Chapter 2
Logical Seeing

Abstract This chapter serves as an interlude. Our goal in the following chapters is to present a formalized approach to numbers, and then we will look at the number systems again to see how tools of logic are used to uncover their essential features. We will be inspecting the *structure* of the number systems with our logic glasses on, but we need to get used to wearing those glasses. In this chapter we will take a look at some simple finite structures—finite graphs—and we will examine them from the logical perspective. In other words, later, logic will help us to see structures; now, some simple structures will help us to see logic. An important concept of symmetry of a graph is introduced in Definition 2.1 followed by equally important Theorem 2.1. Both, the definition and the theorem, will be generalized later to arbitrary mathematical structures.

Keywords Finite graphs · Graph symmetry · Fixed points · Types of elements · Addition and multiplication as relations

2.1 Finite Graphs

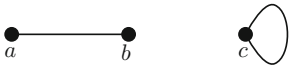
A graph is a collection points, called vertices, some of which are connected by edges. Graphs are mathematical structures. The collection of vertices is the domain of the structure, and the vertices are related by the *edge relation*. The edge relation is *symmetric*: An edge between vertices v and w is also an edge (the same edge) between w and v. Loops are allowed. A *loop* is an edge between a vertex and itself. In graph theory one encounters different types of graphs. There are directed graphs in which the edge relation is not symmetric, and there are graphs in which multiple edges between vertices are allowed. To keep the matters as simple as possible, we will not discuss directed graphs, and we will not allow multiple edges.

The first-order language for graphs has one binary relation symbol. We will use E. Thus, the atomic formulas (see Definition 1.1) are all formulas of the form $E(x_i, x_j)$, where x_i and x_j are arbitrary variables. If v and w are vertices in a graph, and x_i is interpreted by v, and x_j by w, then $E(x_i, x_j)$ is true under this evaluation if there is an edge between v and w, otherwise it is false.

R. Kossak, *Mathematical Logic*, Springer Graduate Texts in Philosophy 3,
https://doi.org/10.1007/978-3-319-97298-5_2

So a graph is a set of vertices equipped with an edge relation. We will be looking at visual representations of graphs and from those representations the graphs will inherit some additional features. Some vertices will be on the right, some on the left, some higher, some lower. Those are not the intrinsic properties of graphs, and they have to be disregarded. It will help to think of a graph as a database: just a list of vertices and edges.

Let us consider a graph, let's call it G, with the set of three vertices, which we will call a, b, and c. The edges of G are represented by ordered pairs (a, b), (b, a), and (c, c). The graph G has an edge between a and b, and a loop from c to c. There is a small technical inconvenience: the edge relation is symmetric, an edge between a and b is also an edge between b and a. Because relations are represented as sets of ordered pairs, this requires that for each edge between a vertex a and a vertex b, both pairs (a, b) and (b, a) are included in the edge relation of G. Here is a picture of the graph G:



The picture of the graph G tells the whole story. Nothing is hidden, and much of what we see is formally expressible. For example, $\exists x \exists y[\neg(x = y) \wedge E(x, y)]$ expresses that G has an edge connecting two different vertices, and $\exists x E(x, x)$ that G has a loop. The sentence

$$\exists x E(x, x) \wedge \neg\{\exists x \exists y[\neg(x = y) \wedge E(x, x) \wedge E(y, y)]\}$$

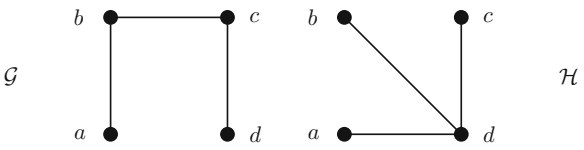
expresses that G has only one loop.

For each graph G, by V_G we will denote the set of vertices of G and by E_G the set of ordered pairs representing the edges of V_G. Consider two graphs G and H each with only one vertex, as in the picture blow. V_G has one vertex a, and E_G is empty (no edges); V_H has one vertex b, and E_H has one edge, the loop (b, b).

The graphs are different, we see it instantly, and first-order logic detects this difference as well. Consider the sentence $\exists x\ E(x, x)$. It is true when interpreted in H and false in G.

