Math Logic, Week#IV

Moon's Day

Digression: Prenex normal form.

Sometimes it's easiest to deal with formulas with all their quantifiers in front.

Definition: A formula is in prenex norl form if it is the form

$$Q_1x_1Q_2x_2\cdots Q_kx_k\psi.$$

- where each Q_i is either $\forall$ or $\exists$, and
- φ contains no quantifiers.

Theorem: *For any formula φ of first order logic, there is a logically equivalent formula φ' in prenex normal form where also each variable occurs in only one scope.*

Proof: By induction on formulas — but I'll, instead, give an algorithm that fairly obviously translates into a proof.

Sometimes it is also useful, given such a formula $Q_1x_1Q_2x_2\cdots Q_kx_k\psi$, to translate ψ to conjunctive normal form (CNF).

Since ψ is quantifier-free, the atomic subformulas of ψ can be treated (for this purpose) as proposition letters.

Motivating Examples:

$p(x,y) \wedge (\neg p(z,x))$		is in the desired form.
$\exists z \forall x (p(x,y) \wedge \neg p(z,x))$		is in the desired form.
$(\neg(\forall x \exists y p(x,y))$	$\models\!\!=\!\!\mid$	$\exists x \forall y (\neg p(x,y))$
$\forall x p(x,y) \wedge \forall x p(z,x)$	$\models\!\!=\!\!\mid$	$\forall x (p(x,y) \wedge p(z,x))$

. . . but

$\exists x p(x,y) \wedge \exists x p(z,x)$ **is'nt equiv. to** $\exists x (p(x,y) \wedge p(z,x))$

However,

$\exists x p(x,y)$	$\models\!\!=\!\!\mid$	$\exists v p(x,v)$ **so**,
$\exists x p(x,y) \wedge \exists x p(z,x)$	$\models\!\!=\!\!\mid$	$\exists v (p(v,y) \wedge \exists w (p(z,w))$

so also

$\exists x p(x,y) \wedge \exists x p(z,x)$	$\models\!\!=\!\!\mid$	$\exists v \exists w (p(v,y) \wedge p(z,w))$

Also

$\forall x p(x,y) \wedge \forall x p(z,x)$	$\models\!\!=\!\!\mid$	$\forall v \forall w (p(v,y) \wedge p(z,w))$

More trouble:

$\forall v p(v,y) \rightarrow \forall w p(z,w)$ **is'nt equiv. to** $\forall v \forall w (p(v,y) \rightarrow p(z,w))$

However,

$\forall v p(v,y) \rightarrow \forall w p(z,w)$	$\models\!\!=\!\!\mid$	$\neg \forall v p(v,y) \vee \forall w p(z,w)$
	$\models\!\!=\!\!\mid$	$\exists v \neg p(v,y) \vee \forall w p(z,w)$
	$\models\!\!=\!\!\mid$	$\exists v \forall w ((\neg p(v,y)) \vee p(z,w))$

Vocabulary: $\exists x p(x,y) \wedge \exists x p(z,x)$ and $\exists v (p(v,y) \wedge \exists w (p(z,w))$ are *alphabetic variants* of each other.

Algorithm *prenex*(θ)

1. Find a formula φ_0 logicically equivalent to θ whose only propositional connectives are $\wedge$, $\vee$, and $\neg$.
2. Find an alphabetic variant φ of φ_0 with no variable used in two different scopes.
   (Work from the inner-most quantifiers out, as needed picking new variables not occurring at all elsewhere in the formula.)
3. **Recursive Algorithm** *prenexAux*(φ):

   If φ contains no quantifiers , let $\varphi' = \varphi$.

   Otherwise:

   If $\varphi = \forall x \psi$ (or $\exists x \psi$, respectively)**,**

   (a) set ψ' = *prenexAux*(ψ)

   (b) return $\varphi' = \forall x \psi'$ (or $\varphi' = \exists x \psi'$, resp.)

