ADVENTURES IN LOGIC FOR UNDERGRADUATES

by
Harvey M. Friedman
Distinguished University Professor
Mathematics, Philosophy, Computer Science
Ohio State University

Lecture 1. Logical Connectives

LECTURE 1. LOGICAL CONNECTIVES. Jan. 18, 2011

LECTURE 2. LOGICAL QUANTIFIERS. Jan. 25, 2011

LECTURE 3. TURING MACHINES. Feb. 1, 2011

LECTURE 4. GÖDEL'S BLESSING AND GÖDEL'S CURSE.
Feb. 8, 2011

LECTURE 5. FOUNDATIONS OF MATHEMATICS
Feb. 15, 2011

SAME TIME - 10:30AM

SAME ROOM - Room 355 Jennings Hall

WARNING: CHALLENGES RANGE FROM EASY, TO MAJOR PARTS OF COURSES

AND, OR, NOT

We start with one common way of connecting sentences.

Suppose I tell you

Mike is a wibel AND
Jane is a zibel

I don't know about you, but I don't know what this means!
But there is something about this sentence that we do know.

This sentence, taken as a whole, is true or false according to whether its two constituents are true or false.

AND is an example of a logical connective.

OR, NOT are also examples of logical connectives:

AND, OR, NOT

Mike is a wibel OR
Jane is a zibel

Again, this sentence, taken as a whole, is true or false according to whether its two constituents are true or false.

Mike is not a wibel

This sentence, taken as a whole, is true or false according to whether its one constituent is true or false.

What is the rule that determines the truth value of these three example sentences in terms of the truth values of their constituents?

TRUTH VALUES: T if true; F if false.

RULES FOR AND, OR, NOT
THE TRUTH TABLES

Using letters for the constituents, we write

$A \wedge B$ (A and B)

$A \vee B$ (A or B)

$\neg A$ (not A)

There are four possibilities for A,B.

A	B	$A \wedge B$	$A \vee B$	$\neg A$
T	T	T	T	F
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

LOGICAL EQUIVALENCE: REDUCING CONNECTIVES

Two formulas are *logically equivalent* (written $\equiv$) if and only if they have the same truth values under the same assignments of truth values to their aggregate letters. E.g.,

$$(A \wedge B) \equiv (B \wedge A) \vee (C \wedge \neg C)$$

It is also easy to check that

$$A \vee B \equiv \neg(\neg A \wedge \neg B)$$

$$A \wedge B \equiv \neg(\neg A \vee \neg B)$$

CHALLENGE. Every propositional formula in $\neg, \vee$ is logically equivalent to one in $\neg, \wedge$. Every propositional formula in $\neg, \wedge$ is logically equivalent to one in $\neg, \vee$. (Use induction).

COMPLETENESS OF CONNECTIVE SETS

We have been using the connective set $\neg,\wedge,\vee$. This set is *logically complete* in the following sense.

Let S be a set of assignments of truth values to the letters $A_1,\ldots,A_n$, $n \geq 1$. There is a propositional formula in $\neg,\wedge,\vee,A_1,\ldots,A_n$ which has truth value T under exactly the assignments in S.

Because of the reductions from the last page, we conclude that the connective sets $\{\neg,\vee\}$ and $\{\neg,\wedge\}$ are also complete.

The connective set $\{\vee,\wedge\}$ is not complete because every propositional formula in $\{\vee,\wedge,A\}$ is T under some assignment.

CHALLENGE. Verify all of the statements on this page.

MATH NEEDS AND USES MORE CONNECTIVES

Although $\{\neg,\wedge,\vee\}$, or even $\{\neg,\wedge\}$, $\{\neg,\vee\}$, is logically complete, math needs more connectives to be humanly manageable. The most crucial addition is "implies" or "if then". Also, less crucially, "iff" is in common use.

A	B	$A \rightarrow B$	$A \leftrightarrow B$
T	T	T	T
T	F	F	F
F	T	T	F
F	F	T	T

So math uses the five connectives $\neg,\wedge,\vee,\rightarrow,\leftrightarrow$.

