Ma 116a

Me: Garrett Ervin
gervin@caltech.edu
office: 204 Kellogg
OH: Tu/Th 2:30 – 4:00 (for now)

TA: Michael Wolman
mwolman@caltech.edu
OH: TBD

Evaluation

- ~ 8 HW sets
- first due: Oct. 10th
- submit via Gradescope
- one 48 hour extension no Q's asked (email me)
- o/w late HW not accepted
- grading cutoffs: not worse than standard

The class

↳ intro to model theory
↳ topics include: first-order logic basics, compactness theorem!, Löwenheim-Skolem theorem, Fraïssé theory, quantifier elim'n, omitting types

Textbook

Marker "Model Theory: An Introduction"

↳ cover first 3+ chapters

First-order languages + structures

- Intuitively, a (first-order) structure is a set equipped w/ distinguished functions, relations, constants

- e.g. consider $\mathbb{Q}$
  - ↳ could study as a "pure set" (no distinguished f's/R's/c's)
  - ↳ or as an additive group
    $(\mathbb{Q}, +, 0)$ — distinguished constant (identity); $+$ — distinguished binary function (group operation)
  - ↳ or as a linear order
    $(\mathbb{Q}, <)$ — binary relation
  - ↳ or as a ring with 1
    $(\mathbb{Q}, +, \times, 0, 1)$

- the f's/R's/c's we distinguish determine the formal statements we can write about our structure

- e.g. over $(\mathbb{Q}, +, 0)$ can assert
  $\forall x \exists y (x+y=0)$
  which is true in this struct.

- over $(\mathbb{Q}, <)$ can assert
  $\exists x (x<x)$
  which is false " "
  ↳ could not have asserted this of $(\mathbb{Q}, +, 0)$ which isn't equipped w/ $<$.

- to formalize the above, need to define what a (first-order) language is

Def'n A language $\mathcal{L}$ consists of
① a set of function symbols $\mathcal{F}$ and for each symbol $f \in \mathcal{F}$, an arity $n_f \in \mathbb{N}$
② " " relation symbols $\mathcal{R}$
" " $R \in \mathcal{R}$, an arity $n_R \in \mathbb{N}$.
③ " " constant symbols $\mathcal{C}$

- we write $\mathcal{L} = \mathcal{F} \cup \mathcal{R} \cup \mathcal{C}$
- note: a lang. $\mathcal{L}$ is just a collection of symbols + associated arities
- will interpret symbols as actual f's/R's/c's, but the same symbol can be interpreted in diff. ways over diff. structures.

Ex's – could consider $\mathcal{L}_0 = \emptyset$, the empty language ("lang of pure sets")
– or $\mathcal{L}_1 = \{f, c\}$ w/ f a binary function symbol, c a constant symbol ("lang of groups")
– or $\mathcal{L}_2 = \{R\}$ w/ single ~~binary~~ binary rel'n symbol ("lang of orders/equiv. relations ...")

~~Also~~ Again: language = symbols + arities only.

↳ sometimes will have in mind an intended interpretation of the symbols in $\mathcal{L}$
↳ e.g. could say "let $\mathcal{L} = \{+, 0\}$" where I mean $+$ is a binary ~~fcn~~ fcn symbol I intend to interpret as the actual function $+$, 0 is a constant symbol I intend to interpret as 0
↳ is an abuse of notation

Def'n Given a lang. $\mathcal{L}$ an $\mathcal{L}$-structure $\mathcal{M}$ consists of:
1. a nonempty set M ("universe" of $\mathcal{M}$)
2. for each $f \in \mathcal{F}$, ~~a function~~ a function $f^{\mathcal{M}}: M^{n_f} \to M$
3. for each $R \in \mathcal{R}$, a relation $R^{\mathcal{M}} \subseteq M^{n_R}$

4. for each $c \in \mathcal{C}$, an element $c^{\mathcal{M}} \in M$.

- the objects $f^{\mathcal{M}}, R^{\mathcal{M}}, c^{\mathcal{M}}$ are the <u>interpretations</u> of the symbols f, R, c.