Now let us consider two other graphs $\mathcal{G}$ and $\mathcal{H}$.[1] each with four vertices a, b, c and d. Let the set of edges of $\mathcal{G}$ be (a, b), (b, a), (b, c), (c, b), (c, d), (d, c) and the set of edges of $\mathcal{H}$ be (a, d), (d, a), (b, d), (d, b), (c, d), and (d, c).

Before we take a closer look at $\mathcal{G}$ and $\mathcal{H}$, let me stress an important point one more time. The illustrations are graphic representations of graphs. Those representations show all graph-theoretic features, but they also show more. We see points of a particular size, edges with a particular thickness. We see angles at which edges meet, we can measure distances between vertices. All those additional features are not features of the structures we want to analyze. The only information that matters is provided in the lists of vertices of edges. That information determines each graph uniquely, and we will see how much of that information can be expressed in first-order statements. For example, the statement $\forall x \forall y[E(x, y) \implies E(y, x)]$ is a first-order statement that is true in all graphs, because the edge relation is symmetric. Given a finite list of vertices, and a set of ordered pairs of vertices, one can verify that it is an edge relation, by checking that for every pair (a, b) in the set, the pair (b, a) is also there.



Graphs $\mathcal{G}$ and $\mathcal{H}$ are visibly different. How does logic see it? Consider the sentence

$$\exists x \forall y[\neg(x = y) \implies E(x, y)].$$

It says that there is a vertex connected by an edge to every other vertex. This sentence is true about $\mathcal{H}$ and false about $\mathcal{G}$. There are other first-order expressible properties that allow us to see the difference between $\mathcal{G}$ and $\mathcal{H}$ (think of some).

As graphs grow larger it becomes more difficult to spot differences between them just by looking, and so it is with logic, sentences expressing differences are becoming more complex. A commonly used measure of complexity of a sentence is the number of quantifiers it has. In our first example we could logically see the difference between G and H with the aid of a sentence with just one quantifier. In the second example the statement that expresses the difference between $\mathcal{G}$ and $\mathcal{H}$ has two quantifiers. It can be shown, and it is a consequence of a general theorem about first-order logic, that for any first-order sentence φ with only one quantifier, if φ is a

[1] Notice the change of font. Throughout the book we will discuss various mathematical objects, and for brevity, we give them "names" that are typically just single letters, sometimes using different fonts. There is nothing formal about those names, and choices of names are quite arbitrary.

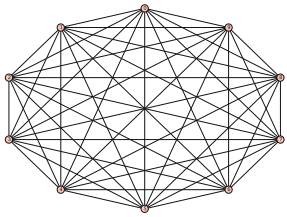
true statement about $\mathcal{G}$ if and only if it is true about $\mathcal{H}$. It is not that surprising once one realizes that not much can be expressed about a graph by a sentence with only one quantifier. Think about it and try to come up with such sentences. For example, here is one $\exists x E(x, x)$. Seeing the difference between $\mathcal{G}$ and $\mathcal{H}$ is harder than seeing the difference between G and H. How hard can it get in general?

In graph theory, K_n denotes a complete graph on n vertices, i.e. the graph that has n vertices and every two vertices are connected with an edge. Here are picture showing K_3 and K_4:

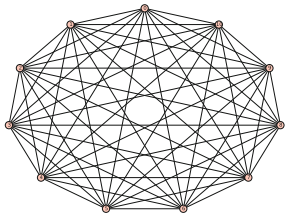
K_3 and K_4 are clearly different, but now it is a bit harder to find a first-order sentence that describes this difference. In fact, to see the difference in logical terms, we have to resort to a crude idea. Consider the sentence

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 [\neg(x_1 = x_2) \wedge \neg(x_1 = x_3) \wedge \neg(x_1 = x_4) \wedge \neg(x_2 = x_3) \wedge \neg(x_2 = x_4) \wedge \neg(x_3 = x_4)].$$

