Supplementary Lecture E

A Crash Course in Logic

In this lecture we present the basics of first-order logic as background for the lectures on the complexity of logical theories (Lectures 21–25).

What Is Logic?

Logic typically has three parts: *syntax*, *semantics*, and *deductive apparatus*. *Syntax* is concerned with the correct formation of expressions—whether the symbols are in the right places. *Semantics* concerns itself with meaning—how to interpret syntactically correct expressions as meaningful statements about something. Finally, the *deductive apparatus* gives rules for deriving theorems mechanically. We do not concern ourselves much with deductive apparatus here.

Relational Structures

First-order logic is good for expressing and reasoning about basic mathematical properties of algebraic and combinatorial structures. Examples of such structures are: groups, rings, fields, vector spaces, graphs, trees, ordered sets, and the natural numbers.

Such a structure $\mathcal{A}$ typically consists of a set A, called the *domain* or *carrier* of $\mathcal{A}$, along with some distinguished n-ary functions $f^{\mathcal{A}} : A^n \to A$

for various n, constants $c^{\mathcal{A}} \in A$ (which can be viewed as 0-ary functions), and n-ary relations $R^{\mathcal{A}} \subseteq A^n$ for various n. The number of inputs n is called the *arity* of the function or relation. Functions or relations of arity 0, 1, 2, 3, and n are called *nullary, unary, binary, ternary*, and *n-ary*, respectively.

The list of distinguished functions and relations of $\mathcal{A}$ along with their arities is called the *signature* of $\mathcal{A}$. It is usually represented by an alphabet Σ of function and relation symbols, one for each distinguished function or relation of $\mathcal{A}$, each with a fixed associated arity.

Example E.1 The structure $\mathbb{N}$ of number theory consists of the set $\omega = \{0, 1, 2, \dots\}$, the natural numbers, along with the binary operations of addition and multiplication, constant additive and multiplicative identity elements, and the binary equality relation. The signature of number theory is $(+, \cdot, 0, 1, =)$, where $+$ and $\cdot$ are binary function symbols, 0 and 1 are constant symbols, and $=$ is a binary relation symbol.

A *group* is any structure consisting of a set with a binary multiplication operation, a unary inverse operation, a constant identity element, and a binary equality relation, satisfying certain properties. The signature of group theory is $(\cdot, {}^{-1}, 1, =)$, where $\cdot$ is a binary function symbol, ${}^{-1}$ is a unary function symbol, 1 is a constant symbol, and $=$ is a binary relation symbol.

A *partial order* is any set with a binary inequality relation and a binary equality relation satisfying certain properties. The signature of the theory of partial orders is $(\leq, =)$, where $\leq$ and $=$ are binary relation symbols. □

When discussing structures in general, we usually assume a fixed but arbitrary signature Σ. We usually use $f, g, \dots$ to denote function symbols of arity at least one, $c, d, \dots$ to denote constant symbols, and $R, S, \dots$ to denote relation symbols. The functions and relations they represent in the structure $\mathcal{A}$ are denoted $f^{\mathcal{A}}$, $c^{\mathcal{A}}$, $R^{\mathcal{A}}$, and so on.

At the risk of confusion, when working in a specific structure, we often use the same symbol for both the symbol of Σ and the semantic object it denotes; for example, in number theory, we might use $+$ to denote both the symbol of the signature of number theory and the addition operation on the natural numbers.

Syntax

The syntax of first-order logic can be separated into two parts, the first application-specific and the second application-independent. The application-specific part specifies the correct formation of *terms* from the symbols of Σ. The application-independent part specifies the correct formation of *formulas* from propositional connectives $\vee$, $\wedge$, $\neg$, $\rightarrow$, $\leftrightarrow$, 0 (falsity),

and 1 (truth), variables $x, y, z, \ldots$, quantifiers $\forall$ and $\exists$, and parentheses. These symbols are part of every first-order language.