- will write $\mathcal{M} = (M, f^{\mathcal{M}}, g^{\mathcal{M}}, \ldots R^{\mathcal{M}}, S^{\mathcal{M}}, \ldots c^{\mathcal{M}}, d^{\mathcal{M}}, \ldots)$

<u>Ex's</u> ① for $\mathcal{L}_0 = \emptyset$, an $\mathcal{L}_0$-structure $\mathcal{M}$ is just a set M!

② Consider $\mathcal{L}_1 = \{f, c\}$

- if G is a group w/ operation $\cdot$ and identity e, can get an $\mathcal{L}_1$-structure $\mathcal{G} = (G, \cdot, e)$ w/ universe G and $f^{\mathcal{G}} = \cdot$ and $c^{\mathcal{G}} = e$

... eg $(\mathbb{Z}, +, 0)$

③ But also $\mathcal{M} = (\mathbb{R}, f^{\mathcal{M}}, c^{\mathcal{M}})$ is an $\mathcal{L}_1$-structure where:

$f^{\mathcal{M}}: \mathbb{R}^2 \to \mathbb{R}$ $\quad f^{\mathcal{M}}(x,y) = xy^3$

$c^{\mathcal{M}} = \pi^7$

... can interpret symbols however you want, just have to respect arities

<u>Def'n</u> Given $\mathcal{L}$-structures $\mathcal{M}, \mathcal{N}$ w/ universes M, N, an <u>$\mathcal{L}$-embedding</u> is an injective map $\eta: M \to N$ s.t.

1. for every $f \in \mathcal{F}$ and tuple $a_1, \dots, a_{n_f} \in M$:
$$\eta(f^{\mathcal{M}}(a_1, \dots, a_{n_f})) = f^{\mathcal{N}}(\eta(a_1), \dots, \eta(a_{n_f}))$$
2. For every $R \in \mathcal{R}$ and $a_1, \dots, a_{n_R} \in M$:
$$(a_1, \dots, a_{n_R}) \in R^{\mathcal{M}} \text{ iff } (\eta(a_1), \dots, \eta(a_{n_R})) \in R^{\mathcal{N}}$$
3. for every $c \in \mathcal{C}$, $\eta(c^{\mathcal{M}}) = c^{\mathcal{N}}$

- if η also surjective, it is an <u>isomorphism</u> of $\mathcal{M}$ w/ $\mathcal{N}$; we write $\mathcal{M} \cong \mathcal{N}$
- We say $\mathcal{M}$ is a <u>substructure</u> of $\mathcal{N}$ if $M \subseteq N$ and inclusion map is an embedding
  - ↳ just means the $\mathcal{M}$ functions = $\mathcal{N}$ functions restricted to M
  - ↳ likewise for relations / constants

<u>Ex's</u> ① $\mathcal{M} = (\mathbb{Z}, +, 0)$ is a substructure of $\mathcal{N} = (\mathbb{R}, +, 0)$

② Consider the structures $\mathcal{M} = (\mathbb{Z}, +, 0)$ and $\mathcal{N} = (\mathbb{R}, \cdot, 1)$
- define $\eta: \mathbb{Z} \to \mathbb{R}$ by $\eta(z) = 2^z$
- observe $\eta(z + z') = 2^{z+z'} = 2^z \cdot 2^{z'}$
  also $\eta(0) = 1$

- view language as $\mathcal{L} = \{f, c\}$
  So: $f^{\mathcal{M}} = +$ $c^{\mathcal{M}} = 0$
  $f^{\mathcal{N}} = \cdot$ $c^{\mathcal{N}} = 1$
- then above says $\eta(f^{\mathcal{M}}(z, z')) = f^{\mathcal{N}}(\eta(z), \eta(z'))$
  $\eta(c^{\mathcal{M}}) = c^{\mathcal{N}}$
- i.e. η is an embedding.

