First-order logic

Robert J. Carroll

Last revised: 5 May 2026

---

1 Motivation

Propositional logic treats whole sentences as atomic. "D," for "the regime is democratic," is a single propositional letter; we cannot peer inside it. As soon as we want to say things like "every regime that is democratic is peaceful with other democracies," or "there exists a Condorcet winner," or "some voter prefers a to b and some other voter prefers b to a," the propositional language runs out of room. We need variables that range over individuals, predicates we can apply to those variables, and the quantifiers $\forall$ ("for every") and $\exists$ ("there exists") to bind them. *First-order logic* (FOL) is the formal language that adds these things, and it is the lingua franca of essentially all rigorous mathematical talk: every theorem in algebra, analysis, set theory, social choice, and formal political theory can be stated as a first-order formula over an appropriate signature.

The handout works through the syntax (signatures, terms, formulas, quantifier binding), the semantics (structures and satisfaction — the formal version of "writing down a model"), and the natural extension to *many-sorted* first-order logic, where the universe of discourse comes pre-divided into typed regions (voters, candidates, strategies, alternatives) rather than being a single undifferentiated set. The vocabulary developed here — signature, structure, interpretation, satisfaction — is the entry point to model theory, which is taken up in the next handout. The proof-system side (Gödel's completeness theorem extending the propositional-logic one of the previous handout) we mention but do not develop.

2 Syntax

The syntactic apparatus of FOL is heavier than that of propositional logic, but most of the weight is bookkeeping. We need to fix what symbols are available, what counts as a term (an expression denoting an individual), what counts as a formula (an expression that has a truth value once we know what everything means), and how the quantifiers bind variables.

Definition 1. A *signature* (or *vocabulary*, or *similarity type*) is a triple $\sigma = (\mathcal{C}, \mathcal{F}, \mathcal{R})$ together with an arity function α assigning a positive integer to each function and relation symbol:

- $\mathcal{C}$ is a set of *constant symbols*;
- $\mathcal{F}$ is a set of *function symbols*, with $\alpha(f) \in \{1, 2, 3, \ldots\}$ giving the arity of each $f \in \mathcal{F}$;
- $\mathcal{R}$ is a set of *relation symbols* (or *predicate symbols*), with $\alpha(R) \in \{1, 2, 3, \ldots\}$ giving the arity.

A signature is the "grammar in advance" of any specific theory — it pins down which non-logical symbols the language has, with no commitment yet to what they mean.

Example 2 (Some signatures)**.**

- *Pure equality*: $\sigma = (\emptyset, \emptyset, \emptyset)$. The language has no non-logical symbols at all, just variables, equality, and the logical apparatus. (Equality is treated as a built-in two-place relation, not part of $\mathcal{R}$.)
- *Preferences*: $\sigma_{\text{pref}} = (\emptyset, \emptyset, \{\succsim\})$, where $\succsim$ is a binary relation symbol. The intended interpretation is "at least as good as" on some set of alternatives.
- *Arithmetic*: $\sigma_{\text{arith}} = (\{0,1\}, \{+,\cdot\}, \{<\})$, with $0, 1$ constants, $+$ and $\cdot$ binary functions, $<$ a binary relation. This is the signature of standard number theory.
- *Voting*: $\sigma_{\text{vote}} = (\emptyset, \emptyset, \{P\})$, with P a ternary relation symbol; $P(v, c, c')$ is intended to mean "voter v prefers candidate c to candidate c'."

We assume an inexhaustible supply of *variables*, written $x, y, z, x_1, x_2, \ldots$, distinct from any of the symbols in σ.

Definition 3. The *terms* of σ are the smallest set such that:

- every variable is a term;
- every constant in $\mathcal{C}$ is a term;
- if $f \in \mathcal{F}$ has arity n and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.

Terms are the names we have available for individual elements of the (eventual) universe of discourse: pure variables (x), specific named elements (the constants), and combinations built up by function application ($x + 1$, $f(g(y), x)$, etc.).

