CSC 438F/2404F **Notes** **Fall, 1999**

S. Cook

REFERENCES

The first two references have especially influenced these notes and are cited from time to time:

[Buss] Samuel Buss: Chapter I: An introduction to proof theory, in **Handbook of Proof Theory**, Samuel Buss Ed., Elsevier, 1998, pp1-78.

[B&M] John Bell and Moshe Machover: **A Course in Mathematical Logic**. North-Holland, 1977.

Other logic texts: The first is more elementary and readable.

[Enderton] Herbert Enderton: **A Mathematical Introduction to Logic**. Academic Press, 1972.

[Mendelson] E. Mendelson: **Introduction to Mathematical Logic**. Wadsworth & Brooks/Cole, 1987.

Computability text:

[Sipser] Michael Sipser: **Introduction to the Theory of Computation**. PWS, 1997.

Propositional Calculus

Throughout our treatment of formal logic it is important to distinguish between *syntax* and *semantics*. Syntax is concerned with the structure of strings of symbols (e.g. formulas and formal proofs) without regard to their meaning. Semantics is concerned with their meaning.

Syntax

Formulas are certain strings of symbols as specified below. In this chapter we use formula to mean *propositional formula*. Later the meaning of formula will be extended to *first-order formula*.

(Propositional) formulas are built from *atoms* $P_1, P_2, P_3, \ldots$, the *unary connective* $\neg$, the binary connectives $\wedge, \vee$, and parentheses (,). (The symbols $\neg, \wedge$ and $\vee$ are read "not", "and" and "or", respectively.) We use $P, Q, R, ...$ to stand for atoms. Formulas are defined recursively as follows:

Definition of Propositional Formula

1) Any atom P is a formula.
2) If A is a formula so is $\neg A$.
3) If A, B are formulas, so is $(A \wedge B)$.
4) If A, B are formulas, so is $(A \vee B)$.

All (propositional) formulas are constructed from atoms using rules 2) - 4).

Examples of formulas: P , $(P \vee Q)$, $(\neg(P \wedge Q) \wedge (\neg P \vee \neg Q))$.

We will use $\supset$ ("implies") and $\leftrightarrow$ ("is equivalent to") as abbreviations as follows:

$(A \supset B)$ stands for $(\neg A \vee B)$

$(A \leftrightarrow B)$ stands for $((A \supset B) \wedge (B \supset A))$

Unique Readability Theorem: (The grammar for generating formulas is unambiguous) Suppose A, B, A', B' are formulas, c and c' are binary connectives, and $(AcB) =_{syn} (A'c'B')$. Then $A =_{syn} A'$, $B =_{syn} B'$ and $c =_{syn} c'$.

Here we write $A =_{syn} A'$ instead of $A = A'$ to emphasize that A and A' are equal as strings of symbols (syntactic identity, rather than semantic identity). Note that $=_{syn}$ is a symbol of the "metalanguage" rather than the formal "object language".

Proof Assign weights

0	to $\neg$
1	to each binary connective $\wedge, \vee$
1	to (
-1	to)
-1	to each atom P.

Def'n The *Weight* of A is the sum of the weights of the symbols in A.

Lemma The weight of any formula is -1, but the weight of any proper initial segment is ≥ 0. (Hence no proper initial segment of a formula is a formula.)

Proof Structural induction on length of A. By *structural induction* we mean induction on the length of A, following the definition of propositional formula given above. The base case of the induction is the case in which A is an atom P. The lemma is obvious in this case. The induction step has one case for each of the three ways of constructing new formulas from simpler formulas, using $\neg, \wedge, \vee$. For example, in the case of $\wedge$, the task is to prove the lemma for $(A \wedge B)$, assuming (by the induction hypothesis) that the lemma holds for both A and B. We leave this as an exercise.

The Unique Readability Theorem follows from the Lemma.

