INTRODUCTION TO MODEL THEORY: FORMAL METHODS SEMINAR

What is Model Theory?

In model theory we study the objects that satisfy formal theories. We do this by providing interpretations of the names, predicates and parts of the language that the sentences of that theory are written in.

In addition to the logical connectives, equality symbol and quantifiers familar from introductory logic courses, the languages we use in model theory generally contain constant symbols c_i, relation symbols R_j and function symbols f_k. These are called the nonlogical symbols. We provide an interpretation of these symbols over domain $\mathcal{D}$ in the following way:[1]

- We assign objects of the domain to constant symbols. The object assigned to c_i in model $\mathcal{A}$ is $c_i^{\mathcal{A}} \in \mathcal{D}$.
- We assign subsets of n-tuples of the domain to n-ary relation symbols. The set assigned to R_j is $R_j^{\mathcal{A}} \in \mathcal{P}(\mathcal{D}^n)$. Predicate symbols are 1-place relation symbols.
- We assign m-ary functions on the domain to m-ary function symbols. The function assigned to f_k is $f_k^{\mathcal{A}} : \mathcal{D}^m \to \mathcal{D}$.

Definition 1. $\mathcal{L}$-structure

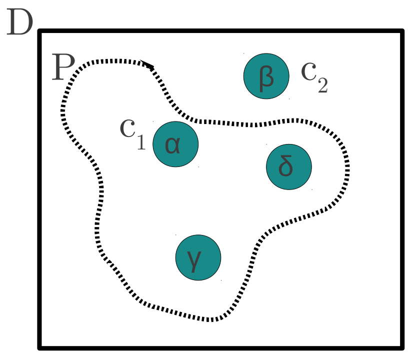
A domain with an interpretation of language $\mathcal{L}$ is called an $\mathcal{L}$-structure. An $\mathcal{L}$-structure that makes a theory T true is called a *model* of T.

Example 2. We could interpret a language containing the 2-place relation symbol R over a a domain consisting of books on a shelf in such a way that the symbol, R, picks out the relation of being next to, so that Rxy means 'x is next to y'. Then, the relation symbol is interpreted to be all those pairs of books that lie next to one another. The language might not have names for all or any of the books, but that is no obstacle to defining the next-to relation, for the relation is simply the pairs of books themselves, not any description of them.

Example 3. For a more abstract example, suppose that the domain consists of the objects α, β, γ, δ, pictured below, and that the language has two constant symbols, c_1 and c_2, and one relation symbol, P.

Date: 2nd February 2012.

[1] In any particular model it may happen that not every kind of symbol appears in the theory. To give a model of theory T, we only interpret symbols that appear in T.



P, c_1 and c_2 are all parts of the language; we write this as $\mathcal{L} = \langle c_1, c_2, P\rangle$. To define the $\mathcal{L}$-structure, then, in addition to the domain $\mathcal{D} := \{\alpha, \beta, \gamma, \delta\}$ we need an interpretation of $\mathcal{L}$, which we shall now give.

(i) $P^{\mathcal{A}} = \{\alpha, \gamma, \delta\}$
(ii) $c_1^{\mathcal{A}} = \alpha$
(iii) $c_2^{\mathcal{A}} = \beta$

N.b. there is a strict language/meta-language distinction here: the names 'α', 'β' and so on are labels that we use to refer to the objects of the domain, but they do not feature in the language $\mathcal{L}$. So although we've interpreted P to be satisfied by the object labelled 'δ', we can't express this fact in $\mathcal{L}$. We can, however, say things like 'exactly three things are satisfied by P'. The interpreted constant symbols act as names, and allow us to say in $\mathcal{L}$ that c_1 is P, and c_2 is not P.

This abstract example more closely represents the work of model-theorists, who use set domains comprised of sets or *urelemente* rather than concrete objects. However, a grasp on the interpretation relation and the language/meta-language distinction is useful for discussing more general philosophical issues, such as representation. It is also essential for understanding the difference between syntax and semantics of formal languages.

Satisfaction

The interpretation of the $\mathcal{L}$-structure lets us assign truth-values to formulas in the language $\mathcal{L}$. In model theory we do this in the following way, specified recursively:

(A) Interpretation.
Suppose that $\mathcal{A}$ is an $\mathcal{L}$-structure with domain $\mathcal{D}$.

Definition 4. *Terms*, $\tau(v_1, \ldots, v_n)$, of $\mathcal{L}$ are the basic building blocks of the language; they're anything that we can get by building up from constants and variables using functions. In natural language, terms correspond to singular noun phrases. Formally, they are any of the following:

(i) a variable v_j
(ii) a constant symbol c
(iii) $f(\tau_1, \ldots, \tau_n)$; the result of applying function f to terms $\tau_i : 1 \leq i \leq n$.

Each of the closed terms (the ones without free variables) is assigned an object from the domain. Constant symbols are interpreted as single elements, and likewise the image of a function on those elements is assigned an element according to the interpretation of f.

(B) Assigning truth-values to $\mathcal{L}$-formulas in $\mathcal{L}$-structures.