This sentence says that there are at least four vertices, so it is true about K_4, but false about K_3. It is a crude sentence. It does not mention anything about the edge relation. It is hard to tell the difference between K_3 and K_4 in any other way, because the edge relations in both graphs are very similar; every vertex is connected by an edge to any other vertex. Nothing special happens there. The sentence witnessing the difference has four quantifiers, so it is rather complex. It follows from the same general theorem we mentioned before that the difference between K_3 and K_4 cannot be expressed by a sentence with fewer than four quantifiers. Now, given a visualization, we see the difference immediately, but that is because triangles and squares are the kind of shapes we can recognize instantly. A quick glance reveals that K_3 is a complete graph with fewer vertices than K_4, and that K_4 is also complete. But how can one see the difference between graphs such as K_{10} and K_{11}? The graphs are similar; every vertex is connected to any other vertex, and except that K_{11} has visibly more edges than K_{10} it is hard to describe any other difference. In fact, K_{10} and K_{11} share all properties that can be expressed by a first-order sentence with fewer than 11 quantifiers. In a sense, the only difference between K_{10} and K_{11} is their size. Logic can express this difference by a sentence with 11 quantifiers, and we cannot do much better. Given two large complete graphs with similar sizes, we cannot tell whether they are different or not without counting their vertices.



(a) K_{10}



(b) K_{11}

2.2 Symmetry

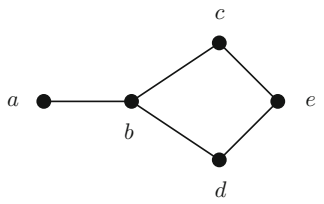
Symmetry is a subtle concept. In visual arts, music, and poetry it is used in many either straightforward or nuanced ways. Sometimes it is a desired effect, sometimes a feature to avoid. In classical geometry, the concept of symmetry is related to invariance under transformations. Rotating a square about its center by 90° results in an identical picture. If you think of a square as a collection of points, under such a rotation all points of the square have moved, but the picture looks the same as before. This is why the rotation by 90° is a symmetry of the square, but the one by 45° is not. In contrast, every rotation of a circle about its center is its symmetry. In this sense, the circle is more symmetric than the square. This example illustrates an important theme: We can study mathematical structures by investigating their symmetries. Knowing all symmetries of a structure tells us a lot about it, and in interesting cases it actually can characterize a structure completely.

Squares and circles are geometric objects, and what determines whether a certain transformation is a symmetry or not is whether the picture "looks" the same before and after. A similar idea can be applied to all mathematical structures, but we need to be precise about what is meant by "looking the same" before and after. What can change and what remains fixed? Logical seeing will help here.

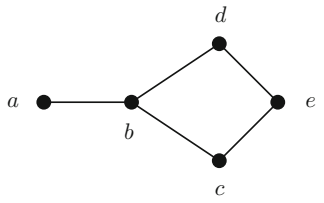
Caveat: We will now discuss first-order properties of graphs that are easy to visualize, but the discussion requires a higher level of precision. We will talk about functions and we will use mathematical notation you may not be familiar with.

Since we represent edges as ordered pairs of vertices, if the pair (v, w) is in the set of edges of a graph, ten the pair (w, v) must be there as well. To save space, in the lists of edges of the graphs below we will list each pair only once with understanding that its symmetric counterpart is in the set as well.

Consider the graph G with the set of vertices $\{a, b, c, d, e\}$, and the set of edges $E_G = \{(a, b), (b, c), (b, d), (c, e), (d, e)\}$.



Let us rearrange the picture by swapping c and d.



The image has changed. The vertex c which was up in the first picture, now is down, but what changed is not a first-order property of our structure. The picture representing the graph is now different, but it represents the same graph. Up, and down, are not "visible" without a geometric representation. The set of edges corresponding to the second picture, is the same as before. In other words, some vertices have moved, but the edge relation remained the same. Our transformation is a symmetry.

What can we learn about a structure by studying its symmetries? Let us examine our example more. Exchanging c with d was an obvious choice for the symmetry. Are there any other choices? A quick inspection shows that there are none. For example, exchanging a with b, is not a symmetry, and that is because a is connected by an edge only to one vertex, while b is connecter to three. If we swap a and b, the result will be a new graph G' whose edge relation is $E_{G'} = \{(a, b), (a, c), (a, d), (c, e), (d, e)\}$, so it is not the same as E_G. We can see the difference by looking at representations. How does logic see it? The notion of degree of a vertex will be helpful.