In practice we will omit some of the parentheses in a formula when it does not cause ambiguity. We use the convention associativity to the right for $\wedge$ and $\vee$. For example,

$$(A_1 \vee A_2 \vee A_3 \vee A_4) \text{ stands for } (A_1 \vee (A_2 \vee (A_3 \vee A_4)))$$

Semantics

Def'n A *truth assignment* is a map $\tau : \{\text{atoms}\} \to \{T, F\}$.

(Here $\{T, F\}$ represents { true, false }). A truth assignment τ can be extended to assign either T or F to every formula, as follows:

1) $(\neg A)^\tau = T$ iff $A^\tau = F$

2) $(A \wedge \beta)^\tau = T$ iff $A^\tau = T$ and $\beta^\tau = T$

3) $(A \vee \beta)^\tau = T$ iff $A^\tau = T$ or $\beta^\tau = T$

Def'n τ *satisfies* A iff $A^\tau = T$; τ *satisfies* a set Φ of formulas iff τ satisfies A for all $A \in \Phi$. Φ is *satisfiable* iff some τ satisfies Φ; otherwise Φ is *unsatisfiable*. Similarly for A.

IMPORTANT DEFINITION $\Phi \models A$ (i.e. A is a *logical* consequence of Φ) iff τ satisfies A for every τ such that τ satisfies Φ.

Def'n A formula A is *valid* iff $\models A$ (i.e. $A^\tau = T$ for all τ). A valid propositional formula is called a *tautology*. We say that A and B are *equivalent* (written $A \Longleftrightarrow B$) iff $A \models B$ and $B \models A$.

Note that $\iff$ refers to semantic equivalence, as opposed to $=_{syn}$, which indicates syntactic equivalence. For example, $(P \vee Q) \iff (Q \vee P)$, but $(P \vee Q) \neq_{syn} (Q \vee P)$.

Proposition $\Phi \models A$ iff $\Phi \cup \{\neg A\}$ is unsatisfiable. Also A is a tautology iff $\neg A$ is unsatisfiable.

Proof: Immediate from the definitions of "unsatisfiable" and $\models$.

Examples:

The following are tautologies:
$P \vee \neg P$
$P \supset P$
$\neg(P \wedge \neg P)$
$(\neg P \vee ((P \wedge Q) \vee (P \wedge \neg Q))$

Logical consequence:
$(P \wedge Q) \models (P \vee Q)$

Equivalence:
$(P \vee Q) \iff (Q \vee P)$
$(P \wedge Q) \iff (Q \wedge P)$
$(P \wedge (Q \vee R)) \iff ((P \wedge Q) \vee (P \wedge R))$ ($\wedge$ distributes over $\vee$.)
$(P \vee (Q \wedge R)) \iff ((P \vee Q) \wedge (P \vee R))$ ($\vee$ distributes over $\wedge$.)
$\neg(P \vee Q) \iff (\neg P \wedge \neg Q)$ (De Morgan's Law)
$\neg(P \wedge Q) \iff (\neg P \vee \neg Q)$ (De Morgan's Law)
$(P \supset Q) \iff (\neg Q \supset \neg P)$ (contrapositive)

Formal Proofs (Gentzen Systems)

One way to establish that a formula A with n atoms is a tautology is to verify that $A^\tau = T$ for all 2^n truth assignments τ to to the atoms of A. A similar exhaustive method can be used to verify that A is a logical consequence of a finite set Φ of formulas. However another way is to use the notion of a formal proof, which may be both more efficient and more illuminating. A formal proof is a syntactic notion, in contrast to validity, which is a semantic notion. Many formal proof systems have been studied. The one we present here is the very elegant sequent calculus, introduced by Gerhard Gentzen in 1935 (see [Buss], section 1.2.1).

In the propositional sequent calculus system PK, each line in a proof is a *sequent* of the form

$$A_1, \ldots, A_k \rightarrow B_1, \ldots, B_\ell \tag{1}$$

where $\rightarrow$ is a new symbol (not to be confused with $\supset$), and $A_1, \ldots, A_k$ and $B_1, \ldots, B_\ell$ are sequences of formulas ($k, \ell \geq 0$) called *cedents*. We call the cedent $A_1, \ldots, A_k$ the *antecedent* and $B_1, \ldots, B_\ell$ the *succedent*.

