Chapter 1

Preliminaries

1.1 First-order logic

1.1.1 Languages

A *formal language* is a set of symbols. There are three kinds of symbols: constant symbols, function symbols, and relation symbols. The names of these categories of symbols specify the intended use for the symbol: a constant symbols are intended to represent fixed elements of a *structure* (the definition will be given later), function symbols are intended to represent operations in a structure, and relation symbols are intended to represent relations.

Formally, we should also mention the logical symbols of the language, but since these will always be the same, we will only specify the non-logical symbols of languages. For the sake of completeness, I will mention here that all languages in this text are assumed to include a symbol for true equality ($=$), the logical symbols $\neg$ and $\wedge$, brackets, a countably infinite supply of variable symbols, and the quantifier $\exists$.

We will work in *classical first-order* logic. This means two things: first, the rules of inference will be those of classical logic (modus ponens, elimination of double negations, the usual stuff. See ASDF CITE), and that quantification can be done only over *individual variables*, that is, variables intended to represent members of a structure (as opposed to operations or relations). This is part of a hierarchy of logics; for example, the propositional calculus (i.e., no quantification) is sometimes called zeroth-order logic, and quantification over relations and functions is allowed in second-order logic.

We use strings of symbols from the language (together with variables and logical symbols) to express mathematical properties. To do this, we need to agree on a set of meaningful expressions (formulas) and what it means for a formula to be true in a structure, but first we need to know how many arguments each relation and function symbol requires. Thus, relation and function symbols have an associated *arity* which specifies precisely that. For example, the edge relation in graphs requires the two endpoints of the edge, so it needs a binary (arity 2) relation symbol to represent it.

Language is a choice. One could express all the axioms of group theory using only a binary function symbol for the group operation and a constant for the identity element, or including also a function symbol for the inverse operation.

1.1.2 Terms and formulas

In this subsection, we will define special sets of strings of symbols. One could think of terms as the nouns of a natural language, and formulas as statements, sometimes with free variables that we are free to interpret.

Definition 1. *The* terms *of a language L are the elements of the least set of strings of symbols that satisfies all of the following conditions:*

1. *Every variable is a term.*
2. *Every constant symbol from L is a term.*
3. *If f is an n-ary function symbol from L and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.*

So, in the language $L = \{+, *\} \cup \{z : z \in \mathbb{Z}\}$, where $+$ and $*$ are binary function symbols, the following are terms: 7, x_2, $*(6, 3)$, $+(5, *(x_1, x_2))$. The latter two can be written as 6*3 and $5 + x_1 * x_2$. At this point, we have not assigned any meaning to the symbols, so try to avoid interpreting them.

Definition 2. *The* atomic formulas *of a language L are the elements of the least set of strings of symbols that satisfy:*

1. *If s, t are terms of L, then $s = t$ is an atomic formula.*
2. *If R is an n-ary relation symbol from L and $t_1, \ldots, t_n$ are terms of L, then $R(t_1, \ldots, t_n)$ is an atomic formula.*

As an example $x_1 \in \varnothing$ and $h(\varnothing) = f(g(x_1, x_2, h(x_3)))$ are atomic formulas in the language $\{\in, f, g, h, \varnothing\}$, where f and h are unary function symbols, g is a ternary function symbol, $\in$ is a binary relation symbol, and $\varnothing$ is a constant symbol. I don't want to push the analogy with natural languages too far, but you can see that relation symbols (and here I include equality as a binary relation symbol) act as verbs; the atomic formulas are similar to very simple sentences such as "the ball is blue" etc.

The next level of complexity is formulas. This is where logical connectives and quantifiers come into play.

Definition 3. *The set of* formulas *over a language L (or L-formulas) is the least set of strings that satisfy:*

1. *Every atomic formula over L is a formula.*
2. *If φ is a formula, then $\neg\varphi$ is a formula.*

3. *If φ and ψ are L-formulas, then $\varphi \wedge \psi$ is a formula.*

4. *If φ is a formula and x is a variable, then $\exists x \varphi$ is a formula.*

As a special case, the set of atomic and negated atomic L-formulas is called the set of *literals* of L.