The *degree of a vertex* v is the number of vertices connected to v by an edge. In our graph G, the degree of a is 1, the degree of b is 3, and c, d, and e have degree 2. The formula

$$\exists y[E(x, y) \wedge (\forall z(E(x, z) \Longrightarrow z = y))]$$

expresses that the degree of a vertex x is 1. It says that there is a vertex y that is connected to x by an edge, and that there is only one such y; which is expressed by saying that if a vertex z forms an edge with x, then that z must be y. Let us call this formula $\varphi_1(x)$. When x is interpreted in G by a, $\varphi_1(x)$ becomes a true statement, and it is false when x is interpreted by any other vertex.

A small digression: As with some first-order statements we considered before, $\varphi_1(x)$ expresses the property we have in mind, but does it not in the most natural way. We have an advantage over logic, because we see the structure in its totality in its representation. We see that a is connected to b and to no other vertex. Logic does not have access to representations, it only knows the relation E_G as a set of ordered pairs. Given the set E_G, to verify that the degree of a is 1, one has to search the entire set and check that there is an edge (a, v) for some vertex v, and that there are no other edges (a, w), and this is really what $\varphi_1(x)$ expresses.

Here is a formula $\varphi_2(x)$ expressing that the degree of a vertex is 2, since we need more variables, we will be indexing them:

$$\exists x_1 \exists x_2 [\neg(x_1 = x_2) \wedge E(x, x_1) \wedge E(x, x_2) \wedge (\forall y(E(x, y) \Longrightarrow (y = x_1 \vee y = x_2))].$$

For each natural number n there is a first-order formula φ_n expressing that the degree of a vertex is n. Thus, logic can see degrees of all vertices, and this gives us a necessary condition for a transformation to be a symmetry. Any symmetry has to map vertices of a given degree to vertices of the same degree. In the graph G, a is the only vertex of degree 1, and b is the only vertex of degree 3, neither a nor b can be moved by a symmetry. The other three vertices have degree 2, and we already saw that swapping c and d, while fixing all other elements is a symmetry. But how about swapping c and e? A quick glance at the representation of G shows that this is not a symmetry. We see it, and logic sees it too. Here is how. Consider the formula $\varphi(x)$ in which for brevity we use $\varphi_1(x)$ defined above:

$$\exists x_1 \exists x_2 [E(x, x_1) \wedge E(x_1, x_2) \wedge \varphi_1(x_2)].$$

The formula expresses that in the graph one can get from the vertex x to a vertex of degree 1 moving along two edges. The formula is true when interpreted by c, but false when interpreted by e. In other words, $\varphi(x)$ is a first-order property that c has, but e does not. This is why swapping c and e is not a symmetry. That it is not a symmetry can also be checked by a direct inspection of the two sets of edges. Let H be the graph obtained by swapping c and e (draw a picture). Edge (b, e) is in E_H, but it was not in E_G. The edge relation has changed, so this transformation is not a symmetry.

We will now formally define the notion of symmetry of a graph, but first we need to fix some terminology. Above, we used the terms "transformations" and "mappings" of vertices without defining them. Now we will talk about functions. Functions are set-theoretic objects, and they will be introduced properly in Chap. 6. Informally, one can think of functions as operating on inputs, which in our case will be vertices of graphs, and turning out outputs, which will also be vertices. Functions

are given names, for example f, and when we write $f(x) = y$ we mean that y is the output of f on the input x. Often, the inputs are called arguments, and the outputs are called values. To indicate that the inputs of a function f are all the elements of a set X and the outputs are all in the set Y, we write $f : X \longrightarrow Y$.

Functions can be defined by specifying a procedure for obtaining the output on given input, or they can be defined by a list of inputs and outputs. The transformation that turned out to be a symmetry in our first example above, is a function, let us call it f, that is defined by $f(c) = d$, $f(d) = c$, $f(a) = a$, $f(b) = b$, $f(e) = e$. A function is called *one-to-one* if does not assign the same output to distinct inputs. The function f above is one-to-one, if we altered it slightly by defining $f(a)$ to be d (for example) it would not be one-to-one.