The semantics of sequents is given as follows. We say that a truth assignment τ *satisfies* the sequent (1) iff either τ falsifies some A_i or τ satisfies some B_i. Thus the sequent is equivalent to the formula

$$(A_1 \wedge A_2 \wedge \ldots \wedge A_k) \supset (B_1 \vee B_2 \vee \ldots \vee B_\ell)$$

(In other words, the conjunction of the A's implies the disjunction of the B's.) In the cases in which the antecedent or succedent is empty, we see that the sequent $\rightarrow A$ is equivalent to the formula A, and $A \rightarrow$ is equivalent to $\neg A$, and just $\rightarrow$ (with both antecedent and succedent empty) is false (unsatisfiable). We say that a sequent is *valid* if it is true under all truth assignments (which is the same as saying that its corresponding formula is a tautology).

Examples: The following are valid sequents, for any formulas A, B:

$A \rightarrow A$
$\rightarrow A, \neg A$
$A, \neg A \rightarrow$
$\rightarrow A \vee \neg A$
$A, A \supset B \rightarrow B$

A formal proof (or just *proof*) in the propositional sequent calculus PK is a finite rooted tree in which the nodes are (labeled with) sequents. The sequent at the root (written at the bottom) is what is being proved, and is called the *endsequent*. The sequents at the leaves, written at the top, are *logical axioms*, and must be of the form $A \rightarrow A$, where A is a formula. Each sequent other than the logical axioms must follow from its parent sequent(s) by one of the following rules of inference (here Γ and Δ denote finite sequences of formulas).

weakening rules

$$\textbf{left}\ \frac{\Gamma \rightarrow \Delta}{A, \Gamma \rightarrow \Delta} \qquad\qquad \textbf{right}\ \frac{\Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta, A}$$

exchange rules

$$\textbf{left}\ \frac{\Gamma_1, A, B, \Gamma_2 \rightarrow \Delta}{\Gamma_1, B, A, \Gamma_2 \rightarrow \Delta} \qquad\qquad \textbf{right}\ \frac{\Gamma \rightarrow \Delta_1, A, B, \Delta_2}{\Gamma \rightarrow \Delta_1, B, A, \Delta_2}$$

contraction rules

$$\textbf{left}\ \frac{\Gamma, A, A \rightarrow \Delta}{\Gamma, A \rightarrow \Delta} \qquad\qquad \textbf{right}\ \frac{\Gamma \rightarrow \Delta, A, A}{\Gamma \rightarrow \Delta, A}$$

$\neg$ introduction rules

$$\textbf{left}\ \frac{\Gamma \rightarrow \Delta, A}{\neg A, \Gamma \rightarrow \Delta} \qquad\qquad \textbf{right}\ \frac{A, \Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta, \neg A}$$

$\wedge$ introduction rules

$$\textbf{left}\ \frac{A, B, \Gamma \rightarrow \Delta}{(A \wedge B), \Gamma \rightarrow \Delta} \qquad\qquad \textbf{right}\ \frac{\Gamma \rightarrow \Delta, A \quad \Gamma \rightarrow \Delta, B}{\Gamma \rightarrow \Delta, (A \wedge B)}$$

$\vee$ introduction rules

$$\textbf{left}\ \frac{A, \Gamma \rightarrow \Delta \quad B, \Gamma \rightarrow \Delta}{(A \vee B), \Gamma \rightarrow \Delta} \qquad\qquad \textbf{right}\ \frac{\Gamma \rightarrow \Delta, A, B}{\Gamma \rightarrow \Delta, (A \vee B)}$$

cut rule

$$\frac{\Gamma \rightarrow \Delta, A \quad A, \Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta}$$

Note that there is one **left** introduction rule and one **right** introduction rule for each of the three logical connectives $\wedge, \vee, \neg$. Further, these rules seem to be the simplest possible, given that the fact that in each case the bottom sequent is valid iff all top sequents are valid.