As before, let $\mathcal{A}$ be an $\mathcal{L}$-structure with domain $\mathcal{D}$. Let $\varphi(v_1, \ldots, v_n)$ an $\mathcal{L}$-formula with free variables $v_1, \ldots, v_n$ and $\overline{a} = \langle a_1, \ldots a_n \rangle \in \mathcal{D}^n$. Then, we say that φ is *true in* $\mathcal{A}$, and write $\mathcal{A} \vDash \varphi(\overline{a})$ in accordance with the following rules:

(i) $\mathcal{A} \vDash \tau_1(\overline{a}) = \tau_2(\overline{a})$ iff $\tau_1^{\mathcal{A}}(\overline{a}) = \tau_2^{\mathcal{A}}(\overline{a})$;
(ii) $\mathcal{A} \vDash P_i(\tau_1(\overline{a}), ..., \tau_r(\overline{a}))$ iff $\left\langle \tau_1^{\mathcal{A}}(\overline{a}), ..., \tau_r^{\mathcal{A}}(\overline{a}) \right\rangle \in P_i^{\mathcal{A}}$;
(iii) $\mathcal{A} \vDash \varphi_1(\overline{a}) \bigwedge \varphi_2(\overline{a})$ iff $\mathcal{A} \vDash \varphi_1(\overline{a})$ and $\mathcal{A} \vDash \varphi_2(\overline{a})$;
(iv) $\mathcal{A} \vDash \neg\varphi_1(\overline{a})$ iff $\mathcal{A} \nvDash \varphi_1(\overline{a})$;
(v) $\mathcal{A} \vDash \exists v_n \varphi(a_1, \ldots, a_{n-1}, v_n)$ iff there is an $a_n \in \mathcal{D}$ such that $\mathcal{A} \vDash \varphi(a_1, \ldots, a_{n-1}, a_n)$.

Definition 5. Sentence

A *sentence* of $\mathcal{L}$ is a formula with no free (unbound) variables.

Definition 6. Theory

A the*ory*, Γ in a formal language $\mathcal{L}$ is a consistent set of $\mathcal{L}$-sentences.

Rather than proving theorems of a formal theory, in model theory we look at the features of its models. Model theory is concerned with semantic consequence, rather than syntanctic consequence – derivability in a particular formal system.

Definition 7. Semantic consequence

ϕ is a semantic, or *logical, consequence* of theory Γ if every $\mathcal{L}$-structure that makes Γ true also makes ϕ true. We write $\Gamma \vDash \phi$.

Definition 8. Logically valid

We say that ϕ is *logically valid*, and write $\vDash \phi$ if every $\mathcal{L}$-structure makes ϕ true.

Definition 9. Satisfiable

We say that a sentence ϕ is *satisfiable* if there is some $\mathcal{L}$-structure that makes ϕ true.

Definition 10. Unsatisfiable

A sentence ϕ is *unsatisfiable* if there is no $\mathcal{L}$-structure that makes it true (if it is not valid).

Definition 11. Invalid

A sentence is *invalid* if there is some $\mathcal{L}$-structure that makes it false (its negation is satisfiable).

Definitions 6-10 are also correct when we replace ‘sentence’ by ‘theory’.

The Completeness Theorem

The completeness theorem for a particular proof procedure is the claim that the procedure is semantically complete. This means that whenever a sentence ϕ is a semantic consequence of a set of sentences Γ, then ϕ can be derived deductively from Γ: in symbols, if $\Gamma \vDash \phi$ then $\Gamma \vdash \phi$. The theorem is the converse to the soundness theorem for first order logic, which states that $\Gamma \vdash \phi$ implies $\Gamma \vDash \phi$.

Theorem 12. *Gödel's completeness theorem*

For any first-order theory Γ and any sentence ϕ in the language of the theory, if $\Gamma \vDash \phi$ then there is a formal deduction of ϕ from Γ.

Corollary 13. *A generalisation of Gödel's completeness theorem*
If a sentence ϕ is logically valid then there is a finite deduction of ϕ.

Proof. Omitted. (But see Barwise [1, p.22 ff.].) □

Corollary 14. *If a set of first-order sentences Γ is consistent then Γ has a model.*

Proof. Let ϕ be any inconsistent sentence $\perp$. □

(1) If Γ is unsatisfiable then $\Gamma \vDash \perp$ (since $\Gamma \vDash \perp$ means that any model of Γ must be a model of $\perp$, and $\perp$ is a sentence that has no models given our definition of satisfaction).
(2) By Theorem 12, if $\Gamma \models \perp$ then Γ proves $\perp$.
(3) If Γ proves$\perp$ then Γ is inconsistent, by definition.
(4) So, if Γ is unsatisfiable then it is inconsistent. (By 1, 2, 3)
(5) By contraposition, if Γ is consistent then it is satisfiable.

References

[1] Barwise, *Handbook of Mathematical Logic*, North-Holland Publishing Company (1977)

[2] Boolos et al., *Computability and Logic*, 5th Ed., Cambridge University Press (2007)

E-mail address: `Kate.Hodesdon@bristol.ac.uk`