Definition 4. The *formulas* of σ are the smallest set such that:

- if t_1, t_2 are terms, then $t_1 = t_2$ is a formula (an *atomic formula*, called an *equality*);
- if $R \in \mathcal{R}$ has arity n and $t_1, \ldots, t_n$ are terms, then $R(t_1, \ldots, t_n)$ is a formula (an *atomic formula*, called a *relation atom*);
- if φ, ψ are formulas, so are $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \to \psi$, $\varphi \leftrightarrow \psi$;
- if φ is a formula and x is a variable, then $\forall x\, \varphi$ and $\exists x\, \varphi$ are formulas.

Example 5. With σ_{pref}, the formula $\forall x\, \forall y\, (x \succsim y \vee y \succsim x)$ says that every two elements are comparable — the completeness axiom on a preference relation. The formula $\forall x\, \forall y\, \forall z\, ((x \succsim y \wedge y \succsim z) \to x \succsim z)$ says transitivity. With σ_{vote}, the formula $\forall v\, \forall c\, \forall c'\, (P(v, c, c') \to \neg P(v, c', c))$ says voter preferences are asymmetric.

A variable in a formula is *bound* if it is in the scope of a quantifier over that variable, and *free* otherwise. In $\forall x\, P(x, y, z)$, the variable x is bound, and y, z are free. A formula with no free variables is a *sentence*. Sentences are the formulas that have a definite truth value once we fix a structure to interpret them in; formulas with free variables only acquire truth values once those free variables are also assigned.

We adopt the standard quantifier-precedence convention: quantifiers extend their scope as far to the right as possible, so $\forall x\, P(x) \to Q(x)$ means $\forall x\, (P(x) \to Q(x))$, not $(\forall x\, P(x)) \to Q(x)$.[1]

[1] Two pieces of bookkeeping that we mention but do not belabor. First, *substitution*: when we substitute a term t

3 Structures and satisfaction

A signature is half the data. The other half is an *interpretation*: a non-empty set A (the universe or domain) together with an interpretation of each non-logical symbol in σ as a specific object of the appropriate kind — constants as elements of A, function symbols as functions, relation symbols as relations.

Definition 6. A σ*-structure* is a pair $\mathfrak{A} = (A, \sigma^{\mathfrak{A}})$ where:

- A is a non-empty set, the *universe* (or *domain*) of $\mathfrak{A}$;
- for each constant $c \in \mathcal{C}$, $c^{\mathfrak{A}} \in A$ is a designated element;
- for each function symbol $f \in \mathcal{F}$ of arity n, $f^{\mathfrak{A}} : A^n \to A$ is a function;
- for each relation symbol $R \in \mathcal{R}$ of arity n, $R^{\mathfrak{A}} \subseteq A^n$ is a subset (the set of tuples on which the relation holds).

A structure is the formal version of "writing down a model." Specifying a structure means saying what the universe is, what each named element is, what each function does, and what each relation tracks. This setup — a fixed signature, with a structure providing all the missing referents — is the foundation of model theory.

Example 7 (A preference structure)**.** With σ_{pref}, a structure is a pair $(X, \succsim^{\mathfrak{A}})$ where X is a set of alternatives and $\succsim^{\mathfrak{A}}$ is a binary relation on X. Whether the structure satisfies the completeness sentence $\forall x \forall y (x \succsim y \lor y \succsim x)$ is exactly the question whether $\succsim^{\mathfrak{A}}$ is a complete preorder. (The transitivity sentence is satisfied if and only if the relation is transitive, etc.) The structure-and-its-axioms picture is what political-economy modelers are doing whenever they say "a preference relation on X is a binary relation satisfying completeness, transitivity, and reflexivity."

Example 8 (A voting structure)**.** With σ_{vote}, a structure has a universe (the union of voters and candidates — one-sorted FOL forces them to share a domain) and an interpretation of P. The asymmetry sentence $\forall v \forall c \forall c' (P(v, c, c') \to \neg P(v, c', c))$ is then a literal mathematical claim about the structure; it is true or false depending on what triples are in $P^{\mathfrak{A}}$.