   Otherwise, if $\varphi = (\neg \psi)$

   (a) set ψ' = *prenexAux*(ψ)

   say $\psi' = Q_1 x_1 \cdots Q_k x_k \chi$,

   where each Q_i is $\forall$ or $\exists$ and χ is quantifier free

   (b) and return $\varphi' = Q'_1 x_1 \cdots Q'_k x_k (\neg \chi)$

   where each Q'_i is $\exists$ if Q_i is $\forall$, and $\forall$ if Q_i is $\exists$

   Otherwise if $\varphi = (\theta \wedge \chi)$ (or $(\theta \vee \chi)$, respectively)**,**

   (a) set $Q_1 x_1 \cdots Q_k x_k \alpha$ = *prenexAux*(θ)

   (b) set $Q' y_1 \cdots Q'_m y_m (\beta)$ = *prenexAux*(χ)

   (c) return $\varphi' = Q_1 x_1 \cdots Q_k x_k Q' y_1 \cdots Q'_m y_m (\alpha \wedge \beta))$

   (respectively, $\varphi' = Q_1 x_1 \cdots Q_k x_k Q' y_1 \cdots Q'_m y_m (\alpha \vee \beta)$

Notes: (1) We can obviously eliminate double negatives when we find them.

(2) in the final step, we can use quantifier prefix $Q_1 x_1 \cdots Q_k x_k Q' y_1 \cdots Q'_m y_m$, $\quad Q' y_1 \cdots Q'_m y_m Q_1 x_1 \cdots Q_k x_k$, or any interleaving of $Q_1 x_1 \cdots Q_k x_k$ and $Q' y_1 \cdots Q'_m y_m$.

For you to think through: Check that each of the quantifier manipulations in the algorithm produces a logically equivalent result. This requires you, in each case, to go back to the definition of satisfaction in terms of structures $\mathfrak{A}$ and variable assignments s.

Aside for my fellow pedants: Origin of the strange-sounding word — from `www.oxforddictionaries.com`:

> *1930s; earliest use found in Journal of Symbolic Logic. From post-classical Latin praenexus tied or bound up in front, past participle of praenectere (though only recorded in past participle) from classical Latin prae- + nectere to bind, connect.*

Note on coverage: We're skipping §2.3.

Read of most of §2.4, and prepare initial quesitons.

- §2.4 covers one approach to formalizing the notion of a mathematical proof.
- There are many other formalisms — all of which are equivalent.
- Enderton chooses his, I presume, to get to his theorems a fast as possible.

  If you were going to program a proof procedure, you'd probably use a different one.

  And I, at least, find some other ones more intuitively obvious.

Open Homework Problems:

- §1.7 ## 3, 8
- §2.2 ## 6, 9, 11, 13, 14, 15

Wodin's Day

Comment from after class Moon's Day: In class I introduced the next topic: formalizing the notion of correct arguments — especially mathematical proofs.

In §§2.1-2.2, Enderton defined structures, satisfaction, and logical inference — thus a *semantics* (meaning): a relation between formulas and their truth.

Now we hope to model proofs.

There are 3 standard desiderata here:

Soundness: If there is a proof of φ from a set Γ of axioms, then $\Gamma \models \varphi$.

Completeness: If $\Gamma \models \varphi$, then there is a proof of φ from Γ.

Effectiveness: Given any finite Γ, φ, and a candidate proof P, we can effectively decide whether P is a correct proof of φ from Γ.

The same is true Γ is itself decidable.

But, don't read too much into Effectiveness.

- We can recognize whether any given P is a proof of φ from Γ.
- But it turns out that there is, in general, no effective procedure to test whether such a P exists. And there is no effective procedure to test whether a formula φ is valid.

  Contrast propositional logic: there truth tables let us check validity of formulas (although in exponential time).

  It turns out that the set of logically valid formulas o f first order logic is *semi-decidable*:.

