A,B,$\Gamma\}$ (note that $U \neq \mathcal{U}$, the universe). Is S an equivalence relation?

Answer: Let's start with a little review. To be an equivalence relation on U, S must first be ...? Right: A relation on U. Represented as a set of ordered pairs, $S = \{$(A,A),(B,B),(Γ,Γ)$\}$. S is clearly a subset of $U \times U$, making it a relation on U.

To be an equivalence relation, S must be reflexive, symmetric, and transitive. Let's consider each in turn.

Reflexive: This is very easy to verify. The only three ordered pairs in S are the three reflexive ordered pairs that S needs to be a reflexive relation.

Symmetric: Do we have $(y, x) \in S$ for each $(x, y) \in S$? We clearly do, because in all of our ordered pairs, $x = y$. Or, we can remember that the reflexive ordered pairs can be ignored for symmetry, leaving us with no other ordered pairs, making the relation vacuously symmetric.

Transitive: The only $(f, g) - (g, h)$ pairs of ordered pairs in S are those where $f = g = h$, and so all necessary (f, h) ordered pairs are present, too.

Yes, S is an equivalence relation.

It is not difficult to believe that a relation that only pairs three letters with themselves is an equivalence relation; A=A, B=B, and $\Gamma = \Gamma$ all seem very equivalent. But what if we add the lower–case letters?

Example 188:

Question: We have S and U as in Example 187, let $L = \{\alpha, \beta, \gamma\}$, and change the base relation of S to $L \cup U$. Is S still an equivalence relation?

Answer: First, remember that S is defined to say that letters are the same regardless of case. With the new, larger base set, the content of S is also larger: $S = \{(A, A), (A, \alpha), (\alpha, A), (\alpha, \alpha), (B, B), (B, \beta), (\beta, B), (\beta, \beta), (\Gamma, \Gamma), (\Gamma, \gamma), (\gamma, \Gamma), (\gamma, \gamma)\}$ S is clearly still a relation on $L \cup U$, so let's get to the good stuff, starting with the matrix representation of S, M_S:

$$M_S = \begin{array}{c c} & \begin{array}{cccccc} A & \alpha & B & \beta & \Gamma & \gamma \end{array} \\ \begin{array}{c} A \\ \alpha \\ B \\ \beta \\ \Gamma \\ \gamma \end{array} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Reflexive: The main diagonal of M_S contains only ones, showing that S is reflexive.

Symmetric: Ignoring the reflexive ordered pairs, we only have the ordered pairs $(A, \alpha), (\alpha, A), (B, \beta), (\beta, B), (\Gamma, \gamma)$, and $(\gamma, \Gamma)\}$ to check. As every (x, y) has the corresponding (y, x) right next to it, S is easily verified to be symmetric.

Transitive: There is no joy like the joy of manually computing M_S^2

on a 6×6 matrix. Due to the quantity and location of the zeros in the matrix, computing M_S^2 really isn't very hard. The result:

$$M_S^2 = \begin{array}{c c} & \begin{array}{cccccc} A & \alpha & B & \beta & \Gamma & \gamma \end{array} \\ \begin{array}{c} A \\ \alpha \\ B \\ \beta \\ \Gamma \\ \gamma \end{array} & \begin{bmatrix} 2 & 2 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{bmatrix} \end{array}$$

All of the zeros of M_S are still zeros in M_S^2. S is transitive.

Thus, S on $L \cup U$ is also an equivalence relation.

Are you thinking, "OK, S satisfied the definition, but A and α are *not* the same letter! How can S be an equivalence relation when A and α are not equivalent?" Leave the definition aside for a moment, and think about how we used set builder notation to describe S: "x and y are the same letter, regardless of case." As case doesn't matter, A and α are equal, according to the instructions used to create S.

8.6.3 Equivalence Classes

You probably noticed that the non–zero values in the M_S matrix of Example 188 seem to be in three clusters along the main diagonal. Those clusters are prominent due to the way the base set values are ordered. Still, the groupings suggest that the alphas have a affinity for one another, as do the betas and the gammas. Those sets of associated elements inspires the concept of an *equivalence class*.

equivalence class

Definition 63: Equivalence Class

Let E be an equivalence relation on set B, with $b \in B$. b's *equivalence class*, denoted $[b]$, is the set $\{c \mid (b, c) \in E\}$, where $c \in B$.

In plain(er?) English, the equivalence class of b is the set of all of the values c that appear on the right of b in ordered pairs of E.