From the above truth table we see that
(A = T, B = T) and (A = F, B = F) are both satisfiers of both

$A \rightarrow B$ and $A \leftrightarrow B$.

SATISFIER OF A COMPLICATED FORMULA

Consider the example formula

$$((A \rightarrow B) \leftrightarrow (A \vee \neg B)) \rightarrow ((C \wedge \neg D) \vee (B \leftrightarrow \neg E))$$

($A = T$, $B = T$, $C = F$, $D = T$, $E = F$) is a satisfier of this:

A	B	C	D	E	$A \rightarrow B$	$\neg B$	$A \vee \neg B$	$(A \rightarrow B) \leftrightarrow (A \vee \neg B)$
T	T	F	T	F	T	F	T	T

$\neg D$	$C \wedge \neg D$	$\neg E$	$B \leftrightarrow \neg E$	$(C \wedge \neg D) \vee (B \leftrightarrow \neg E)$
F	F	T	T	T

$((A \rightarrow B) \leftrightarrow (A \vee \neg B)) \Rightarrow ((C \wedge \neg D) \vee (B \leftrightarrow \neg E))$
T

This is just one the $2^5 = 32$ rows of its truth table.

MORE REDUCTIONS AND CONNECTIVE COMPLETENESS

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

$$(A \leftrightarrow B) \equiv (A \wedge B) \vee (\neg A \wedge \neg B)$$

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

From previous pages, $\{\neg,\wedge\}$, $\{\neg,\vee\}$ are logically complete. From the last of the above, $\{\neg,\rightarrow\}$ is logically complete.

CHALLENGE. Determine which of the 32 subsets of $\{\neg,\wedge,\vee,\rightarrow,\leftrightarrow\}$ are logically complete.

TAUTOLOGIES, CONTRADICTIONS

$$A \vee \neg A \qquad A \rightarrow A \qquad A \leftrightarrow A$$

All three of these come out T for all truth assignments to A. A more sophisticated example is

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

which comes out T for all truth assignments to A,B.

$$A \wedge \neg A \qquad A \leftrightarrow \neg A$$

comes out F for all truth assignments to A.

A *tautology* is a propositional formula whose truth value is T under all assignments of truth values to its letters. A *contradiction* is a propositional formula whose truth value is F under all assignments of truth values to its letters.

TAUTOLOGIES
SATISFIABILITY

A propositional formula is *satisfiable* if and only if it has a satisfier. I.e., its truth value is T under some assignment of truth values to its letters.

Life may become very difficult when there are a lot of letters:

$$(((\neg A \leftrightarrow B) \vee (A \rightarrow \neg C)) \wedge ((B \vee D) \vee E)) \rightarrow (E \wedge \neg\neg B)$$

Here there are $2^5 = 32$ rows in the truth table.

CHALLENGE. Any way of determining whether propositional formulas are tautologies can be easily converted to a way of determining whether propositional formulas are satisfiable, and vice versa.

SATISFIABILITY
$1,000,000.00

The obvious foolproof method for determining whether a propositional formula is satisfiable is by “merely” trying all assignments of truth values to the letters.

This could take an exponential amount of time relative to the size of the input formula.

Can this be done using at most a polynomial amount of time relative to the input size? The method is required to work flawlessly for all propositional formulas.

A proof of this one way or the other wins a $1,000,000.00 prize from the Clay Foundation.

LOGICAL IMPLICATION
LOGICAL EQUIVALENCE

Formula α *logically implies* formula β if and only if any truth assignment to the aggregate of letters making α true, makes β true. We write this as $\alpha \Rightarrow \beta$.

Don't confuse $\Rightarrow$ with the connective $\rightarrow$.

THEOREM. $\alpha \Rightarrow \beta$ if and only if $\alpha \rightarrow \beta$ is a tautology.

$\alpha \equiv \beta$ if and only if $\alpha \Rightarrow \beta$ and $\beta \Rightarrow \alpha$.

$\alpha \equiv \beta$ if and only if $\alpha \leftrightarrow \beta$ is a tautology.