Terms

Fix a signature Σ, and let X be a set of *variables*. A *term* is a well-formed expression built from the function symbols of Σ and variables X, regarding elements of X as symbols of arity 0. Here *well formed* means that the arities of all the symbols are respected. For example, if f is a binary function symbol, g is a unary function symbol, c, d are constant symbols, and x, y are variables, then

$$c \qquad x \qquad f(g(x), f(c, g(y))) \qquad g(f(g(x), c), f(d, g(y)))$$

are typical terms.

Depending on custom, terms involving binary function symbols are sometimes written in infix notation, as in $(x + 1) \cdot y$, and those involving unary function symbols are sometimes written in postfix notation, as in x^{-1}.

Valuations and the Meaning of Terms

A *valuation* over a structure $\mathcal{A}$ with domain A is a map from variables to values:

$$u : X \quad \to \quad A.$$

These maps are often called *environments* in programming language semantics. Any valuation extends uniquely by induction to a map

$$u : \{\text{terms}\} \quad \to \quad A$$

as follows: for any terms $t_1, \ldots, t_n$ and n-ary function symbol f,

$$u(f(t_1, \ldots, t_n)) \quad \stackrel{\text{def}}{=} \quad f^{\mathcal{A}}(u(t_1), \ldots, u(t_n)).$$

This definition also includes the case $n = 0$: for constants c, $u(c) = c^{\mathcal{A}}$. A term with no variables is called a *ground term*. Note that for ground terms t, the value $u(t)$ is independent of u. For this reason we often write $t^{\mathcal{A}}$ instead of $u(t)$ for ground terms t.

If u is a valuation, x is a variable, and $a \in A$, we denote by $u[x/a]$ the valuation that agrees with u except on variable x, on which it takes the value a. In other words,

$$u[x/a](y) \quad \stackrel{\text{def}}{=} \quad \begin{cases} u(y), & \text{if } y \neq x, \\ a, & \text{otherwise.} \end{cases}$$

The operator $[x/a]$ is called a *rebinding operator*.

Formulas and Sentences

An *atomic formula* is either a Boolean constant 0 or 1 or an expression of the form $R(t_1, \ldots, t_n)$, where R is an n-ary relation symbol of the signature and $t_1, \ldots, t_n$ are terms. Depending on the application, atomic formulas involving binary relation symbols are sometimes written in infix notation, as in $g(x) = y$.

Formulas are defined inductively:

- Every atomic formula is a formula;
- If φ and ψ are formulas and x is a variable, then the following are formulas: $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \to \psi$, $\varphi \leftrightarrow \psi$, $\neg\varphi$, $\exists x\ \varphi$, and $\forall x\ \varphi$.

We use parentheses in ambiguous situations when it is not clear how to parse the formula. Quantifiers may appear more than once with the same variable in the same formula.

For example,

$$\exists x\ ((\forall z\ y \leq z) \to x \leq y) \tag{E.1}$$

is a typical formula of the first-order language of ordered structures.

Scope, Free and Bound Occurrences of Variables

Suppose the formula φ has an occurrence of a subformula of the form $\mathsf{Q}x\ \psi$, where Q is a quantifier, either $\exists$ or $\forall$. The *scope* of the $\mathsf{Q}x$ in that occurrence of $\mathsf{Q}x\ \psi$ is that occurrence of ψ. (We have to say "occurrence" because quantifiers and subformulas can have more than one occurrence in a given formula.)

Consider an occurrence of a variable x in a formula φ (as a term, not as part of a quantifier expression $\mathsf{Q}x$). Such an occurrence of x is called *bound* if it is in the scope of a quantifier $\mathsf{Q}x$, *free* if not. A bound occurrence of x is *bound to* the occurrence of $\mathsf{Q}x$ with the smallest scope in which that occurrence of x occurs.

For example, in (E.1), the scope of the $\exists x$ is $((\forall z\ y \leq z) \to x \leq y)$, and the scope of the $\forall z$ is $y \leq z$. The single occurrence of x is bound to the $\exists x$, the single occurrence of z is bound to the $\forall z$, and the two occurrences of y are free. In

$$\exists x\ ((\forall y\ y \leq z) \to x \leq y), \tag{E.2}$$

on the other hand, the single occurrence of x is bound to the $\exists x$, the first occurrence of y is bound to the $\forall y$, and the single occurrence of z and the second occurrence of y are free.