A function $f : X \longrightarrow Y$ is *onto* if for every b in Y there is an a such that $f(a) = b$. If a function $f : X \longrightarrow X$ (not a typo) is one-to-one and onto, then we call it a *permutation* of the set X.

Definition 2.1 Let G be a graph. A permutation $f : V_G \longrightarrow V_G$ is a *symmetry* of G if and only if for any pair of vertices v and w,

$$(v, w) \text{ is in } E_G \text{ if and only if } (f(v), f(w)) \text{ is in } E_G. \qquad (*)$$

Notice that while $(*)$ in Definition 2.1 looks similar to the first-order formulas we have examined before, it is not a formula in the first-order language of graphs. The reason is that it involves the symbol f that is not part of the vocabulary. This statement is not about the graph G, rather, it is a statement about the graph G and the function f.

Definition 2.1 will be generalized later to arbitrary structures.

In the introduction to this section, we said that a symmetry is a transformation that moves elements of the structure, but does not change the way the structure "looks," and we tied the "look" of the structure to the first-order properties of its elements. Definition 2.1 does not refer to all first-order properties, but only to those that are expressible by the simplest formulas—the atomic ones. It would seem that by considering more properties in the definition, we could obtain a sharper, more restrictive notion of symmetry. This however is not the case, and it is due to the following theorem.

Theorem 2.1 *Let G be a graph. A permutation $f : V_G \longrightarrow V_G$ is a symmetry if and only if for every first-order formula $\varphi(x_1, x_2, \ldots, x_n)$ of the language of graphs and for any sequence of vertices $v_1, v_2, \ldots, v_n$*

$$\varphi(v_1, v_2, \ldots, v_n) \text{ holds in } G, \text{ if and only if } \varphi(f(v_1), f(v_2), \ldots, f(v_n)) \text{ holds in } G. \qquad (**)$$

Compare $(*)$ and $(**)$. The statements are similar. The former says that symmetries preserve the edge relation; the latter says that in fact symmetries preserve all first-order properties as well. Theorem 2.1 has important consequences. Once know that

a one-to-one function preserves the edge relation, we also know that it preserves all first-order properties. Hence; it implies that to show that a certain function is not a symmetry, it is enough to find one first-order property that it does not preserve.

2.3 Types and Fixed Points

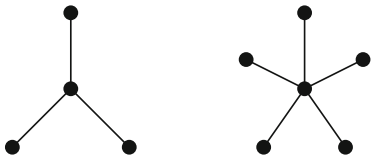
In the previous section, we learned that if f is a symmetry of a graph G, and $f(v) = w$, then the vertices v and w must have exactly the same first-order properties. For finite graphs, the converse to this statement is also true. It is a theorem and to formulate it neatly, let us introduce one more definition. Let v be a vertex of a graph G. The *type* of v in G is the collection of all first-order formulas $\varphi(x)$ in the language of graphs, which are true in G when x is interpreted by v. Knowing a structure means, in particular, knowing all types of its elements. The definition of type is straightforward, and it does not involve anything technically complicated—the type is a set of certain formulas. As a set, it is itself a mathematical object, but as with many other mathematical objects, one thing is to define them, another is to know them. By analyzing a structure we can establish that an element in it has this or that first-order property, but this is far from being able to fully describe the complete type of the element. Types are usually very complex.

Theorem 2.1 implies that if f is a symmetry of a graph G, then for every vertex v, the type of v and the type of $f(v)$ are the same. If the graph G is finite, then this statement has an interesting converse, that follows from Theorem A.1 that is discussed in the Appendix.

Theorem 2.2 *Suppose v and w are vertices of a finite graph G, and that the type of v and the type of w are the same. Then, there is a symmetry f of G such that $f(v) = w$.*

Theorem 2.2 has important consequences. A complete description of a structure must include a list of types of its elements, but those types are often not easy to classify. The theorem says that we can see how many different types of elements a structure has by analyzing its symmetries. If two vertices of a finite graph have the same type, then there is a symmetry mapping one to the other. It follows that if a vertex is fixed by all symmetries of the graph, then it must have a unique type. Identifying those unique vertices is an important stage in getting to know a structure. Let us illustrate this with a simple example.