To define when a formula is true in a structure, we need to handle free variables. A *variable assignment* is a function $s : \mathit{Vars} \to A$ assigning each variable to an element of the universe. Given a structure $\mathfrak{A}$ and an assignment s, every term t has a value $t^{\mathfrak{A},s} \in A$, defined by recursion: $x^{\mathfrak{A},s} = s(x)$ for a variable; $c^{\mathfrak{A},s} = c^{\mathfrak{A}}$ for a constant; $f(t_1, \ldots, t_n)^{\mathfrak{A},s} = f^{\mathfrak{A}}(t_1^{\mathfrak{A},s}, \ldots, t_n^{\mathfrak{A},s})$.

Definition 9 (Tarski's definition of satisfaction)**.** For a σ-structure $\mathfrak{A}$, an assignment s, and a formula φ, we define $\mathfrak{A} \models \varphi[s]$ ("$\mathfrak{A}$ satisfies φ under s") by recursion on the structure of φ:

- $\mathfrak{A} \models (t_1 = t_2)[s]$ iff $t_1^{\mathfrak{A},s} = t_2^{\mathfrak{A},s}$.

for the free occurrences of x in φ, written $\varphi[t/x]$, we have to avoid *variable capture* — if t contains a variable y, and φ contains a quantifier $\forall y$ or $\exists y$ in whose scope x occurs free, the substitution would silently bind y, changing the meaning. The standard repair is to alpha-rename ($\forall y$ to $\forall y'$ for some fresh y') before substituting. Second, *closure*: a formula φ with free variables $x_1, \ldots, x_n$ has a *universal closure* $\forall x_1 \ldots \forall x_n \, \varphi$ and an *existential closure* similarly; for many purposes one works with closures so that the object under discussion is always a sentence. These are standard pieces of formal-logic hygiene; see Enderton (2001) or Dalen (2013) for the careful versions.

- $\mathfrak{A} \models R(t_1, \ldots, t_n)[s]$ iff $(t_1^{\mathfrak{A},s}, \ldots, t_n^{\mathfrak{A},s}) \in R^{\mathfrak{A}}$.
- $\mathfrak{A} \models (\neg\varphi)[s]$ iff not $\mathfrak{A} \models \varphi[s]$.
- $\mathfrak{A} \models (\varphi \wedge \psi)[s]$ iff $\mathfrak{A} \models \varphi[s]$ and $\mathfrak{A} \models \psi[s]$. (Similarly for $\vee, \to, \leftrightarrow$, following the propositional truth tables.)
- $\mathfrak{A} \models (\forall x\, \varphi)[s]$ iff for every $a \in A$, $\mathfrak{A} \models \varphi[s_{x\mapsto a}]$, where $s_{x\mapsto a}$ is the assignment that agrees with s except at x, where it equals a.
- $\mathfrak{A} \models (\exists x\, \varphi)[s]$ iff for some $a \in A$, $\mathfrak{A} \models \varphi[s_{x\mapsto a}]$.

For a sentence φ (no free variables), the assignment s does not matter, and we just write $\mathfrak{A} \models \varphi$. The structure-and-satisfaction apparatus is what gives content to the syntactic machinery: the formula $\forall x \exists y\, x \succsim y$ is a string of symbols until we have a structure, but in any specific structure it is a definite mathematical claim about the relation $\succsim^{\mathfrak{A}}$.