Exercise 1 *Write down each of the six introduction rules from memory.*

Note that repeated use of the exchange rules allows us to execute an arbitrary reordering of the formulas in the antecedent or succedent of a sequent. In presenting a proof in the system PK, we will usually omit mention of the steps requiring the exchange rules, but of course they are there.

As an example, we give a PK proof of one of DeMorgan's laws:

$$\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$$

To find this (or any) proof, it is a good idea to start with the conclusion at the bottom, and work up by removing the connectives one at a time, outermost first, by using the introduction rules in reverse. This can be continued until some atom P occurs on both the left and right side of a sequent. Then this sequent can be derived from the axiom $P \rightarrow P$ using weakenings and exchanges. The cut and contraction rules are not necessary, and weakenings are only needed immediately below axioms. (The cut rule can be used to shorten proofs, and contraction will be needed later for the predicate calculus.)

$$
\dfrac{
  \dfrac{\dfrac{P \rightarrow P}{P \rightarrow P, \neg Q}\,(weakening)}{\rightarrow P, \neg P, \neg Q}\,(\neg\ right)
  \qquad
  \dfrac{\dfrac{Q \rightarrow Q}{Q \rightarrow Q, \neg P}\,(weakening)}{\rightarrow Q, \neg P, \neg Q}\,(\neg\ right)
}{
  \dfrac{\dfrac{\rightarrow P \wedge Q, \neg P, \neg Q}{\rightarrow P \wedge Q, \neg P \vee \neg Q}\,(\vee\ right)}{\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q}\,(\neg\ left)
}\,(\wedge\ right)
$$

Exercise 2 *Give PK proofs for each of the following valid sequents:*

$\neg P \vee \neg Q \rightarrow \neg(P \wedge Q)$
$\neg(P \vee Q) \rightarrow \neg P \wedge \neg Q$
$\neg P \wedge \neg Q \rightarrow \neg(P \vee Q)$

Exercise 3 *Suppose that we allowed $\supset$ as a primitive connective, rather than one introduced by definition. Give the approprate left and right introduction rules for $\supset$.*

Now we prove that PK is both sound and complete. That is, a propositional sequent is provable in PK iff it is valid.

Soundness Theorem: Every sequent provable in PK is valid.

Proof: We show that the endsequent in every PK proof is valid, by induction on the number of sequents in the proof. For the base case, the proof is a single line; an axiom $A \rightarrow A$. This is obviously valid. For the induction step, one need only verify for each rule, if all top sequents are valid, then the bottom sequent is valid. $\square$

Completeness Theorem: Every valid propositional sequent is provable in PK without using cut or contraction.

Proof: The idea is discussed in the example proof above of DeMorgan's laws. We need to use the inversion principle.

Inversion Principle: For each PK rule except weakening and cut, if the bottom sequent is valid, then all top sequents are valid.

This principle is easily verified by inspecting each of the ten rules in question.

Now for the completeness theorem: We show that every valid sequent $\Gamma \to \Delta$ has a PK proof, by induction on the total number of logical connectives $\wedge, \vee, \neg$ occurring in $\Gamma \to \Delta$. For the base case, every formula in Γ and Δ is an atom, and since the sequent is valid, some atom P must occur in both Γ and Δ. Hence $\Gamma \to \Delta$ can be derived from the axiom $P \to P$ by weakenings and exchanges.

For the induction step, let A be any nonatomic formula (i.e. A is not an atom) in Γ or Δ. Then by the definition of propositional formula A must have one of the forms $(B \wedge C)$, $(B \vee C)$, or $\neg B$. Thus $\Gamma \to \Delta$ can be derived from $\wedge$ introduction, $\vee$ introduction, or $\neg$ introduction, respectively, using either the **left** case or the **right** case, depending on whether A is in Γ or Δ. In each case, each top sequent of the rule will have at least one fewer connective than $\Gamma \to \Delta$, and the sequent is valid by the inversion principle. Hence each top sequent has a PK proof, by the induction hypothesis. $\square$

Remark: The soundness and completeness theorems relate the *semantic* notion of validity to the *syntactic* notion of proof.

