Homework #8

For Friday, March 18

Problems 2–6 have to do with a more explicit proof of the restricted version of the completeness theorem: if $\models \varphi$, then $\vdash \varphi$. Note that, other than Problem 3, none are starred.

1. Read through Section 2.5 in van Dalen.

◦ 2. Using the proof of Theorem 1.3.8 in van Dalen, describe an algorithm that converts any formula φ to formulas $\varphi^\vee$ and $\varphi^\wedge$, in disjunctive and conjunctive normal form, respectively.

⋆ 3.

a. Remember that a formula φ is *satisfiable* if there is a truth assignment v such that $[\![\varphi]\!]_v = 1$. φ is *unsatisfiable* if it is not satisfiable. Show that for any formula φ, φ is unsatisfiable if and only if $\models \neg\varphi$.

b. Now suppose φ is of the form

$$\varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_k$$

in disjunctive normal form, so that each formula φ_i is a conjunction of atomic formulas and their negations. Show that φ is satisfiable if and only if one of the conjunctions φ_i does not contain an atomic formula p_j together with its negation $\neg p_j$.

c. Use this, together with the previous problem, to give an algorithm to determine whether or not a formula φ is valid.

◦ 4. This problem outlines a more constructive approach to the following special case of the completeness theorem: if $\models \varphi$, then $\vdash \varphi$.

a. Modify the algorithm of problem 5 so that given φ, it outputs not only a formula $\varphi^\wedge$, but also a proof of $\varphi \leftrightarrow \varphi^\wedge$.

b. Show that if $\varphi^\wedge$ is valid, it is easy to prove. (The previous problem is relevant.)

◦ 5. If φ is any formula, show that

$$length(\varphi^\vee) \leq 2^{length(\varphi)+3}.$$

(Use the definition of $\varphi^\vee$ implicit in Theorem 1.3.9 on page 26.)

$\circ$ 6.

a. Assuming φ has length n, what is the worst-case running time of the algorithm in part (c) of the problem 6?

b. Another way to determine if a formula φ is a tautology is to compute its value on every truth assignment (the "truth table" method). What is the worst-case running time of this algorithm?

c. Come up with a polynomial-time algorithm for determining if a propositional formula φ is a tautology or not, or prove that no such algorithm exists. (Note: a successful solution to this problem amounts to settling the famous open question, $P = NP$?.)

$\star$ 7. Treating the connectives $\wedge$, $\vee$, $\to$, and $\bot$ as basic (i.e. $\vee$ is not defined from other connectives, but $\neg$, $\leftrightarrow$, and $\top$ are), we get intuitionistic (propositional) logic by deleting RAA from our set of inference rules. The remaining rules, then, are $\wedge$ intro and elim; $\vee$ intro and elim; $\to$ intro and elim; and $\bot$ elim (that from $\bot$ you can conclude anything). We show that RAA does not follow from these rules by constructing a sound semantics for intuitionistic logic in which RAA does not hold. Let I be the closed unit interval,

$$I = [0,1] = \{x \in \mathbb{R} | 0 \leq x \leq 1\}$$

a. Given a valuation

$$v : \{p_0, p_1, \ldots\} \to I$$

justify briefly that there is a unique function

$$[\![-]\!]_v : \mathrm{PROP} \to I$$

such that

$$[\![\bot]\!]_v = 0 \tag{1}$$

$$[\![\phi \wedge \psi]\!]_v = min\{[\![\phi]\!]_v, [\![\psi]\!]_v\} \tag{2}$$

$$[\![\phi \vee \psi]\!]_v = max\{[\![\phi]\!]_v, [\![\psi]\!]_v\} \tag{3}$$

$$[\![\phi \to \psi]\!]_v = \begin{cases} 1, & \text{if } [\![\phi]\!]_v \leq [\![\psi]\!]_v \\ [\![\psi]\!]_v, & \text{otherwise} \end{cases} \tag{4}$$

b. Define *semantic consequence with respect to valuations in I* by

$$\{\phi_1, \ldots, \phi_n\} \vDash^I \psi \Leftrightarrow \text{if } min\{\phi_1, \ldots, \phi_n\} \leq [\![\psi]\!]_v$$

for all valuations $v : \{p_0, p_1, \ldots\} \to I$, where Γ is a finite set of formulas. (Accordingly, ϕ is valid, $\vDash^I \phi$, iff $[\![\phi]\!]_v = 1$ for all valuations

v.) Show by induction on derivations that this defines a sound semantics for intuitionistic logic. The induction step has one case for each derivation rule, do at least two of these cases. The $\vee$-elim case is tricky, so you might want to avoid that one.