Example 189:

Question: What is $[\Gamma]$, using the equivalence relation S from Example 188?

Answer: Let's answer this using the set of ordered pairs representation of S, and then using M_S.

There are two ordered pairs in S that have Γ on the left side: (Γ, Γ) and (Γ, γ). All we have to do to create the equivalence class $[\Gamma]$ is build a set consisting of the two right–side values of those ordered pairs. $[\Gamma] = \{\Gamma, \gamma\}$.

Equivalence classes are even easier to see within matrix representations, because all of the ordered pairs that begin with a particular element will be non–zero values within that element's row of the matrix. The column labels of those non–zero values are the elements of the equivalence class of the row label. In M_S of Example 188, there are two non–zero values in Γ's row. Their column labels are Γ and γ. It follows that $[\Gamma] = \{\Gamma, \gamma\}$.

Remember that only equivalence relations have equivalence classes.

The set of all of an equivalence relation's equivalence classes is known as the *quotient* of the relation's base set.

Definition 64: Quotient

quotient

Again let E be an equivalence relation on set B. B's *quotient* with respect to E, denoted B/E, is the set of all of E's equivalence classes.

Together, all of the equivalence classes of a relation "divide up" (form a partition of) the relation's base set. Looked at this way, the term 'quotient,' and its division–like notation, makes some sense.

Example 190:

There are three equivalence classes of relation S on the base set $L \cup U$ in Example 188: $[A] = [\alpha] = \{A, \alpha\}$, $[B] = [\beta] = \{B, \beta\}$, and $[\Gamma] = [\gamma] = \{\Gamma, \gamma\}$. The quotient of $L \cup U$, $(L \cup U)/S$, is $\{\{A, \alpha\}, \{B, \beta\}, \{\Gamma, \gamma\}\}$.

8.7 Partial Orders

An activity we hope you experience at least twice a day: Teeth–brushing. Imagine that you are standing just outside of your bathroom. Your not–too–old toothbrush and your non–empty tube of toothpaste are both sitting on the counter. Before brushing can happen, you need to walk into the bathroom, you must grab hold of the toothbrush, you must grab the toothpaste tube, and you need to apply a blob of toothpaste to the brush. Here are two sequences of these actions that can result in your teeth being brushed with toothpaste:[16]

Sequence #1	Sequence #2
Walk in	Walk in
Grab toothbrush	Grab toothpaste
Grab toothpaste	Apply paste to brush
Apply paste to brush	Grab toothbrush
Brush teeth	Brush teeth

Some of these actions need to be performed in a specific order (for example, grabbing the toothpaste before applying it to the brush), while others can be performed in either order (such as, we can grab either the brush or the tube first) and our goal of clean(er) teeth can be achieved.

Figure 8.16 shows the immediate ordering relationships between the five actions, as a digraph. Note that, although entering the bathroom ("Walk In") is ordered before "Brush Teeth," there is no edge directly connecting them. This is because the ordering is transitively implied by the edges that already exist. However, there are no edges, direct or implied, between the actions of grabbing the toothpaste tube and grabbing the toothbrush. Thus, some pairs of actions are ordered and some are not. Such a relation is known as a *partially ordered relation*, or, more compactly, a *partial order*.

[16] Yeah, Sequence #2 assumes that your toothbrush is pretty stable on the countertop. Imagine you can afford nice, heavy, solid gold toothbrush handles ...

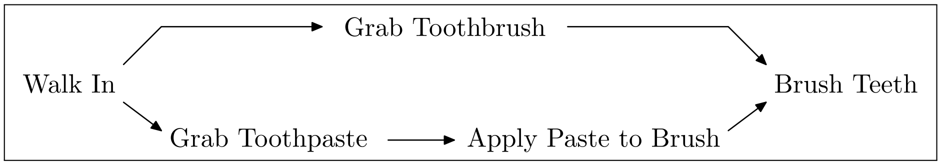


Figure 8.16: Orderings of the Teeth–brushing Actions

Just as we have two types of disjunction, we have two kinds of partial order. We'll define both because both are useful, but we'll do so one at a time.

8.7.1 Reflexive Partial Orders

The name "reflexive partial order" is a big clue as to one of the three relation properties that combine to define it. We dropped another clue above, when we commented on transitivity implying unshown edges in Figure 8.16. That leaves one relation property, which is antisymmetry. That's right: The only difference between an equivalence relation and a reflexive partial order is the "anti" in front of "symmetric."

reflexive partial order

Definition 65: Reflexive Partial Order

A relation R on set P is a reflexive partial order if R is reflexive, *antisymmetric*, and transitive.