Notice that each of the definitions above allows us to evaluate whether a given string is a term or formula. The rules of construction also allow us to perform proofs by *induction on formulas*: if we wish to prove that a statement P holds for all formulas, we prove that it holds for atomic formulas (this is the basis for induction) and then, assuming that P holds for φ and ψ, we prove that P holds for $\neg\varphi, \varphi \wedge \psi$, and $\exists x \varphi$.

Another important remark here is that, since we only allow finite arities, formulas are always finite strings. There are also systems in which infinite strings are allowed as formulas; those interested should look into *infinitary logic.*ASDF CITE

1.1.3 Structures, homomorphisms, substructures

So far, formulas are simply well-formed strings of symbols, free of any meaning. If we wish to use them to express properties of, say, the ring of integers or a metric space, then we need a mechanism to interpret what they mean.

Definition 4. *Let L be a language. An L-structure $\mathcal{A}$ is a set A (the domain or universe of the structure) equipped with:*

- *Distinguished elements $c^{\mathcal{A}}$ for each constant symbol $c \in L$.*
- *For each k, a k-ary relation $R^{\mathcal{A}} \subset A^k$ for each k-ary relation symbol $R \in L$.*
- *For each k, a k-ary function $f^{\mathcal{A}} \colon A^k \to A$ for each k-ary function symbol $f \in L$.*

For example, the ordered field of rational numbers is the $\{+, -, 0, 1, *, <\}$-structure $(\mathbb{Q}; +, -, 0, 1, *, <)$, with the usual interpretations of the symbols, but any other interpretation of the symbols is also an $\{+, -, 0, 1, *, <\}$-structure.

We can think of a structure as the base set and three (partial) functions $I_C \colon L \to A, I_R \colon L \to \bigcup\{A^n : n \in \mathbb{N}\}, I_F \colon L \to \bigcup\{A^{A^n} : n \in \mathbb{N}\}$ that yield the distinguished elements, relations and symbols in the structure. These are called *interpretation functions.*

Interpreting formulas

Once we have an L-structure, the interpretation functions give us almost directly a way to interpret terms and atomic formulas. In the formal definition, we will use the word *complexity* to mean the number of symbols in a term. We write $t(x_1, \dots, x_n)$ for a term in which some of the variables $x_1, \dots, x_n$ appear.

Definition 5. *If $t(x_1, \ldots, x_n)$ is an L-term, $\mathfrak{A}$ is an L-structure, and $\bar{a} = (a_1, \ldots, a_n)$ is a tuple of elements from A, we define $t^{\mathfrak{A}}[\bar{a}]$ as follows:*

- *If t is the variable x_i, then $t^{\mathfrak{A}}[\bar{a}]$ is a_i.*
- *If t is the constant $c \in L$, then $t^{\mathfrak{A}}[\bar{a}]$ is $c^{\mathfrak{A}}$.*
- *If t is $f(s_1, \ldots, s_n)$ and each s_i is a term $s_i(x_1, \ldots, x_n)$, then $t^{\mathfrak{A}}[\bar{a}]$ is $f^{\mathfrak{A}}(s_1^{\mathfrak{A}}[\bar{a}], \ldots, s_n^{\mathfrak{A}}[\bar{a}])$.*

The notation is a little heavy, but the meaning should be clear. Say we have the Abelian group of integers as a structure $Z = (\mathbb{Z}; 0, +, -)$ for $L = \{e, +, -\}$, interpreting e as 0 and $+, -$ in the usual way. Consider the terms $t_1(x_1, x_2, x_3) = x_2$, $t_2(x_1, x_2, x_3) = +(+(x_1, x_3), -(x_2)))$, $t_3(x_1, x_2, x_3) = e$. Then

- $t_1^Z[(5, 3, 7)] = 3$.
- $t_1^Z[(53, 13, 274)] = 13$.
- $t_2^Z[(5, 3, 7)] = +(+(5, 7), -(3))$, or 5+7+(-3), which we can evaluate as 9 using the interpretations of $+$ and $-$.
- $t_3^Z[(5, 3, 7)] = 0$.

Now we can define what it means for a formula $\varphi(x_1, \ldots, x_n)$ to be true in a structure $\mathfrak{A}$.