THEOREM. $\Rightarrow$ is reflexive and transitive, but not symmetric. $\equiv$

is an equivalence relation (i.e., reflexive, transitive and symmetric).

CHALLENGE: Prove these Theorems.

COMMUTATIVITY, ASSOCIATIVITY, DISTRIBUTIVITY

There are some very basic logical equivalences that are like those that you see in algebra (with some differences).

$A \wedge B \equiv B \wedge A$ $\quad A \vee B \equiv B \vee A$ $\quad A \leftrightarrow B \equiv B \leftrightarrow A$

$(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$

$(A \vee B) \vee C \equiv A \vee (B \vee C)$

$(A \leftrightarrow B) \leftrightarrow C \equiv A \leftrightarrow (B \leftrightarrow C)$

$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$ (like $a(b+c) = ab+ac$)

$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$ (like $a+bc = (a+b)(a+c)$) ??

These hold no matter what propositional formulas we substitute for the letters A,B,C, so long as we substitute the same formulas for the same letters.

DISJUNCTIVE NORMAL FORM

It is often very convenient to put an arbitrary propositional formula in a standard form, that you can more easily work with.

By associativity and commutativity, we can speak freely of "disjunctions of formulas" and of "conjunctions of formulas".

A *literal* is a letter or a negated letter (i.e., $\neg A_i$).

A formula is in *disjunctive normal form* if and only if it is a disjunction of formulas, each of which is a conjunction of literals.

THEOREM. Every propositional formula in $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, is logically equivalent to a formula in disjunctive normal form.

CHALLENGE: Prove this. Use satisfiers.

CONJUNCTIVE NORMAL FORM

A formula is in *conjunctive normal form* if and only if it is a conjunction of formulas, each of which is a disjunction of literals.

THEOREM. **Every propositional formula in $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ is logically equivalent to a formula in conjunctive normal form.**

CHALLENGE: Prove this. Use satisfiers.

SATISFIABILITY OF INFINITE SETS OF PROPOSITIONAL FORMULAS

So far, we have been talking about satisfiability of propositional formulas. We now consider satisfiability of sets of propositional formulas. This means that there is a single assignment of truth values (i.e.,a satisfier) that makes all propositional formulas in the set true.

We will assume the fixed alphabet of distinct letters $A_1, A_2, A_3, \ldots$ for use in propositional formulas.

The *satisfiers* consist of the assignments of truth values to all of the letters $A_1, A_2, A_3, \ldots$. Thus satisfiers may assign truth values to letters that do not occur in the set of propositional formulas. This is very convenient, and clearly does not affect satisfiability.

COMPACTNESS THEOREM - PROOF

COMPACTNESS THEOREM. A set S of propositional formulas is satisfiable if and only if every finite subset of S is satisfiable.

For this interesting proof, assume that every finite subset of S is satisfiable. We want to show that S is satisfiable.

Every finite subset of S has a satisfier which assigns T to A_1, or every finite subset of S has a satisfier which assigns F to A_1.

CHALLENGE: Prove this claim. Assume the claim is false, and derive a contradiction. (You MUST use proof by contradiction here!).

This gives us a "partial satisfier" f_1 that assigns only to A_1. Every finite subset of S has a satisfier extending f_1.

COMPACTNESS THEOREM - PROOF

Every finite subset of S has a satisfier extending f_1.

We claim that every finite subset of S has a satisfier extending f_1 that assigns T to A_2, or every finite subset of S has a satisfier extending f_1 that assigns F to A_2.

CHALLENGE: Prove this claim. As before, assume the claim is false, and derive a contradiction. (You MUST use proof by contradiction here!).

This gives us a partial satisfier f_2 extending f_1, which assigns only to A_1,A_2. Every finite subset of S has a satisfier extending f_2.

Continue in this way, obtaining an infinite series of growing partial satisfiers. The union of all of these partial satisfiers gives us a satisfier for all of S. WHY?

LOGICAL IMPLICATIONS

We now consider logical implications of infinite sets of propositional formulas.