A *sentence* is a formula with no free variables.

It is customary to write $\varphi(x_1, \ldots, x_n)$ to indicate that all free variables of φ are among $x_1, \ldots, x_n$.

Interpretation of Formulas and Sentences

Given a structure $\mathcal{A}$ and a valuation of variables u over $\mathcal{A}$, every formula has a truth value defined inductively as follows. We write

$$\mathcal{A}, u \models \varphi$$

and say "φ is true in $\mathcal{A}$ under valuation u" if the truth value associated with the formula φ is 1 (true) under the inductive definition we are about to give.

For atomic formulas, $\mathcal{A}, u \models 1$ always, $\mathcal{A}, u \models 0$ never, and

$$\mathcal{A}, u \models R(t_1, \ldots, t_n) \iff^{\text{def}} R^{\mathcal{A}}(u(t_1), \ldots, u(t_n)).$$

For compound formulas,

$$\begin{aligned}
\mathcal{A}, u \models \varphi \wedge \psi &\iff^{\text{def}} \mathcal{A}, u \models \varphi \text{ and } \mathcal{A}, u \models \psi \\
\mathcal{A}, u \models \varphi \vee \psi &\iff^{\text{def}} \mathcal{A}, u \models \varphi \text{ or } \mathcal{A}, u \models \psi \\
\mathcal{A}, u \models \neg\varphi &\iff^{\text{def}} \text{it is not the case that } \mathcal{A}, u \models \varphi \\
\mathcal{A}, u \models \exists x\ \varphi &\iff^{\text{def}} \text{there exists } a \in \mathcal{A} \text{ such that } \mathcal{A}, u[x/a] \models \varphi \\
\mathcal{A}, u \models \forall x\ \varphi &\iff^{\text{def}} \text{for all } a \in \mathcal{A},\ \mathcal{A}, u[x/a] \models \varphi.
\end{aligned}$$

Whether $\mathcal{A}, u \models \varphi$ depends only on the values that u assigns to the free variables of φ. In other words, if u and v agree on all variables with a free occurrence in φ, then $\mathcal{A}, u \models \varphi$ iff $\mathcal{A}, v \models \varphi$. This can be shown by induction on the structure of φ. In particular, for sentences (formulas with no free variables), whether $\mathcal{A}, u \models \varphi$ does not depend on u at all. In this case we omit the u and write $\mathcal{A} \models \varphi$ and say "φ is true in $\mathcal{A}$" if the sentence φ is true in $\mathcal{A}$ under any valuation (hence all valuations).

If Φ is a set of sentences, we write $\mathcal{A} \models \Phi$ if $\mathcal{A} \models \varphi$ for all $\varphi \in \Phi$.

If the free variables of φ are all among $x_1, \ldots, x_n$, that is, if $\varphi = \varphi(x_1, \ldots, x_n)$, and if $a_1, \ldots, a_n \in \mathcal{A}$, it is common to abuse notation by writing

$$\mathcal{A} \models \varphi(a_1, \ldots, a_n) \tag{E.3}$$

for

$$\mathcal{A}, u \models \varphi(x_1, \ldots, x_n), \tag{E.4}$$

where u is some valuation such that $u(x_i) = a_i$, $1 \leq i \leq n$. This is an abuse of notation because it is mixing syntactic objects (φ) with semantic objects ($a_1, \ldots, a_n$). Some authors deal with this by including a constant for each element of the domain of $\mathcal{A}$ and substituting the constant a_i for x_i in the definition of truth. Please just remember that anytime you see (E.3), although strictly speaking it is a type error, it really should be taken as an abbreviation for (E.4).