Definition 6. *Let $\varphi(x_1, \ldots, x_n)$ be an L-formula, $\mathfrak{A}$ an L-structure, and $\bar{a} = (a_1, \ldots, a_n)$ be a tuple of elements of A.*

- *If φ is the formula $t_1 = t_2$ where t_1 and t_2 are L-terms, then $\mathfrak{A} \models \varphi(\bar{a})$ if $t_1^{\mathfrak{A}}(\bar{a}) = t_2^{\mathfrak{A}}(\bar{a})$.*
- *If φ is the formula $R(s_1, \ldots, s_n)$ where R is an n-ary relation symbol and $s_1, \ldots, s_n$ are L-terms, then $\mathfrak{A} \models \varphi(\bar{a})$ if $(t_1^{\mathfrak{A}}[\bar{a}], \ldots, t_n^{\mathfrak{A}}[\bar{a}]) \in R^{\mathfrak{A}}$.*
- *If φ is the formula $\neg\psi$, then $\mathfrak{A} \models \varphi[\bar{a}]$ if $\mathfrak{A} \models \psi[\bar{a}]$ does not hold.*
- *If φ is the formula $\psi \wedge \xi$, then $\mathfrak{A} \models \varphi[\bar{a}]$ if $\mathfrak{A} \models \psi[\bar{a}]$ and $\mathfrak{A} \models \xi[\bar{a}]$.*
- *If φ is the formula $\exists x \psi(x, x_1, \ldots, x_n)$, then $\mathfrak{A} \models \varphi[\bar{a}]$ if there exists some b such that $\mathfrak{A} \models \psi[b, \bar{a}]$.*

Each formula $\varphi(x_1, \ldots, x_n)$ defines a subset of A^n, namely $\{\bar{a} : \mathfrak{A} \models \varphi[\bar{a}]\}$. These are known as *definable sets.* Definable functions are definable sets which happen to be functions.

If we have chosen the language and interpretation wisely, then all the relations and operations that we care about will be either interpretations of symbols from the language or definable from them. When studying a theory, we are also interested in mappings between structures that preserve relations and operations (think for example of linear transformations: their defining characteristic is that linear relations are preserved).

Definition 7. *Let $\mathscr{A}, \mathscr{B}$ be L-structures. A function $h\colon A \to B$ is a* homomorphism *if*

- *for all relation symbols in L, if $(a_1, \ldots, a_n) \in R^{\mathscr{A}}$, then we have*

$$(h(a_1), \ldots, h(a_n)) \in R^{\mathscr{B}}.$$

- *for all function symbols in L we have*

$$h(f^{\mathscr{A}}(a_1, \ldots, a_k)) = f^{\mathscr{B}}(h(a_1), \ldots, h(a_k)).$$

As an exercise, prove that a function $h\colon A \to B$ is a homomorphism $\mathscr{A} \to \mathscr{B}$ if and only if all atomic formulas are preserved (meaning that $\mathscr{A} \models \varphi[\bar{a}]$ implies $\mathscr{B} \models \varphi[h(\bar{a})]$).

There are some stronger notions of homomorphism that get special names. A homomorphism is *strong* if it preserves the complements of relations; an *embedding* is a strong injective homomorphism; and an *isomorphism* is a surjective embedding. You can extend the exercise from the paragraph above to proving that h is an embedding iff it preserves all literals. Can you think of analogue statements for strong homomorphisms and isomorphisms?

Definition 8. *An L-structure $\mathscr{A}$ is a* substructure *of $\mathscr{B}$ if:*

- *$A \subset B$,*
- *$c^{\mathscr{A}} = c^{\mathscr{B}}$ for all constant symbols $c \in L$,*
- *$R^{\mathscr{A}} = R\mathscr{B} \cap A^n$ for each n-ary relation symbol $R \in L$, and*
- *$f^{\mathscr{A}} = f^{\mathscr{B}} \restriction_{A^n}$ for each n-ary function symbol $f \in L$.*

Here we should be careful about language. As mentioned before, all the axioms of group theory can be written in different languages, but the substructures that we get from each language may not be subgroups. If we write the axioms of group theory in the language $L_1 = \{*\}$, then any subset of the group which is closed under multiplication will be a substructure; if we include a constant for the identity, $L_2 = \{*, e\}$, then the substructures will be the submonoids of the group; but if we include also a symbol for the inverse then we get subgroups as substructures.