  - In Enderton's proof system, all proofs are finite sequences of characters in some finite alphabet.
  - Most finite sequences of those characters are nonsense,
and some are invalid proofs,
but we can recognize those that are valid proofs.
  - So the semi-decision procedure — to see whether φ is valid:
1 by 1, generate all such finite sequences of characters.
As you generate each one, check to see whether it's a valid proof of ϕ. If so, output "*proved*."
  - On input of a non-valid formula φ, the above procedure will grind on forever.
(We'll prove that later in the semester — assuming what is called the *Church-Turing-Hypothesis*: that the formalism we provide really does capture the intuitive notion of what is computable.)

“Theorems” and “Metatheorems”

- We’re going to build a mathematical model of proofs.
- But we’ll also prove theorems *about* the formal proofs system.

  These are often called *meta-theorems*.

  Enderton calls the formal proofs *deductions*, to emphasize the difference.

  (This is analogous to the *formal language* / *meta-language* distinction discussed earlier.)

  > *(I don’t know what Kurt Gödel’s inspiration for his work was, but it looks to me as if sometimes he started out with a paradox caused by confusing formal language with meta-language and found a way to* almost *capture it all in the formal language, producing something profound.)*

Components of Enderton's "Hilbert-style" deduction system:

A proof P of a formula φ from a set Γ of formulas is a finite sequence

$$\begin{array}{l}\psi_0\\ \psi_1\\ \vdots\\ \psi_k \;= \varphi\end{array}$$

Where each ψ_i

- is an element of Γ,
- is an element of a chosen, decidable, set Λ of logically valid formulas, or
- follows from previous ψ_js by some sound *inference rule*.

The set Λ of varies from one system to another.

Enderton has a large set of them. However, we can relatively easily show they are all logical truths.

Enderton uses only one inference rule in his system, *modus ponens*:

$$\begin{array}{c}\alpha\\ \vdots\\ (\alpha \to \beta)\\ \vdots\\ \beta\end{array} \qquad \text{or} \qquad \begin{array}{c}(\alpha \to \beta)\\ \vdots\\ \alpha\\ \vdots\\ \beta\end{array}$$

— if α and $(\alpha \to \beta)$ are both true, then β must be true also.

¿Is that inference rule sound?

Note: Enderton's concern seems to be that *there is* such a proof system, and he gets there reasonably fast.

Enderton's Axiom set Λ:

Defn: $\forall x_1 \forall x_2 \ldots \forall x_k \varphi$ (all $\forall$'s) is a *generalization* of φ.

Defn: Λ includes all generalizations of 5 classes of formulas:

1. **Tautologies** (of first order logic)**:**

- Start with a tautology of propositional logic — *e.g.*,

$$(A \rightarrow (\neg B)) \rightarrow (B \rightarrow (\neg A))$$

- and a formula of first order logic to substitute for each proposition letter, say

$$\forall y(\neg \forall x(\neg(p(x,y)) \rightarrow (p(z,y))))$$

and

$$(\neg \forall y(\neg(\neg(p(y,y))) \rightarrow (p(y,z))))$$

- Substitute the 1st formula for 1 of the proposition letters:

$$(\forall y(\neg \forall x(\neg(p(x,y)) \rightarrow (p(z,y)))) \rightarrow (\neg B)) \\ \rightarrow \\ (B \rightarrow (\neg \forall y(\neg \forall x(\neg(p(x,y)) \rightarrow (p(z,y))))))$$

- Substitute the 2nd formula for the other proposition letter:

$$(\forall y(\neg \forall x(\neg(p(x,y)) \rightarrow (p(z,y)))) \\ \rightarrow (\neg(\neg \forall y(\neg(\neg(p(y,y))) \rightarrow (p(y,z)))))) \\ \rightarrow \\ ((\neg \forall y(\neg(\neg(p(y,y))) \rightarrow (p(y,z)))) \\ \rightarrow (\neg \forall y(\neg \forall x(\neg(p(x,y)) \rightarrow (p(z,y))))))$$

- The result — and all it's generalizations — are in Λ.

- **¿How can we tell whether a formula is a tautology?**

- **Homework: §2.4 # 3.**

2. **Substitution of terms for universally quantified variables:**

- For a formula α, a variable symbol x, and a term t:

  to form α^x_t, replace each <u>*free*</u> occurrence of x in α with t.