The relations $\models$ (semantic consequence) and $\vdash$ (syntactic consequence) extend to first-order logic in the natural way: $\Gamma \models \varphi$ holds when every structure satisfying all of Γ also satisfies φ, and $\Gamma \vdash \varphi$ holds when there is a derivation of φ from Γ in a first-order proof system (extending the natural-deduction rules of the previous handout with rules for $\forall$ and $\exists$). Gödel's completeness theorem (1930) says these two relations coincide: $\Gamma \vdash \varphi$ if and only if $\Gamma \models \varphi$.[2]

4 Many-sorted first-order logic

In one-sorted FOL the universe is a single set A, and every variable ranges over the whole of it. For some applications this is fine: a theory of natural numbers has one kind of object (numbers), and "$\forall x$" should range over all of them. For many other applications it is cumbersome. A voting model has at least two kinds of object (voters and candidates), and "$\forall x$" ranging over the merger of both is the wrong picture — we want voters to range over voters and candidates over candidates, with the language preventing nonsense like "the candidate's preference over the candidate."

The traditional fix in one-sorted FOL is to add a unary predicate for each kind — Voter(x) and Candidate(x) — and to relativize every quantifier: $\forall x\,(\text{Voter}(x) \to \ldots)$ instead of $\forall x\,(\ldots)$. This works but is awkward and clutters the formulas. *Many-sorted first-order logic* (MSFOL) cleans it up by building the typing into the language.

Definition 10. A *many-sorted signature* is a tuple $\sigma = (S, \mathcal{C}, \mathcal{F}, \mathcal{R})$ where:

- S is a non-empty set of *sorts* (or *types*);

[2] The first-order completeness theorem is often confused with Gödel's *incompleteness* theorems, which are different and stronger and are about specific theories rather than about FOL itself. The completeness theorem says that the proof system for first-order logic is sufficient to derive every semantic consequence; the incompleteness theorems (1931) say that any sufficiently expressive recursively axiomatizable theory of arithmetic — meaning a theory whose axioms can be listed by an algorithm and which can express enough number theory to encode its own syntax — has true sentences (in the standard model of arithmetic) that the theory itself cannot derive. The first-order language is complete; specific first-order theories of arithmetic, by contrast, are incomplete in this technical sense. The two results are compatible: completeness of FOL says "the formal system captures all valid first-order consequences"; incompleteness of arithmetic says "no first-order axiomatization of arithmetic captures all true arithmetic statements." For political-economy work the moral is that some questions one might want a model to settle (e.g., decidability of a complex equilibrium condition) may turn out to be undecidable on cardinality or computability grounds — a fact one bumps into very rarely in practice, but which is worth knowing exists. Enderton (2001) treats both theorems carefully.

- each constant $c \in \mathcal{C}$ has an associated sort $\text{sort}(c) \in S$;
- each function symbol $f \in \mathcal{F}$ has an associated *sort signature* $(s_1, \ldots, s_n) \to s$, with $s_i, s \in S$, indicating the sorts of the arguments and the output;
- each relation symbol $R \in \mathcal{R}$ has an associated sort signature $(s_1, \ldots, s_n)$.

Variables are sorted: each variable x is associated with a sort $\text{sort}(x) \in S$, written $x : s$ when we want to make the sort explicit.

Definition 11. A *many-sorted σ-structure* $\mathfrak{A}$ assigns:

- to each sort $s \in S$, a non-empty set A_s (the *carrier* for sort s);
- to each constant c of sort s, an element $c^{\mathfrak{A}} \in A_s$;
- to each function symbol $f : (s_1, \ldots, s_n) \to s$, a function $f^{\mathfrak{A}} : A_{s_1} \times \cdots \times A_{s_n} \to A_s$;
- to each relation symbol $R : (s_1, \ldots, s_n)$, a subset $R^{\mathfrak{A}} \subseteq A_{s_1} \times \cdots \times A_{s_n}$.

The well-formedness rules for terms and formulas track sorts: a function $f : (s_1, \ldots, s_n) \to s$ can only be applied to argument terms of sorts $s_1, \ldots, s_n$ respectively, and the result has sort s. The quantifier $\forall x\, \varphi$ (with $x : s$) ranges over A_s only.

Example 12 (A two-sorted voting signature)**.** Let $S = \{V, C\}$ (voters and candidates), $\mathcal{C} = \emptyset$, $\mathcal{F} = \emptyset$, and $\mathcal{R} = \{P\}$ with P of sort signature (V, C, C). The intended reading is $P(v, c, c')$: "voter v prefers candidate c to candidate c'."

