ITCS 531: Logic 5 - Basic model theory

Rob Egrot

Structures and languages

- In the last class we introduced the idea of a first-order language $\mathscr{L}$.
- A mathematical structure is an $\mathscr{L}$-structure if it has relations, functions and constants corresponding to the non-logical symbols of $\mathscr{L}$.
- If ϕ is an $\mathscr{L}$-sentence and A is an $\mathscr{L}$-structure, if ϕ is true in A we say A is a *model* for ϕ (we write $A \models \phi$).
- Given a set of $\mathscr{L}$-sentences, we can try to understand them by studying its models.
- Conversely, given a structure A we can try to think of it as an $\mathscr{L}$-structure for some $\mathscr{L}$.
- We can study A by looking at the $\mathscr{L}$-sentences that it models.

Model theory

- This two-way study of logic through structures, and structures through logic, is the starting point of *model theory*.
- In this class we will look more closely at the concept of models and semantic implication.
- We will also define a concept of deduction based on that for propositional logic.
- We will state soundness and completeness theorems for first-order logic.
- We will see some limitations of first-order logic for studying infinite structures.

Logical implication

- If Γ is a set of $\mathscr{L}$-formulas, and if ϕ is an $\mathscr{L}$-formula, we write $\Gamma \models \phi$ if, whenever v is an assignment of the variables of $\mathscr{L}$ into an $\mathscr{L}$-structure A, we have

$$A, v \models \Gamma \implies A, v \models \phi.$$

- We say that ϕ is a *logical consequence* of Γ.
- Remember *sentences* are formulas that have no free variables.
- For sentences we can just write, e.g. $A \models \phi$.
- When $A \models \phi$, we say A is a *model* for ϕ.
- If Δ is a set of $\mathscr{L}$-sentences we can write e.g. $A \models \Delta$ when $A \models \phi$ for all $\phi \in \Delta$, and say A is a model for Δ.

Validity and satisfiability

Definition 1

If ϕ is an $\mathscr{L}$-formula then we say ϕ is:

- *Valid* if $A, v \models \phi$ whenever A is an $\mathscr{L}$-structure and v is an assignment.
- *Satisfiable* if there is an $\mathscr{L}$-structure A and an assignment v with $A, v \models \phi$.
- A *contradiction* if it is not satisfiable, i.e. if there is no A, v with $A, v \models \phi$.

Similarly, if Γ is a set of $\mathscr{L}$-formulas then Γ is:

- *Valid* if $A, v \models \Gamma$ whenever A is an $\mathscr{L}$-structure and v is an assignment.
- *Satisfiable* if there is an $\mathscr{L}$-structure A and an assignment v with $A, v \models \Gamma$.
- *Contradictory* if it is not satisfiable, i.e. if there is no A, v with $A, v \models \Gamma$. If Γ is not satisfiable we write $\Gamma \models \bot$.

Example - arithmetic

Example 2

Let $\mathscr{L} = \{0, 1, \times, +\}$ be the language of arithmetic.

1. Let $\phi = \forall x\big((x \approx 0) \vee \neg(x \approx 0)\big)$. Then ϕ is valid. More generally, if $\mathscr{L}$ is a language, and if $\phi_1, \dots, \phi_n$ are $\mathscr{L}$-sentences, then any propositional tautology constructed by treating the ϕ_i as basic propositions will be valid.

2. Let $\psi = \forall x(\neg(x \approx 0) \to \exists y(x \times y \approx 1))$. This is true if we take $\mathbb{R}$ as our structure, but not if we take $\mathbb{Z}$. So ψ is satisfiable but not valid.

3. If $\phi_1, \dots, \phi_n$ are $\mathscr{L}$-sentences, then any propositional contradiction using the ϕ_i as basic propositions will be a contradiction.

Checking satisfiability

Definition 3 (Theory)

If $\mathscr{L}$ is a language, then an $\mathscr{L}$-*theory* is a satisfiable set of $\mathscr{L}$-sentences.

- Checking logical consequence, validity etc. is much more complicated for first-order logic than for propositional logic.
- In propositional logic, you just construct a truth table.
- In first-order logic, you might have to check every possible $\mathscr{L}$-structure.
- Generally this is not feasible.
- There is no algorithm for saying whether an arbitrary $\mathscr{L}$-sentence has a model (proved next semester).

Intended models

- When we write down axioms in first-order logic, there is often some particular system whose behaviour we are trying to formalize.
- E.g. we might write down axioms for defining real numbers.
- The intended model here is $\mathbb{R}$, and we can choose axioms so that $\mathbb{R}$ is indeed a model.
- But can we choose first-order axioms so that $\mathbb{R}$ is the only model?
- No. In fact, it is impossible to use first-order logic to define a specific infinite structure.
- This is due to the theorem on the next slide.

The Löwenheim-Skolem theorem

Theorem 4

Let Γ be a countable $\mathscr{L}$-theory. Then, if Γ has an infinite model, it has models of every infinite cardinality.

- Theorem 4 gives us an infinite supply of extra models for any theory that has at least one infinite model.
- It also tells us that first-order logic can't 'tell the difference' between different infinite cardinalities.
- Unintended models need not have different cardinalities though, as the example on the next slide illustrates.

Unnatural natural numbers

Example 5

Let $\mathscr{L} = \{0, s\}$, where s is a unary function. Let Γ consist of the following sentences.

ϕ_1: $\forall x(\neg(x \approx 0) \to \exists y(x = s(y))$.

ϕ_2: $\forall x(\neg(x \approx s(x)))$.

ϕ_3: $\forall x \forall y((s(x) \approx s(y)) \to (x \approx y))$.