An alternate name for this kind of partial order is "weak partial order." A third is "non–strict partial order," which hints at the name of our yet–to–come second partial order variety.

Our first reflexive/weak/non-strict partial order example is the classic: Less than or equal to.

Example 191:

Question: Let $L = \{(i, j) \mid i \leq j\}$ on the set $P = \{11, 13, 17, 19\}$. Is L a reflexive partial order?

Answer: To be a reflexive partial order, L must be reflexive, antisymmetric, and transitive.

Reflexive: As any integer is $\leq$ itself, L will contain (11,11), (13,13), etc., and is therefore reflexive.

Antisymmetric: Quick review: To be antisymmetric, a relation that contains (p, q) cannot also contain (q, p) when $p \neq q$. L has several such ordered pairs, including (11,13) and (13,17). But, the reversed pairs (13,11) and (17,13) cannot be in L because $13 \not\leq 11$ and $17 \not\leq 13$. The same is true of all other such pairs of elements of P, telling us that L is antisymmetric.

Transitive: Let's be exhaustive this time, but not *too* exhaustive. We'll make our lives easier by ignoring the (p, p) ordered pairs, as Example 176 taught us we can do. That leaves us with just four $(f, g) - (g, h)$ pairs of ordered pairs to check. As all needed (f, h) pairs are present, L is transitive:

(f, g)	(g, h)	(f, h)	Present?
(11,13)	(13,17)	(11,17)	Yes!
(11,13)	(13,19)	(11,19)	"[17]
(11,17)	(17,19)	(11,19)	"
(13,17)	(17,19)	(13,19)	"

Yes, L is a reflexive partial order.

A question before we leave this example: Is L also an equivalence relation? To check L for equivalence, we only need to consider symmetry, as we've already verified that L is reflexive and transitive. L is not symmetric, because, for example, we have $(11, 13) \in L$ but not $(13, 11)$. So, no, L is not also an equivalence relation.

Poset[18] is a term associated with partial orders. It is an abbreviation of

[17]This is a *ditto mark*, and indicates that the text on the previous line is repeated on this line. Ditto is also the name of a Pokémon and of one of the twins (repeated ... get it?) in the long–running "Hi and Lois" newspaper comic strip.

[18]To us, this sounds like the name of some near–future, gotta–have child's toy, like a Furby or a Tickle Me Elmo. "Mom! Dad! Get your kids a Poset this holiday season! It's adorable **and** relational!" (Fine print: Posets are both flammable and inflammable. Earplugs

the phrase "partially ordered set," and is the name of the base set of a reflexive partial order.

Definition 66: Poset

poset

A *poset* (partially ordered set) is the base set B of a reflexively partially ordered relation R, and is denoted by the ordered pair (B, R).

Example 192:

Question: Does relation L from Example 191 have a poset? If so, what is it?

Answer: That example showed that L is a reflexive partial order, which means that L does have a poset, and that poset is its base set, P.

Beyond the notation given in the poset definition, there's a bit more poset notation that you should understand, and learning it soon after a less–than–or–equal–to example is the best time.

Say that R is a reflexively partially ordered relation on B, with $x, y \in B$ and $(x, y) \in R$. When discussing reflexive partial orders, instead of writing "$(x, y) \in R$," people will shorten it to "$x \preceq y$" (LaTeX: `\preceq`). Notice that the symbol '$\preceq$' is very similar to the more common '$\leq$' symbol, but with more artistic flair. And, because '$\preceq$' represents the recursively partially ordered relation, people will often use the symbol '$\preceq$' to also be the relation's name, instead of using a letter, making the poset notation $(B, \preceq)$.

Still with us? We hope so, but there's more to the story. Instead of tracking down the correct symbol, people will just use '$\leq$' instead, resulting in the notation $(B, \leq)$ and leading people new to posets into thinking that posets are only for less–than–or–equal–to relations. Which raises a question: Why not pick a totally new symbol instead of one that looks like '$\leq$'? The reason is that partial orders based on '$\leq$' are the classic examples, which, unfortunately, caused '$\leq$' to be adopted as the symbol for all reflexive partial

recommended. Requires 13 AAAAA batteries, not included nor available in stores. Manufacturer not responsible for abuse of the good/evil switch.)

orders, whether or not they are based on that operator. People who use '$\preceq$' are both sticking with tradition while trying to be (slightly) distinct from '$\leq$'.