A structure $\mathfrak{A}$ has carriers A_V (a set of voters) and A_C (a set of candidates), and $P^{\mathfrak{A}} \subseteq A_V \times A_C \times A_C$. The asymmetry condition becomes

$$\forall v : V\, \forall c, c' : C\ (P(v, c, c') \to \neg P(v, c', c)),$$

in which the sorting of variables is part of the syntax rather than something we have to relativize against in the body of the formula. The condition that voter preferences over candidates are total can be written

$$\forall v : V\, \forall c, c' : C\ (c = c' \lor P(v, c, c') \lor P(v, c', c)),$$

again with no need for Voter or Candidate predicates.

Example 13 (A signature for cooperative-game environments)**.** Let $S = \{N, R\}$ (players and real numbers), $\mathcal{C} = \emptyset$, $\mathcal{F} = \{v\}$ with v of sort signature $(\mathcal{P}(N) \to R)$ — where $\mathcal{P}(N)$ is shorthand for "a sort whose elements are subsets of the player sort." (Strictly speaking $\mathcal{P}(N)$ is a higher-type construction beyond pure first-order logic, but with multiple sorts we can declare it as a separate sort with an additional "element" relation $\in$ between sort N and sort $\mathcal{P}(N)$.) The function $v : \mathcal{P}(N) \to R$ is a coalitional value function; the standard cooperative-game axioms (superadditivity, monotonicity, etc.) become first-order sentences in this signature.

MSFOL is, on the surface, "just" a notational reorganization: every MSFOL theory can be translated into a one-sorted FOL theory by introducing a unary sort predicate for each sort and relativizing all quantifiers. The translation preserves derivability and satisfiability, so completeness and soundness for MSFOL are corollaries of the one-sorted versions. But the reorganization is substantive in practice. It mirrors how applied mathematicians and modelers actually think (in terms of typed data: a player, a strategy, a payoff, a state); it makes formulas dramatically more readable; and it bridges to the working idiom of formal-economic and political modeling, which is essentially many-sorted from the start.

The model-theoretic reason to develop MSFOL alongside one-sorted FOL is that the next handout's discussion of *theories as classes of structures* naturally takes "a structure" to mean a tuple of carriers plus interpretations — the MSFOL version. When a political economist writes "the model has a set of agents N, a set of states Ω, a payoff function $u_i : \prod_j A_j \to \mathbb{R}$ for each player," they are exhibiting a many-sorted structure, not a one-sorted one.

5 What's next

The next handout, *Model theory and the semantic view*, picks up the signature-and-structure machinery developed here and extends it in two directions:

- *Theories as classes of models.* A theory in this handout is a set of sentences over some signature; the natural object associated with a theory is the class of all structures that satisfy it. Many basic notions of model theory — elementary equivalence, isomorphism, definability, categoricity — are statements about classes of structures rather than about syntactic theories per se.
- *The semantic view of theories.* Patrick Suppes, Bas van Fraassen, and others argued in the philosophy-of-science literature that scientific theories are best understood as classes of models rather than as syntactic axiomatizations — the so-called "semantic view" or "model-theoretic view" of theories. This perspective fits very naturally with how political scientists and economists already use the word "model"; it is worth a careful look, because once one sees it, the relationship between formal logic and applied modeling stops looking like an analogy and starts looking like an identity.

The proof-theoretic side of FOL — Gödel's completeness theorem, compactness, the Löwenheim–Skolem theorems — is a fourth direction we mention but do not develop. The pattern is the same as in the proof-systems handout: a canonical-model construction (Henkin's, due to 1949) shows that every consistent set of first-order sentences has a model, and from that the rest of the basic apparatus follows. Enderton (2001) or Dalen (2013) gives a full treatment.

6 Exercises

Exercise 14 (Translating axioms)**.** Write each of the following as a first-order sentence over an appropriate signature. State the signature in each case.