We generalize the (semantic) definition of logical consequence from formulas to sequents in the obvious way: A sequent S is a *logical consequence* of a set Φ of sequents iff every truth assignment τ that satisfies Φ also satisfies S. We generalize the (syntactic) definition of PK proof of a sequent S to a PK proof of S *from a set* Φ *sequents* by allowing sequents in Φ to be leaves (or nonlogical axioms) in the proof tree, in addition to the logical axioms $A \to A$. It turns out that soundness and completeness generalize to this setting.

Derivational Soundness and Completeness Theorem: A sequent S is a logical consequence of a set Φ of sequents iff there is a PK proof of S from Φ.

A remarkable aspect of completeness is that a finite proof exists even in case Φ is an infinite set. This is because of the compactness theorem (below) which implies that if S is a logical consequence of Φ, then S is a logical consequence of some finite subset of Φ.

Derivational soundness is proved in the same way as simple soundness: by induction on the number of sequents in the PK proof. In the previous proof we observed that if the top sequents of a rule are valid, then the bottom sequent is valid. Now we observe that the bottom sequent is a logical consequence of the top sequent(s).

To prove completeness, by compactness it suffices to consider the case in which Φ is a finite set.

In general, to prove S from Φ the cut rule is required. In particular, there is no PK proof of $\to P$ from $\to P \wedge Q$ without using the cut rule. To see this, we apply the following result.

Subformula Property: Every formula in every sequent in a PK proof without cut is a subformula of a formula in the endsequent.

This principle is proved by observing that for every rule other than cut, every formula on the top is a subformula of some formula on the bottom.

We illustrate the completeness theorem by proving the special case in which Φ consists of the single sequent $A \to B$. Assume that the sequent $\Gamma \to \Delta$ is a logical consequence of $A \to B$. Then both of the sequents $\Gamma \to \Delta, A$ and $B, A, \Gamma \to \Delta$ are valid (why?). Hence by the earlier completeness theorem, they have PK proofs π_1 and π_2. We can use these proofs to get a proof of $\Gamma \to \Delta$ from $A \to B$ as shown below, where the double line indicates several rules have been applied.

$$\dfrac{\dfrac{\vdots\, \pi_1}{\Gamma \to \Delta, A} \qquad \dfrac{\dfrac{A \to B}{\overline{\overline{A, \Gamma \to \Delta, B}}}\,(weakenings, exchanges) \qquad \dfrac{\vdots\, \pi_2}{B, A, \Gamma \to \Delta}}{A, \Gamma \to \Delta}\,(cut)}{\Gamma \to \Delta}\,(cut)$$

Next consider the case in which Φ has the form $\{\to A_1, \to A_2, ..., \to A_k\}$ for some set $\{A_1, ..., A_k\}$ of formulas. Assume that $\Gamma \to \Delta$ is a logical consequence of Φ in this case. Then the sequent

$$A_1, A_2, ..., A_k, \Gamma \to \Delta$$

is valid (why?), and hence has a PK proof π. Now we can use the assumptions Φ and the cut rule to successively remove $A_1, A_2, ..., A_k$ from the above sequent to conclude $\Gamma \to \Delta$. For example, A_1 is removed as follows:

$$\dfrac{\dfrac{\to A_1}{\overline{\overline{A_2, ..., A_k, \Gamma \to \Delta, A_1}}}\,(weakenings, exchanges) \qquad \dfrac{\vdots\, \pi}{A_1, A_2, ..., A_k, \Gamma \to \Delta}}{A_2, ..., A_k, \Gamma \to \Delta}\,(cut)$$

Exercise 4 *Prove the completeness theorem for the more general case in which Φ is any finite set of sequents.*

Propositional Compactness Theorem: We state three different forms of this result. All three are equivalent.