In summary: When you see the notation $(B, \preceq)$ or $(B, \leq)$, you are being told that B is a poset on a reflexive partial order named '$\preceq$' or '$\leq$', and should be aware that the partial order may or may not involve the less–than–or–equal–to operator.

Example 193:

Question: We began this section with two sequences of teeth–brushing operations. Is that example a reflexive partial order?

Answer: Figure 8.16 showed a graph representation of that example's relation. As a set of ordered pairs, it's $T = \{$(Walk In, Grab Toothbrush), (Walk In, Grab Toothpaste), (Grab Toothpaste, Apply Paste to Brush), (Grab Toothbrush, Brush Teeth), (Apply Paste to Brush, Brush Teeth)$\}$.

As the discussion with the figure made clear, the graph, and thus the relation T, does not include all of the edges necessary for T to be transitive. This means that T cannot be a reflexive partial order.

Even if we added all of the needed ordered pairs to achieve transitivity, T still wouldn't be reflexive, and so still would not be a reflexive partial order.

A follow–up question: Is the base set of T a poset? No. Remember that the 'po' stands for 'partially ordered.' For a base set to be a poset, it must be associated with a reflexive partial order, which T is not. The same base set could be a poset for a different relation, but not for T.

8.7.2 Irreflexive Partial Orders

If you hear a mathematician say that something is a "partial order," odds are that they mean it's a reflexive partial order. However, the teeth–brushing example (with the additional ordered pairs for transitivity) certainly feels as though we could call it a partial order. Insisting on reflexivity works fine for situations such as less–than–or–equal–to, but excludes other relations that we might practically know to be 'partial orders,' too.

This is why we also have *irreflexive partial orders*. Because we only use the concept of irreflexivity for this one kind of partial order, we didn't include the term alongside reflexivity, symmetry, and the rest. But, as we can't define irreflexive partial order without it, . . .

Definition 67: Irreflexivity

irreflexivity

A relation E on set B is *irreflexive* iff no element of B is related to itself in E.

Irreflexivity is at the opposite extreme of reflexivity: To be reflexive, a relation must have *all* (a, a) ordered pairs; to be irreflexive, a relation must have *no* (a, a) ordered pairs.

The parallels between reflexivity/irreflexivity and symmetry/antisymmetry are strong. Just as it is possible for a relation to be both symmetric and antisymmetric, it's possible for a relation to be both reflexive and irreflexive. Similarly, just as there are many relations that are neither symmetric nor antisymmetric, there are also many relations that are neither reflexive nor irreflexive. Finally, just as "antisymmetric" is not the same as "not symmetric," "irreflexive" is not the same as "not reflexive."

Now that that's out of the way, we can return to partial orders.

Definition 68: Irreflexive Partial Order

irreflexive partial order

A relation I on set B is an *irreflexive partial order* if it is *irreflexive*, antisymmetric, and transitive.

Just as we only added the prefix "anti" to the definition of equivalence relation to create the definition of reflexive partial order, we only added the prefix "ir" to the definition of reflexive partial order to get that of irreflexive partial order.

An alternate name for "irreflexive partial order" is "strict partial order." Many people prefer "weak" and "strict" to the longer "reflexive" and "irreflexive," but we like the full names because they perfectly describe the only difference between the two concepts.

Example 194:

Question: Is the teeth–brushing example's relation an irreflexive partial order?

Answer: No, at least not in its original form. Remember, relation T from Example 193 is not transitive.

This time, let's actually add the additional ordered pairs to T that the graph of Figure 8.16 did not include. We'll call the augmented relation T' (read: T–prime), and will abbreviate the action names to make the length more manageable.

The abbreviated original ordered pairs of T are (WI,GTB), (WI,GTP), (GTP,APB), (GTB,BT), and (APB,BT). Transitivity requires the additions of (WI,APB), (GTP,BT), and (WI,BT).[19] Combined, we have $T' = \{$(WI,GTB), (WI,GTP), (GTP,APB), (GTB,BT), (APB,BT), (WI,APB), (GTP,BT), (WI,BT)$\}$. For practice, feel free to verify the transitivity of T' yourself.

Having achieved transitivity, we need to check irreflexivity and antisymmetry, too. T' is irreflexive, because it contains no (a, a) pairs. T' is also antisymmetric because no $(d, e) - (e, d)$ pairs of ordered pairs exist. T' is therefore an irreflexive partial order.