For a natural number n, an *n-star* is a graph that has one distinguished vertex forming edges with n other vertices, and has no other vertices or edges. Let S_n denote an n-star graph. Here are pictures of S_3 and S_5.



S_3 S_5

The stars S_1 and S_2 are exceptions, but beginning with $n = 3$, each graph S_n has vertices of exactly two types. The vertex in the center has degree n, and all other vertices have degree 1. Any permutation that fixes the center, is therefore a symmetry of S_n. Except for the center, a symmetry can swap any vertex with any other vertex. Thus we obtain a complete description of all symmetries of S_n. The center is unique, it is fixed by all symmetries. All other vertices are fungible.

2.4 Seeing Numbers

In the next chapter we will use first-order logic to build and analyze structures made of numbers. As discussed earlier, we will treat addition and multiplication as relations. As a relation, addition is the set of all ordered triples (k, l, m) such that $k + l = m$. Here is what this set looks like:

(0,0,0), (0,1,1), (0,2,2), (0,3,3), (0,4,4), (0,5,5), (0,6,6), (0,7,7), (0,8,8), (0,9,9), ...

(1,0,1), (1,1,2), (1,2,3), (1,3,4), (1,4,5), (1,5,6), (1,6,7), (1,7,8), (1,8,9), (1,9,10), ...

(2,0,2), (2,1,3), (2,2,4), (2,3,5), (2,4,6), (2,5,7), (2,6,8), (2,7,9), (2,8,10), (2,9,11), ...

(3,0,3), (3,1,4), (3,2,5), (3,3,6), (3,4,7), (3,5,8), (3,6,9), (3,7,10), (3,8,11), (3,9,12), ...

(4,0,4), (4,1,5), (4,2,6), (4,3,7), (4,4,8), (4,5,9), (4,6,10), (4,7,11), (4,8,12), (4,9,13), ...

(5,0,5), (5,1,6), (5,2,7), (5,3,8), (5,4,9), (5,5,10), (5,6,11), (5,7,12), (5,8,13), (5,9,14), ...

...

And here is multiplication:

(0,0,0), (0,1,0), (0,2,0), (0,3,0), (0,4,0), (0,5,0), (0,6,0), (0,7,0), (0,8,0), (0,9,0), ...

(1,0,0), (1,1,1), (1,2,2), (1,3,3), (1,4,4), (1,5,5), (1,6,6), (1,7,7), (1,8,8), (1,9,9), ...

(2,0,0), (2,1,2), (2,2,4), (2,3,6), (2,4,8), (2,5,10), (2,6,12), (2,7,14), (2,8,16), (2,9,18), ...

(3,0,0), (3,1,3), (3,2,6), (3,3,9), (3,4,12), (3,5,15), (3,6,18), (3,7,21), (3,8,24), ...

(4,0,0), (4,1,4), (4,2,8), (4,3,12), (4,4,16), (4,5,20), (4,6,24), (4,7,28), (4,8,32), ...

(5,0,0), (5,1,5), (5,2,10), (5,3,15), (5,4,20), (5,5,25), (5,6,30), (5,7,35), (5,8,40), ...

...

The addition and multiplication of natural numbers, as we will treat them in this book, are exactly as illustrated above. They are sets of ordered triples. How can we "see" this structure? How can we study it?

While addition and multiplication presented as relations are more complex than the comparatively simple graphs we considered in this chapter, there is much that both types of structures have in common. In the number structure instead of vertices, we have numbers, and instead of relations being given by sets of ordered pairs, they are given by sets of ordered triples, but the logical framework for studying properties is exactly the same.

Both sets of triples above are infinite, so we cannot see them in their totality, but we can inspect their finite fragments to look for special features and regularities. Also, there are some basic facts about both relations that can guide us. We are interested in identifying numbers, and sets of numbers with interesting properties, and in particular, we want to see which properties can be expressed by formulas of first-order logic.