Form 1: If Φ is an unsatisfiable set of propositional formulas, then some finite subset of Φ is unsatisfiable.

Form 2: If a formula A is a logical consequence of a set Φ of formulas, then A is a logical consequence of some finite subset of Φ.

Form 3: If every finite subset of a set Φ of formulas is satisfiable, then Φ is satisfiable.

Exercise 5 *Prove the equivalence of the three forms. (Note that Form 3 is the contrapositive of Form 1.)*

Proof of Form 1: Let Φ be an unsatisfiable set of formulas, and let $P_1, P_2, P_3, \ldots$ be an infinite list including all atoms occurring in Φ. Organize the set of truth valuations into an infinite rooted binary tree B. Each node except the root is labelled with a literal P_i or $\neg P_i$. The two children of the root are labelled P_1 and $\neg P_1$, indicating that P_1 is assigned T or F, respectively. The two children of each of these nodes are labelled P_2 and $\neg P_2$, respectively, indicating the truth value of P_2. Thus each infinite branch in the tree represents a complete truth assignment, and each path from the root to a node represents a truth assignment to the atoms $P_1, \ldots, P_i$, for some i.

Now for every node ν in the tree B, prune the tree at ν (i.e. remove the subtree rooted at ν, keeping ν itself) if the partial truth assignment τ_ν represented by the path to ν falsifies some formula A_ν in Φ, where all atoms in A_ν get values from τ_ν. Let B' be the resulting pruned tree. Since Φ is unsatisfiable, every path from the root in B' must end after finitely many steps in some leaf ν labelled with a formula A_ν in Φ. If follows from König's Lemma below that B' is finite. Let Φ' be the finite subset of Φ consisting of all formulas A_ν labelling the leaves of B'. Since every truth assignment τ determines a path in B' which ends in a leaf A_ν falsified by τ, it follows that Φ' is unsatisfiable. □

König's Lemma: Suppose T is a rooted tree in which every node has only finitely many children. If every branch in T is finite, then T is finite.

Proof: We prove the contrapositive: If T is infinite (but every node has only finitely many children) then T has an infinite branch. We can define an infinite path in T as follows: Start at the root. Since T is infinite but the root has only finitely many children, the subtree rooted at one of these children must be infinite. Choose such a child as the second node in the branch, and continue. □

Exercise 6 (For those with some knowledge of set theory or point set topology) *The above proof of the propositional compactness theorem only works when the set of atoms is countable, but the result still holds even when Φ is an uncountable set with an uncountable set $\mathcal{A}$ of atoms. Complete each of the two proof outlines below.*

(a) Prove Form 3 using Zorn's Lemma as follows: Call a set Ψ of formulas *finitely satisfiable* if every finite subset of Ψ is satisfiable. Assume that Φ is finitely satisfiable. Let $\mathcal{C}$ be the class of all finitely satisfiable sets $\Psi \supseteq \Phi$ of propositional formulas using atoms in Φ. Order these sets Ψ by inclusion. Show that the union of any chain of sets in $\mathcal{C}$ is again in the class $\mathcal{C}$. Hence by Zorn's Lemma, $\mathcal{C}$ has a maximal element Ψ_0. Show that Ψ_0 has a unique satisfying assignment, and hence Φ is satisfiable.

(b) Show that Form 1 follows from Tychonoff's Theorem: The product of compact topological spaces is compact. The set of all truth assignments to the atom set $\mathcal{A}$ can be given the product topology, when viewed as the product for all atoms P in $\mathcal{A}$ of the two-point space $\{T, F\}$ of assignments to P, with the discrete topology. By Tychonoff's Theorem, this space of assignments is compact. Show that for each formula A, the set of assignments falsifying A is open. Thus Form 1 follows from the definition of compact: every open cover has a finite subcover.