8.7.3 Remembering the Definitions of Equivalence Relation, Reflexive Partial Order, and Irreflexive Partial Order

We promised to present a way for you to easily remember the definitions of equivalence relation and of the two partial orders. We aim to keep our promises!

First, all three definitions are built of three concepts. Second, all three include transitivity. Third, in the order in which we presented them, each definition is a slightly different version of the previous definition. If you can remember the equivalence relation definition and the two adjustments, you'll be able to reconstruct the partial order definitions when you need them.

transitive closure

[19]Bonus knowledge! Because T' is the smallest transitive relation such that $T \subseteq T'$, T' is known as the *transitive closure* of T.

Figure 8.17: Where's Anti Ir? Well, she was ... um ... indisposed.[22] Credit: Metro–Goldwyn–Mayer, "Wizard of Oz."

The equivalence relation definition can be remembered for consisting of only the three 'positive' relation properties: Reflexivity, Symmetry, and Transitivity. (We think of them as being 'positive' because their names do not include any of the English 'not' prefixes.)

Now for the two adjustments. You've probably seen the movie "The Wizard of Oz," or perhaps read the original book, "The Wonderful Wizard of Oz." The last line of the movie is spoken by Dorothy: "Oh, Auntie Em, there's no place like home!" Imagine that Auntie Em has a sister, Irma. It almost goes without saying that Dorothy would call her ... Anti Ir.[20] Those are the two 'negative' prefixes that we need to add, one at a time, in that order, to the equivalence order definition to create the two partial order definitions. You need to remember to which property names to add which prefixes, but that's not hard, either; 'antireflexive' and 'irsymmetric' just sound weird.[21]

[20]We don't care how loudly this made you groan; we refuse to apologize for it. Also, although you might expect Anti Ir to be very, very evil because her name is two negatives, the two cancel out, and so Dorothy loves her just as much as Auntie Em. Isn't that sweet?

[21]We hate to have to mention this, but ... 'anti–reflexive' is actually used an alternate name for irreflexivity by some people. On the bright side, 'irsymmetric' doesn't seem to have a meaning ... so far.

[22]You know, like Chekov was during the entire "Space Seed" episode of "Star Trek." Also, color was illegal in Kansas until 1967.

8.8 Total Orders

Remember the mention of the phrase 'total order' back in Section 8.6.1? It was in the quote from Java's documentation of the `Comparable` interface, alongside 'equivalence relation.' Sounds like they might be connected, doesn't it? They are, though not directly, at least as viewed through their definitions.

Just based on their names, you might assume that a partial order is an incomplete total order. That is, maybe you get a partial order when you start arranging something but get bored and don't complete the job, while a total order is what you'd have produced if you'd finished what you'd started. Actually, that's not too far off!

Before we can define what a total order is, we need to define a familiar term.

comparable

Definition 69: Comparable

Let R be a reflexive partial order on the set W, and let $w, x \in W$. w and x are *comparable* when either $w \preceq x$ or $x \preceq w$.

Note that the definition doesn't need to include "…but not both" in its wording, because reflexive partial orders are, by definition, antisymmetric.

Example 195:

Question: Let $D = \{(x, y) \mid x \text{ divides } y\}$, where $x, y \in \{1, 2\}$. Are 1 and 2 comparable, in D?

Answer: To answer this question properly, we need to remember that comparability depends on the relation being a reflexive partial order. Let's begin there: Is D a reflexive partial order?

For convenience, let's show D as a set of ordered pairs: $D = \{(1, 1), (1, 2), (2, 2)\}$. To be a reflexive partial order, a relation must be reflexive, antisymmetric, and transitive. D is reflexive (it contains $(1, 1)$ and $(2, 2)$), antisymmetric (it has $(1, 2)$ but not $(2, 1)$), and transitive ($(1, 1) - (1, 2) \rightsquigarrow (1, 2)$ and $(1, 2) - (2, 2) \rightsquigarrow (1, 2)$), making it a reflexive partial order.

Now we can worry about 1 and 2. For these values to be comparable, D must contain exactly one of $(1, 2)$ or $(2, 1)$. It contains just the former, so, yes, 1 and 2 are comparable in D.

With comparability explained, we can define *total order*.

Definition 70: Total Order

total order

Let T be a reflexive partial order on the set S, and let $r, s \in S$. T is a *total order* when each pair of elements r and s are comparable. (Or, T is a total order when it is antisymmetric, transitive, and each pair of elements r and s are comparable.)