- One model of Γ is $\mathbb{N}$, where s is interpreted as the 'successor' function.
- Is $\mathbb{N}$ the only model?
- No, for example, the disjoint union of $\mathbb{N}$ and $\mathbb{Z}$ is also a model if we interpret 0 as the zero of $\mathbb{N}$, and s as the successor function in both $\mathbb{N}$ and $\mathbb{Z}$.

Technical aside - substitution

- Let ϕ be an $\mathscr{L}$-formula with free variables $x_1, \dots x_n$.
- We can express this fact by writing $\phi[x_1, \dots, x_n]$.
- Let t be an $\mathscr{L}$-term, and let $i \in \{1, \dots, n\}$.
- We can create a new formula from ϕ by replacing every occurrence of the variable x_i with the term t.
- We use the notation $\phi[x_1, \dots, x_{i-1}, t/x_i, x_{i+1}, \dots, x_n]$ to denote this new formula.
- E.g. let $\phi[x, y] = s(x) \approx y$, let $t = s(s(z))$.
- Substituting t for x gives in ϕ gives $\phi[t/x, y] = s(s(s(z))) \approx y$.
- Note that $\phi[x, y]$ and $\phi[t/x, y]$ have different free variables.

Deduction in first-order logic

- We can extend the natural deduction system for propositional logic to first-order logic.
- We have all the same deduction rules as before (but with first-order formulas in place of propositional sentences).
- We also have extra ones for $\approx$ and the quantifiers $\forall$ and $\exists$.

Inroduction rules

$$\approx_I: \quad \frac{}{t \approx t}$$

$$\forall_I: \quad \frac{\phi[x'/x]}{\forall x \phi}$$

$$\exists_I: \quad \frac{\phi[t/x]}{\exists x \phi}$$

Elimination rules

$$\approx_E: \quad \frac{t_1 \approx t_2 \qquad \phi[t_1/z]}{\phi[t_2/z]}$$

$$\forall_E: \quad \frac{\forall x \phi}{\phi[t/x]}$$

$$\exists_E: \quad \frac{\exists x \phi \qquad \begin{array}{c}[\phi[x'/x]] \\ \hline\hline \psi\end{array}}{\psi}$$

First-order deduction example

Example 6

Let ϕ and ψ be formulas where x occurs free. Then we can deduce $\forall x \psi$ from $\forall x \neg\phi$ and $\forall x(\phi \vee \psi)$.

$$\dfrac{(\forall_E)\,\dfrac{\forall x \neg \phi}{\neg\phi[x'/x]} \qquad (\forall_E)\,\dfrac{\forall x(\phi \vee \psi)}{\phi[x'/x] \vee \psi[x'/x]}}{(\forall_I)\,\dfrac{\psi[x'/x]}{\forall x \psi}}\ \text{(p.d.)}$$

p.d. stands for *propositional deduction*.

Another first-order deduction example

Example 7

Let ϕ and ψ be formulas where x occurs free. Then we can deduce $\exists x\psi$ from $\exists x\neg\phi$ and $\forall x(\phi \vee \psi)$.

$$
(\text{p.d.})\;\dfrac{
(\exists_E)\,\dfrac{\exists x\neg\phi \qquad \dfrac{[\neg\phi[x'/x]]}{\neg\phi[x'/x]}}{\neg\phi[x'/x]}
\qquad\qquad
\dfrac{\forall x(\phi\vee\psi)}{\phi[x'/x]\vee\psi[x'/x]}\,(\forall_E)
}{ (\exists_I)\,\dfrac{\psi[x'/x]}{\exists x\psi} }
$$

Consistency

- As with propositional logic we write $\Gamma \vdash \phi$ if ϕ can be deduced from a set of formulas Γ.
- We say a set of $\mathscr{L}$-sentences, Γ, is **consistent** if we do not have $\Gamma \vdash \bot$.
- We sometimes describe a consistent set of $\mathscr{L}$-sentences as an $\mathscr{L}$-*theory*.
- This is consistent with definition 3 because, as in propositional logic, there is a strong link between $\vdash$ and $\models$ (see next slide).

Soundness and completeness

Theorem 8 (Gödel)

Let Γ be a set of $\mathscr{L}$-formulas. Then Γ is consistent if and only if it is satisfiable.

Theorem 9 (Extended soundness and completeness)

Let Γ be a set of $\mathscr{L}$-formulas and let ϕ be an $\mathscr{L}$-formula. Then

$$\Gamma \vdash \phi \iff \Gamma \models \phi.$$

These two results are equivalent (this is proved in the exercises this week).

Proving theorem 9 - Soundness

- Like the soundness theorem for propositional logic.
- Must show each deduction rule is sound.
- E.g.

  $\wedge_I$:
  - Have deduced ϕ and ψ from Γ, and from these have deduced $\phi \wedge \psi$.
  - Assuming deductions of ϕ and ψ are sound, any pair (A, v) satisfying Γ must satisfy ϕ and ψ.
  - So also satisfies $\phi \wedge \psi$.

  - The new rules are trickier, but as a sketch:

  $\forall_I$:
  - Have deduced $\phi[x'/x]$ for arbitrary choice of x'.
  - Assuming this deduction is sound, any pair (A, v) satisfying Γ will also satisfy $\phi[x'/x]$
  - Must show that $A, v \models \forall x \phi[x]$ too.
  - I.e. if v' agrees with v except possibly about x then $A, v' \models \phi[x]$.
  - With a little fiddling we can do this (see notes).

Proving theorem 9 - Completeness

- Completeness is harder, but conceptually similar to the propositional version.
- Again, proving completeness is equivalent to proving that every consistent set of formulas is satisfiable.
- Rather than just building a true/false assignment that satisfies a consistent set of propositional sentences, we must find a pair (A, v) satisfying a set of first-order formulas.
- It is possible to do this, using the formulas themselves as the base for a model.