1. "The relation $\succsim$ is reflexive."
2. "The relation $\succsim$ is transitive."
3. "The relation $\succsim$ is complete: for any two alternatives, one is at least as good as the other."

4. "There exists a maximum: some alternative is at least as good as every alternative."

Exercise 15 (Free vs. bound)**.** For each formula below, identify the free and bound variables, and decide whether the formula is a sentence:

1. $\forall x\, P(x, y, z)$
2. $\forall x\, \exists y\, P(x, y, z)$
3. $(\forall x\, R(x, y)) \to \exists y\, R(x, y)$
4. $\forall x\, (\exists y\, R(x, y) \to S(x))$

Exercise 16. Let σ_{pref} have a binary relation symbol $\succsim$. Write a sentence saying "the strict part $\succ$ of $\succsim$ is asymmetric." (Define $x \succ y$ as a defined formula in terms of $\succsim$, and use the definition.)

Exercise 17 (Translating $\exists!$)**.** Write a sentence saying "there exists a *unique* x such that $P(x)$." (You will need to use both $\exists$ and $\forall$, plus equality.)

Exercise 18 (A voting structure)**.** Take σ_{vote} with the ternary relation $P(v, c, c')$. Let $\mathfrak{A}$ have universe $A = \{a, b, c, d\}$ where a and b are voters and c, d are candidates, with $P^{\mathfrak{A}} = \{(a, c, d), (b, d, c)\}$. (Voter a prefers c to d; voter b prefers d to c.) Decide whether $\mathfrak{A}$ satisfies each of:

1. $\exists v \exists c \exists c'\, P(v, c, c')$.
2. $\forall v\, \exists c\, \exists c'\, P(v, c, c')$. (Why does this need care given that the universe is one-sorted?)
3. $\forall v \forall c \forall c'\, (P(v, c, c') \to \neg P(v, c', c))$.
4. $\exists v \forall c \forall c'\, P(v, c, c')$.

Exercise 19 (Many-sorted refinement)**.** Repeat the previous exercise with the two-sorted signature $(S = \{V, C\},\ P : (V, C, C))$ and the natural many-sorted structure: $A_V = \{a, b\}$, $A_C = \{c, d\}$, $P^{\mathfrak{A}} = \{(a, c, d), (b, d, c)\}$. Compare the formulas you get for "some voter prefers some candidate to some other." Where does the many-sorted formulation simplify, and where does the one-sorted formulation force extra bookkeeping?

Exercise 20. Take the signature $\sigma_{\text{arith}} = (\{0, 1\}, \{+, \cdot\}, \{<\})$ and the standard structure $(\mathbb{N}, +, \cdot, <, 0, 1)$. Write a first-order sentence in σ_{arith} that is true in this structure and expresses "every nonzero element has a successor." Write another that expresses "addition is commutative."

Exercise 21 (A coalitional game in MSFOL)**.** Specify a many-sorted signature appropriate for a transferable-utility cooperative game with finitely many players. (Sorts: players, coalitions, real numbers. Symbols: a function v from coalitions to reals; element-of relation between players and coalitions.) Then write the *superadditivity* axiom as a sentence over this signature. (Recall: $S \cap T = \emptyset$ implies $v(S \cup T) \geq v(S) + v(T)$.)

Exercise 22. Show that the universal closure of a tautology (in the propositional-logic sense, with atoms replaced by atomic formulas) is satisfied in every structure under every assignment. (Hint: the propositional tautology is true under every truth assignment to its atoms; pull this through Tarski's recursion.)

Exercise 23 (Failure of the converse)**.** Find a sentence φ in σ_{pref} that holds in some preference structures but not all. (Many examples will work; the point is to exhibit how the satisfaction relation is structure-dependent — sentences are not uniformly true the way propositional tautologies are.)

References

Dalen, Dirk van (2013). *Logic and Structure*. 5th ed. London: Springer.

Enderton, Herbert B. (2001). *A Mathematical Introduction to Logic*. 2nd ed. San Diego: Academic Press.