Note that the definition is really two versions. Either is correct. The first version is just one step beyond a reflexive partial order, making it handy if you already know that about the relation. The second version follows the same three–property pattern as the partial order definitions. The reason that the second version doesn't include reflexivity is that comparability requires that all of the reflexive ordered pairs be present.

Example 196:

Question: In Example 191, we demonstrated that the relation $L = \{(i, j) \mid i \leq j\}$ on the set $P = \{11, 13, 17, 19\}$ is a reflexive partial order. Is L also a total order?

Answer: Because we already know L to be a reflexive partial order, all that we need to demonstrate is that all pairs of elements in P are comparable in L. Let's consider 17 as an example. The four ordered pairs in L that include 17 are (11,17), (13,17), (17,17), and (17,19), demonstrating that all elements of P are comparable with 17 (including 17 itself). Because we can also identify the corresponding ordered pairs for all three of the other elements of P, L is a total order.

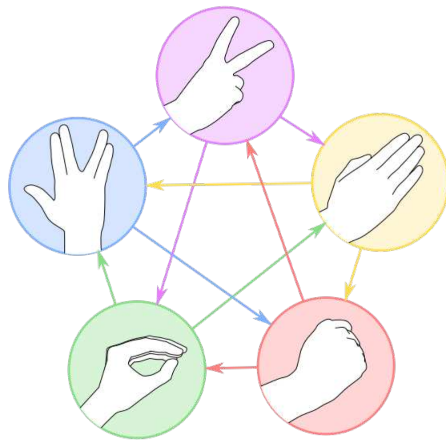


Figure 8.18: A graph of Rock–Paper–Scissors–Lizard–Spock. Clockwise from the top: Scissors, paper, rock, lizard, Spock. Credit: Wikimedia Commons.

Example 197:

The hand game Rock–Paper–Scissors (a.k.a. Roshambo) is a popular way to resolve a dispute with another person. It has variants that date back centuries. Each person simultaneously uses one of their hands to make one of those three shapes. If the first person makes a rock and the second scissors, the first person wins, because rock crushes scissors. Similarly, scissors cut paper, and paper covers rock. If both people make the same shape, they try again until one wins.

A popular modern extension is Rock–Paper–Scissors–Lizard–Spock, created by Sam Kass and Karen Bryla. Figure 8.18 shows a graph of the game's "who defeats whom" relation. As described by Kass: "Scissors cuts Paper covers Rock crushes Lizard poisons Spock smashes Scissors decapitates Lizard eats Paper disproves Spock vaporizes Rock crushes Scissors."[23] What Kass (and every graph we've ever seen of this version) didn't bother to include are the ties (e.g., rock ties rock), which can certainly occur. In ordered pairs notation, including the ties, the game's relation is: $R = \{$(scissors, paper), (paper, rock), (rock, lizard), (lizard, spock), (spock, scissors), (scissors, lizard), (lizard, paper), (paper, spock), (spock, rock), (rock, scissors), (rock, rock), (paper, paper), (scissors, scissors), (lizard, lizard), (spock, spock)} on the set $B = \{$rock, paper, scissors, lizard, spock$\}$.

Question: Is R a total order?

Answer: To show that the answer is 'yes' (which it seems it must be for the game to work), let's use the second version of the total order definition this time. We have three tasks: Show antisymmetry, transitivity, and, finally, comparability of every pair of elements a and b of B within R, as (a, b) or (b, a).

Task #1: Is R antisymmetric? This takes a bit of work, but with a little patience we can verify that no non–reflexive (a, b) pair in R has a (b, a) pair. (If one did, we couldn't determine who wins.) Yes, R is antisymmetric.

Task #2 is our old nemesis, transitivity, which . . . trips us up! Here's an example to demonstrate: R contains (rock, scissors) and (scissors, paper). To be transitive, R must also contain (rock, paper), but it doesn't. Because R is not transitive, the answer to the question is 'no:' R is not a total order.

Does this result bother you? Are you concerned that the game is broken because its relation isn't totally ordered? Don't let it worry you. The game needs antisymmetry and comparability to work, but it doesn't need transitivity, because the game doesn't need $(f, g) - (g, h)$ sequences of ordered pairs. (If you examine R, you'll find that B meets the "$w \preceq x$ or $x \preceq w$" requirement, meaning that the game meets the comparability requirement.) You can think of the game's components as being ordered enough for its needs, but not *totally* ordered. For the same reason, the original Rock–Paper–Scissors isn't based on a total ordering, either.

[23] Source: http://www.samkass.com/theories/RPSSL.html