Recall, that we designated A, M, and L, as relations symbols for addition, multiplication, and the ordering, respectively. Here are some examples of first-order properties written using those symbols.

- $A(x, x, x)$. There is only one number that has this property. It is 0, since $0 + 0 = 0$, and for all other numbers n, $n + n$ does not equal n.
- $M(x, x, x)$. There are exactly two numbers with that property: 0 and 1.
- $\exists y A(y, y, x)$. This formula defines the set of numbers that are of the form $y + y$. Those are exactly the numbers that ere divisible by 2, i.e. the even numbers.
- Here is a more complex property.

$$\exists y \exists z [\neg M(y, y, y) \wedge \neg M(z, z, z) \wedge M(y, z, x)].$$

  In this formula, the only free variable is x. It defines the set of all those natural numbers x that have that property. The formula says that x is a product of two numbers y and z, and that y and z are neither 0 nor 1. In other words, it says that x is a composite number.
- And one more: $\exists y \exists z \exists v \exists w [M(y, y, v) \wedge M(z, z, w) \wedge A(v, w, x)]$. This is saying that for some y and z, $x = y^2 + z^2$, so this formula defines the set of those natural numbers that can be written as a sum of two squares. For example, $0 = 0^2 + 0^2$, $1 = 0^2 + 1^2$, $4 = 0^2 + 2^2$, and $5 = 1^2 + 2^2$, so all those numbers have the property, but 3 does not.

The examples above show how some natural number theoretic properties can be expressed in first-order logic. I call such properties *logically visible*. In the language of first-order logic, using only the relation symbols for addition and multiplication,

one can define almost all properties of natural numbers that number theorists are interested in. This is due to a special features of the arithmetic of natural numbers. In other, similar structures, logic allows one to see much less. This is the theme of the second part of this book.

To show that a certain property is expressible in first-order logic, one has to come up with a formal definition. Sometimes, those definitions are more or less straightforward translations of their informal versions. Sometimes, they are arrived at laboriously in a process that requires deeper insight into their nature.

To show that a certain property is not logically visible is another matter. Here is where new mathematical ideas come to play, and an important role is played by symmetries. We will make all this precise in Part II.

Exercises

Exercise 2.1 *Write a formula $\varphi_0(x)$ in the first-order language of graphs expressing that the degree of a vertex x is 0, and $\varphi_3(x)$, expressing that the degree of x is* 3.

Exercise 2.2 *Write a general form a formula $\varphi_n(x)$ in the first-order language of graphs expressing that the degree of a vertex is n, for each natural number $n > 0$.*

Exercise 2.3 *The random graph is an interesting mathematical structure. It has infinitely many vertices and is defined by the following property: for any two finite disjoint sets of vertices A and B, there is a vertex v that forms an edge with every vertex in A and none of the vertices in B. This property cannot be expressed by a single first-order sentence, but can be expressed by infinitely many. For each n, there is one sentence for sets A and B of size n. For example, the sentence for $n = 1$ is:*

$$\forall x_1 \forall x_2 [\neg(x_1 = x_2) \Longrightarrow \exists x (E(x_1, x) \wedge \neg E(x_2, x)].$$

Write the sentences for $n = 2$ and $n = 3$.

Exercise 2.4 *Let V be $\{a, b, c\}$, and let E be $\{(a, b), (b, a), (b, c), (c, b), (a, a)\}$. Prove that the graph $G = (V, G)$ is rigid, i.e. it has no nontrivial symmetries.*[2]

Exercise 2.5 *Show that the following properties of natural numbers have first-order definitions in the language with relation symbols for addition and multiplication:*

1. *x is divisible by* 3.
2. *x is a sum of three squares.*
3. *x is a product of two prime numbers.*

[2]Every graph has the trivial symmetry, i.e. the symmetry f, such that $f(v) = v$ for each vertex v.

4. *x and $x + 2$ are prime (remember that 2 is not a symbol of first-order logic, so you cannot use 2 explicitly).*

Exercise 2.6 *Find a first-order property using a relation symbol for addition of natural numbers, that 2 has, but 3 does not.*