Let K be a set of propositional formulas, and A be a propositional formula (all in $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$). We say that K *logically implies* A if and only if every satisfier of K is a satisfier of A. We write $K \Rightarrow A$.

THEOREM. Let K,A be as above. If $K \Rightarrow A$, then for some finite subset K_0 of K, $K_0 \Rightarrow A$.

Suppose $K \Rightarrow A$. Then $K \cup \{\neg A\}$ is not satisfiable. By the Compactness Theorem, $K_0 \cup \{\neg A\}$ is not satisfiable, for some finite $K_0 \subseteq K$. Therefore K_0 logically implies A. WHY?

AXIOMATIZATION, HILBERT STYLE

It is common in logic to provide axioms and rules of inference that yield, exactly, some important class of formulas. There are several types of such “formal systems”.

Here we want to give axioms and rules of inference for proving tautologies in some chosen set of connectives.

In a Hilbert style axiomatization of propositional logic, we choose a finite set of formulas, in some chosen set of connectives, as the axioms. We can use the following two rules of inference.

1. Any substitution instance of any axiom is considered derived (using only the chosen set of connectives).
2. If formulas α, and $\alpha \to \beta$ have been derived, then β is considered derived. (Modus Ponens).

SOUNDNESS AND COMPLETENESS

Such a formal system is said to be *sound* if and only if every formula that is derived is a tautology.

Such a formal system is said to be *complete* if and only if every tautology, using the chosen set of connectives, is derivable.

There are sound and complete Hilbert systems for $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$.

They are somewhat cluttered. It is typical to focus on the case $\neg, \rightarrow$. We can use these formulas as axioms (Elliot Mendelson):

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

with SUBSTITUTION for axioms ($\neg, \rightarrow$ only), and MODUS PONENS.

SOUNDNESS AND COMPLETENESS

Mendelson, Introduction to Mathematical Logic, 1979, has a proof of the soundness and completeness of the above axiom system. Also see List of logic systems, Wikipedia, where it cites Jan Lukasiewicz for using a simpler third axiom:

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

with SUBSTITUTION for axioms ($\neg$,$\rightarrow$ only), and MODUS PONENS.

Soundness asserts that every theorem of the system is a tautology.

CHALLENGE: Prove soundness for these two systems by induction.

COMPLETENESS

Completeness for such systems is much harder, and asserts that every tautology is a theorem of the system. First carefully define *proofs* as finite sequences of formulas, each of which is either a substitution instance of an axiom, or follows from previous entries, in the finite sequence, by Modus Ponens.

The next step is to extend this to *proofs from a set of assumptions*. Then state and prove the Deduction Property:

If A can be proved from a set $S \cup \{B\}$,
then $B \rightarrow A$ can be proved from S.

CHALLENGE. Prove the Deduction Property for these two systems. Use induction on the length of proofs.

CHALLENGE. Finish the proof of Completeness for these two systems. Use satisfiers.

HILBERT SYSTEM FOR $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

We incorporate $\wedge, \vee, \leftrightarrow$, into the Mendelson system.

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

$(A \leftrightarrow B) \rightarrow (A \rightarrow B)$

$(A \leftrightarrow B) \rightarrow (B \rightarrow A)$

$(A \leftrightarrow B) \rightarrow ((B \rightarrow A) \rightarrow (A \leftrightarrow B))$

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

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

with SUBSTITUTION for axioms (using $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$), and MODUS PONENS.

HILBERT SYSTEM FOR ¬,∧,∨,→,↔

CHALLENGE: Prove soundness for this system. Use induction.

We can prove completeness for the ¬,∧,∨,→,↔ system assuming completeness for the ¬,→ system.

Let A be a formula in ¬,∧,∨,→,↔. Let A* result from

replacing all ∧,∨,↔ in A with ¬,→ in the usual way.

CHALLENGE. Prove by induction that A and A* are logically equivalent.

CHALLENGE. Prove by induction that in the ¬,∧,∨,→,↔ system, A proves A* and A* proves A.

CHALLENGE. Finish the completeness proof for the ¬,∧,∨,→,↔ system.