- So for $\alpha = (p(x,y) \to \exists z(\neg(p(x,z))))$, and $t = f(y,3)$,

$$\alpha^x_t = (p(f(y,3),y) \to \exists z(\neg(p(f(y,3),z)))).$$

- Enderton gives a formal definition of this process of substitution — by recursion on formulas. **¡Read it!**

- Then $\forall x \alpha \to \alpha^x_t$ is in Λ — as are all of its generalizations.

- In the example,

$$\begin{array}{ll} \forall x(p(x,y) \to \exists z(\neg(p(x,z)))) \to & \\ \quad (p(f(y,3),y) \to \exists z(\neg(p(f(y,3),z)))) & \in \Lambda \\ \forall z(\forall x(p(x,y) \to \exists z(\neg(p(x,z)))) \to & \\ \quad (p(f(y,3),y) \to \exists z(\neg(p(f(y,3),z))))) & \in \Lambda \\ \forall y \forall z(\forall x(p(x,y) \to \exists z(\neg(p(x,z)))) \to & \\ \quad (p(f(y,3),y) \to \exists z(\neg(p(f(y,3),z))))) & \in \Lambda \\ \forall u \forall v \forall w(\forall x(p(x,y) \to \exists z(\neg(p(x,z)))) \to & \\ \quad (p(f(y,3),y) \to \exists z(\neg(p(f(y,3),z))))) & \in \Lambda \end{array}$$

  etc.

- **but we do <u>not</u> include substituting**

$$f(z,3) \text{ for } x \text{ in } (p(x,y) \to \exists z(\neg(p(x,z))))$$

  because making the substitution would trap the z of $f(z,3)$ inside the $\exists z$ quantifier in α — we'd change the scope of this occurrence of z.

- **Vocuabulary:** $f(z,3)$ is not *substitutable* for x in α.

- **¿How can we identify all such axioms?**

Frea's Day

3. **Axiom Group 3:** $\forall x(\alpha \to \beta) \to (\forall x\alpha \to \forall x\beta)$ (for any α, β).

4. **Axiom Group 4:** $\alpha \to \forall x\alpha$

   where α is a formula in which x does not occur free.

   ¿Why in the world would Enderton have included this axiom?

5. **Axiom Group 5:** *If the language includes* =:

   $$v_i = v_i \quad \text{for each variable symbol } v_i.$$

6. **Axiom Group 6:** *If the language includes* =:

   $$x = y \to (\alpha \to \alpha'), \qquad \text{where} \qquad \alpha \text{ is atomic, and}$$

   - and α' is obtained by replacing x in *0 or more — but not necessarily all* — occurrences of x by y.

Caution regarding Wodin's Day's Class:

Definitions need to be both memorized and understood.

(*E.g.*, "*free*". Good time: before you start homework problem.)

Notation: $\Gamma \vdash \varphi$ means there is a nproof of φ from Γ.

We rephrase 2 of our desiderata for a proof system:

Soundness: If $\Gamma \vdash \varphi$, then $\Gamma \models \varphi$.

Completeness: If $\Gamma \models \varphi$, then $\Gamma \vdash \varphi$.

After we prove soundness and completeness of the proof system, we sometimes become a bit casual about whether whether we write $\models$ or $\vdash$, but we must be very careful until we finish those proofs.

Sample deductions from the text

- shown as a sequence, rather than a tree, and
- formatted as I request you do:

1. A deduction of $P(x) \to \exists y P(y)$ from $(\Gamma =)\ \emptyset$:

   First deabbreviate the formula: $P(x) \to \neg\forall y \neg P(y)$

line#	formula	justification
1.	$\forall\neg P(y) \to \neg P(x)$	axiom, group 2
2.	$(\forall y\neg P(y) \to \neg P(x))$ $\to (P(x) \to \neg\forall y\neg P(y))$	axiom, group 1
3.	$P(x) \to \neg\forall y\neg P(y)$	*modus ponens* from 1, 2