- ◦ c. Do all cases in the induction proof above.
- d. Use the definition of $[\![-]\!]_v$ to compute $[\![\phi \vee \neg\phi]\!]_v$. Conclude that RAA does not follow from the other deduction rules.
- e. Is this semantics complete? That is, is it the case that

$$\Gamma \models^I \phi \Rightarrow \Gamma \vdash \phi$$

for Γ a finite set of formulas? Justify your answer.

8. Do problem 1 on page 60.

⋆ 9. Do problem 4 on page 67. For each one, just indicate whether the term is "free" or "not free" for the variable in question, and carry out the substitution either way.

10. Consider a first-order language, with relation symbols $<$ and $=$. The intended interpretation is the natural numbers, with "less-than" and "equality." Formalize the following statements:
    - a. "x is less than or equal to y"
    - b. "0 is the smallest number"
    - c. "there is a smallest number"
    - d. "there is no largest number"
    - e. "every number has an immediate successor" (in other words, for every number, there is another one that is the "next largest")

⋆ 11. The language of arithmetic is intended to describe the natural numbers, with constant symbols 0 and 1, functions symbols $+$ and $\times$, and a relation symbol $<$. Formalize the following:
    - a. "x is prime."
    - b. "There are infinitely many primes."
    - c. the principle of induction for a formula $\varphi(x)$
    - d. the least-element principle for a formula $\varphi(x)$

    You can define auxiliary notions along the way. Remember that a number x is prime if it is greater than 1 and its only divisors are 1 and itself. To say "there are infinitely many primes," say that there are primes that are arbitrarily large. You can find the induction principle and the least element principle on pages 7 and 8 of the notes.

12. The language of sets has a single binary relation symbol $\in$, where $x \in y$ is meant to denote the fact that x is an element of y. In the intended interpretation, everything is a set; that is, every object is a set, whose elements are sets, and so on. In this language, formalize the following statements:

    a. x is a subset of (or equal to) y.

    b. Two sets are equal if and only if they have the same elements.

    c. For any set z, there is another set w consisting of all the subsets of z. (You can use the symbol $\subseteq$ to abbreviate the formula you found in part (a).)

13. In a language with a binary relation symbol $<$, formalize the following statements:

    a. $<$ is transitive.

    b. Between any two things there is another thing.

    c. There is a smallest thing.

    d. There is no largest thing.

$\circ$ 14. In the language of groups, which has a multiplication symbol $\cdot$ and a constant symbol i, formalize the following statements:

    a. $\cdot$ is associative.

    b. $\cdot$ is not commutative.

    c. i is an identity.

    d. Every element has an inverse.

$\star$ 15. Let $\mathcal{A}$ be the structure consisting of "all objects on the planet Earth" with relations *IsCow(x)*, *EatsGrass(x)*, *IsCar(x)*, etc. Give reasonable formalizations of the following sentences:

    a. All cows eat grass.

    b. There is a car that is blue and old.

    c. No car is not pink.

    d. All cars that are old must be inspected annually.

16. In plain English, express the negation of each of these sentences, that is, the assertion that each sentence is false.

$\circ$ 17. In a language with unary relation symbols $Red(x)$, $Car(x)$, and $Broken(x)$, come up with a reasonable formalization of the sentence "The red car is broken."

- $\star$ 18. Do problem 1 on page 72. Use the symbols P, T, S, and c to denote addition, multiplication, successor ("+1"), and 0 respectively.
- 19. Do problems 2 and 3 on page 72.
- $\circ$ 20. Prove unique readability for terms and formulas in a given first-order language (and/or write a parser).