First-order formulas

- intuitively, formulas are assertions about structures
- formally tho: formulas are purely syntactic: strings of symbols built inductively in a certain way
- given language $\mathcal{L}$, symbols allowed in $\mathcal{L}$-formulas are:
  - from $\mathcal{L}$, or
  - variables $v_1, v_2, \ldots$ or (will also use $x, y, z, \ldots$ etc.)
  - connectives $\wedge, \vee, \neg$ ("and", "or", "not")
  - quantifiers $\exists, \forall$ ("exists", "forall")
  - parenthesis ( )

- before $\mathcal{L}$-fmlas, first need to define $\mathcal{L}$-terms
- int'n: terms denote elements/functions of an $\mathcal{L}$-structure (that may not be denoted by a symbol in $\mathcal{L}$)

Def'n Given $\mathcal{L}$, the $\mathcal{L}$-terms are built inductively as follows:

1. every constant symbol $c \in \mathcal{L}$ is a term
2. every variable v_i is a term
3. if $f \in \mathcal{F}$ and $t_1, \dots, t_{n_f}$ are terms, then so is $f(t_1, \dots, t_{n_f})$

E.g. if $\mathcal{L} = \{f, g, c\}$ w/ f binary, g unary

then $c, x, f(c,x), g(f(c,x))$ are $\mathcal{L}$-terms

- we evaluate terms in natural way:

Def'n Given $\mathcal{L}$, an $\mathcal{L}$-struct. $\mathcal{M}$, an $\mathcal{L}$-term t whose variables are among $\bar{v} = (v_{i_1}, \dots, v_{i_n})$ and a tuple $\bar{a} = (a_1, \dots, a_n) \in M^n$ we define $t^{\mathcal{M}}(\bar{a})$ inductively:

1. if t is c, $t^{\mathcal{M}}(\bar{a}) = c^{\mathcal{M}}$
2. if t is v_{i_k}, $t^{\mathcal{M}}(\bar{a}) = a_k$
3. if t is $f(t_1, \dots, t_{n_f})$,
$$t^{\mathcal{M}}(\bar{a}) = f^{\mathcal{M}}(t_1^{\mathcal{M}}(\bar{a}), \dots, t_{n_f}^{\mathcal{M}}(\bar{a}))$$

e.g. if $\mathcal{L} = \{f, g, c\}$ is lang from above
$\mathcal{M} = (\mathbb{R}, +, g, 1)$ where $g: \mathbb{R} \to \mathbb{R}$, $g(x) = x^2$
and $t(x)$ is the term $g(f(c,x))$
then $t^{\mathcal{M}}(4) = g(1+4) = (1+4)^2 = 25$

- now ready to define fmlas
- we define simple fmlas first

Def'n a formula φ is an **atomic $\mathcal{L}$-formula** if it has one of these forms:

① $t_1 = t_2$ where t_1, t_2 $\mathcal{L}$-terms
② $R(t_1, .., t_{n_R})$ where $R \in \mathcal{L}$ is rel'n symbol and $t_1, .., t_{n_R}$ $\mathcal{L}$-terms

e.g. if $\mathcal{L} = \{f, R, c\}$ w/ f unary, R binary

then $f(x) = c$ and $R(f(f(x)), y)$ are atomic $\mathcal{L}$-fmlas.

Def'n the class W of $\mathcal{L}$-fmlas is defined inductively:

① if φ is atomic then $\varphi \in W$
② if $\varphi \in W$, then $\neg \varphi \in W$
③ if $\varphi, \psi \in W$, then $(\varphi \wedge \psi) \in W$ and $(\varphi \vee \psi) \in W$
④ if $\varphi \in W$ and v_i a variable then $\exists v_i \varphi \in W$ and $\forall v_i \varphi \in W$.

e.g. if $\mathcal{L} = \{f, R, c\}$ from above then

$(f(x) = c \wedge R(x, y))$

and $\neg (f(x) = c)$

and

and $\forall x \, (f(x) = y)$
and $\exists y \, \forall x \, (f(x) = y)$
... are all $\mathcal{L}$-fmlas
(I will sometimes, as here, add extra parentheses for clarity)

<u>Note</u>: when dealing w/ common function or relation symbols will write e.g. $x+y$ and $x<y$ instead of $+(x,y)$, $<(x,y)$

<u>Bound vs. free vars</u>

- will be a lil informal about this
- say an occurrence of a var x in a fmla φ is <u>bound</u> if it occurs in scope of a quantifier $\forall x$ or $\exists x$
- o.w. it is <u>free</u>

e.g. in $\forall x \, (f(x) = y)$
x is bound y is free
in $\exists y \, \forall x \, (f(x) = y)$
both bound

- can have bound/free occurrences of same var, e.g. $(x = c \wedge \forall x \, (f(x) = y))$
  free, bound
- but will assume this doesn't happen