2. A deduction of $\forall x(P(x) \to \exists y P(y))$:

   Note: This is just $\forall x($formula proved above$)$.

	formula	justification
1.	$\forall x(\ (\forall y\neg P(y) \to \neg P(x))$ $\to (P(x) \to \neg\forall y\neg P(y)))$	axiom, group 1
2.	$\forall x(\ (\forall y\neg P(y) \to \neg P(x))$ $\to (P(x) \to \neg\forall y\neg P(y)))$ $\to$ $(\forall x(\ (\forall y\neg P(y) \to \neg P(x))\ )$ $\to \forall x(P(x) \to \neg\forall y\neg P(y))))$	axiom, group 3
3.	$\forall x(\forall y\neg P(y) \to \neg P(x))$ $\to \forall x(P(x) \to \neg\forall y\neg P(y))$	*modus ponens* from 1, 2.
4.	$\forall x(\forall y\neg P(y) \to \neg P(x))$	axiom, group 2
5.	$\forall x(P(x) \to \neg\forall y\neg P(y))$	*modus ponens* from 3, 4.

Note how deduction #1 is modified to get deduction #2

Meththeorems:

Generalization (Meta)Theorem:

If $\Gamma \vdash \phi$ *and* x *does not occur free in* Γ, $\Gamma \vdash \forall x\phi$.

Note Comparison:

- We already saw that, for Γ a set of sentences, if $\Gamma \models \phi$, then $\Gamma \models \forall x\phi$
- The same holds for Γ a set of formulas, so long as x does not occur free in Γ.
- ¡So, if we hope to prove Soundness and Completeness, the same had better be true for "$\vdash$"!

Important Point: We must be careful: infer no more than the metatheorem says (unless you first prove the stronger theorem).

- If x does not occur free in Γ and $\Gamma \vdash \varphi$, $\Gamma \vdash \forall x\varphi$.
- But it is **not true** (in general) that, if x does not occur free in Γ,

$$\Gamma \vdash (\varphi \rightarrow \forall x\varphi).$$

- (Nor is the analogous statement for "$\models$" true.
¿Can you show that?)

Proof of the generalization theorem:

- Enderton proves it by induction on the length of the derivation of ϕ.
- Basically, he shows how to transform a proof of ϕ, step by step, to a proof of $\forall x\phi$ —
just as he did in the 2 examples I put in above.
- **Assignment:** Read Enderton's proof carefully.
And either understand it or bring in questions.

Application of the Generalization Theorem: Show that

$$\forall x \forall y \alpha \vdash \forall y \forall x \alpha :$$

1. Note that, in any formula β, any variable z is substitutable for itself. **¿Why?**

   And β^z_z is always just β.

2. Construct a proof of α from $\Gamma = \{\forall x \forall y \alpha\}$:

1.	$\forall x \forall y \alpha$	in Γ
2.	$\forall x \forall y \alpha \to \forall y \alpha$	axiom, group 2
3.	$\forall y \alpha$	*modus ponens*, from 1, 2
4.	$\forall y \alpha \to \alpha$	axiom group 2
5.	α	*modus ponens*, from 3,4

3. Since $\Gamma \vdash \alpha$, and x doesn't occur free in Γ,

   by the Generalization Theorem, $\Gamma \vdash \forall x \alpha$.

4. Since $\Gamma \vdash \forall x \alpha$, and y doesn't occur free in Γ,

   by the Generalization Theorem, $\Gamma \vdash \forall y \forall x \alpha$.

Rule T: *If* $\Gamma \vdash \alpha_1$, $\Gamma \vdash \alpha_2$, $\ldots$, $\Gamma \vdash \alpha_n$,

and $\{\alpha_1, \ldots, \alpha_n\}$ *tautologically implies* β,

then $\Gamma \vdash \beta$.

- Review: What does "*tautologically implies*" mean?
- **Assignment: Read both proofs in the book.**

  If you don't understand them, bring questions about them to class.

  Read & understand the 2 (Meta-)Corollaries, *Contrapostion* and *Reductio ad Absurdum*.