Prenex Form

There are semantics-preserving rules for transforming first-order formulas to a semantically equivalent special form called *prenex form*. In prenex form, all quantifiers occur first, followed by a quantifier free part. The rules are

$$\begin{aligned}
\varphi \wedge \forall x\ \psi(x) &\Leftrightarrow \forall x\ (\varphi \wedge \psi(x)) \\
\varphi \vee \forall x\ \psi(x) &\Leftrightarrow \forall x\ (\varphi \vee \psi(x)) \\
\varphi \wedge \exists x\ \psi(x) &\Leftrightarrow \exists x\ (\varphi \wedge \psi(x)) \\
\varphi \vee \exists x\ \psi(x) &\Leftrightarrow \exists x\ (\varphi \vee \psi(x)) \\
\neg\forall x\ \psi(x) &\Leftrightarrow \exists x\ \neg\psi(x) \\
\neg\exists x\ \psi(x) &\Leftrightarrow \forall x\ \neg\psi(x),
\end{aligned}$$

provided x does not occur free in φ. If x does occur free in φ, one can change the bound variable by applying the rule

$$\forall x\ \psi(x) \quad \Leftrightarrow \quad \forall y\ \psi(y),$$

where y is a new variable. To transform a formula to prenex form, the rules would be applied from left to right.

First-Order Theories

The *first-order theory of a structure* $\mathcal{A}$, denoted $\mathrm{Th}(\mathcal{A})$, is the set of sentences in the first-order language of $\mathcal{A}$ that are true in $\mathcal{A}$:

$$\mathrm{Th}(\mathcal{A}) \stackrel{\text{def}}{=} \{\varphi \mid \mathcal{A} \vDash \varphi\}.$$

For example, *first-order number theory* is the set of first-order sentences true in $\mathbb{N}$.

If $\mathcal{C}$ is a class of structures all of the same signature, the *first-order theory of* $\mathcal{C}$, denoted $\mathrm{Th}(\mathcal{C})$, is the set of sentences in appropriate first-order language that are true in all structures in $\mathcal{C}$:

$$\mathrm{Th}(\mathcal{C}) \stackrel{\text{def}}{=} \bigcap_{\mathcal{A}\in\mathcal{C}} \mathrm{Th}(\mathcal{A}).$$

For example, first-order group theory is the set of sentences in the language of groups that are true in all groups.

Axiomatization

If Φ is a set of sentences over some signature Σ, the class of *models of* Φ is the class of structures of signature Σ that satisfy all the sentences of Φ.

This class is denoted $\mathrm{Mod}(\Phi)$:

$$\mathrm{Mod}(\Phi) \stackrel{\text{def}}{=} \{\mathcal{A} \mid \mathcal{A} \vDash \Phi\}.$$

A sentence φ is called a *logical consequence* of a set of sentences Φ if it is true in all models of Φ. In other words, the set of logical consequences of Φ is the set $\mathrm{Th}(\mathrm{Mod}(\Phi))$.

We often specify a class of structures by giving a set of *axioms*, which are just first-order sentences. The class being specified is defined to be the class of models of those sentences.

Example E.2 A *group* is a structure of signature $(\cdot, 1, {}^{-1}, =)$ satisfying the first-order group axioms

$$\begin{array}{l}
\forall x\ \forall y\ \forall z\ x(yz) = (xy)z \\
\forall x\ x1 = x \\
\forall x\ 1x = x \\
\forall x\ xx^{-1} = 1 \\
\forall x\ x^{-1}x = 1
\end{array}$$

and the axioms of equality

$$\begin{array}{ll}
\forall x\ x = x & \forall x\ \forall y\ \forall z\ x = y \to xz = yz \\
\forall x\ \forall y\ x = y \to y = x & \forall x\ \forall y\ \forall z\ x = y \to zx = zy \\
\forall x\ \forall y\ \forall z\ (x = y \wedge y = z) \to x = z & \forall x\ \forall y\ x = y \to x^{-1} = y^{-1}.
\end{array}$$

□

The Decision Problem

The *decision problem* for a first-order theory is to determine whether a given sentence is an element of the theory. For a theory of a structure such as $\mathbb{N}$, this is just the problem of deciding whether a given sentence in the language of number theory is true in $\mathbb{N}$. For the theory of a class of structures $\mathcal{C}$, the decision problem is to determine whether a given sentence is true in all structures in the class; that is, whether it is in $\mathrm{Th}(\mathcal{C})$. For a set of first-order axioms Φ, the decision problem is to determine whether a given sentence is a logical consequence of Φ.