Lecture 10: Propositional Logic

10:30 AM, Feb 26, 2009

Contents

1 Overview

This part of the course is devoted to **knowledge representation** and manipulation. Knowledge-based agents *know* something about their environment and they use their knowledge together with an inference engine to *reason* about their environment. **Logic** is a formalism in which to express knowledge and reasoning. This lecture is concerned with the syntax and semantics of propositional logic, Hilbert-style proofs and natural deduction, and modus ponens theorem provers.

2 Syntax

The **syntax** of a language are the rules that govern construction of its formulas, given an alphabet. The **alphabet** of propositional logic contains the **constants** $\{\top, \bot\}$, a set of propositional **variables** $\{P, Q, R, \ldots\}$, and a set of **connectives** $\{\neg, \vee, \wedge, \rightarrow\}$. The following inductive definition describes the **formulas** of propositional logic. **Complex** formulas involve one or more of the connectives; **atomic** formulas do not.

- the constants $\top$, $\bot$ are (atomic) formulas
- the propositional variables $P, Q, R, \ldots$ are (atomic) formulas
- if A and B are formulas, then (complex) formulas are formed as follows:
  - $\neg(A)$ is a formula
  - $(A \wedge B)$ is a formula
  - $(A \vee B)$ is a formula
  - $(A \rightarrow B)$ is a formula

In the absence of parentheses, the precedence of the connectives is as follows: $\neg$, $\wedge$, $\vee$, and $\rightarrow$.

Propositional variables represent **propositions** and connectives represent ways of combining propositions. For example, if P represents the proposition "the lights are on" and Q represents the proposition "the door is closed," then $P \wedge Q$ represents the formula "the lights are on *and* the door is closed," and $\neg P$ represents the formula "the lights are *not* on."

3 Semantics

The semantics of a language give meaning to its formulas. The formulas of propositional logic are assigned one of two values, either true (T) or false (F). Atomic formulas are assigned values by an **interpretation** $\mathcal{I} = \langle D, M \rangle$, where

- $D = \{$T,F$\}$ is the domain of meanings
- M maps atomic formulas into domain D *s.t.* $M(\top) =$ T and $M(\bot) =$ F

Logical formulas are evaluated by a semantic evaluation function v, which takes as input interpretation $\mathcal{I}$ and formula A, and outputs an element of domain D. We abbreviate $v(\mathcal{I}, A) =$ T by $\mathcal{I} \models A$ and $v(\mathcal{I}, A) =$ F by $\mathcal{I} \not\models A$. The former is read "$\mathcal{I}$ satisfies A" or "$\mathcal{I}$ is a model of A;" the latter is read inversely. Given interpretation $\mathcal{I}$ and formula A, $\mathcal{I} \models A$ is defined inductively as follows:

- $\mathcal{I} \models \top$ and $\mathcal{I} \not\models \bot$, since $M(\top) =$ T and $M(\bot) =$ F
- $\mathcal{I} \models P$ iff $M(P) =$ T
- $\mathcal{I} \models \neg A$ iff $\mathcal{I} \not\models A$
- $\mathcal{I} \models A \vee B$ iff $\mathcal{I} \models A$ or $\mathcal{I} \models B$
- $\mathcal{I} \models A \wedge B$ iff $\mathcal{I} \models A$ and $\mathcal{I} \models B$
- $\mathcal{I} \models A \rightarrow B$ iff $\mathcal{I} \models B$ whenever $\mathcal{I} \models A$

The inductive cases of the semantics are more precisely expressed as truth tables:

A	$\neg A$
T	F
F	T

A	B	$A \vee B$	$A \wedge B$	$A \rightarrow B$	$A \leftarrow B$
T	T	T	T	T	T
T	F	T	F	F	T
F	T	T	F	T	F
F	F	F	F	T	T

The semantics of propositional logic are **compositional**, since the truth or falsity of complex formulas in propositional logic is *entirely* determined by the truth or falsity of their constituents.

Remark: Logical implication is not equivalent in meaning to causal implication as it is used in natural language. Let A represent the formula "pigs can fly," and let B represent the formula

"Al Gore is the president of the United States." Now consider the formula $A \rightarrow B$. In natural language, one would interpret this formula to mean that the existence of flying pigs is one condition sufficient for Al Gore to be president of the United States. In propositional logic, however, $A \rightarrow B$ is vacuously true (since pigs cannot fly).

The following definitions pertain to the formula A:

- A is **satisfiable** iff there exists $\mathcal{I}$ *s.t.* $\mathcal{I} \models A$
- A is **valid** iff for all $\mathcal{I}$, $\mathcal{I} \models A$ iff there does not exist $\mathcal{I}$ *s.t.* $\mathcal{I} \not\models A$
- A is **unsatisfiable** iff for all $\mathcal{I}$, $\mathcal{I} \not\models A$ iff there does not exist $\mathcal{I}$ *s.t.* $\mathcal{I} \models A$

Unsatisfiable formulas are also called **contradictions**, or logical falsities: *e.g.*, $A \wedge \neg A$. There is no interpretation $\mathcal{I}$ *s.t.* $\mathcal{I} \models A \wedge \neg A$. On the other hand, valid formulas are also called **tautologies**, or logical truths, since they are true under all interpretations.

Examples of tautologies are listed below:

- Law of Double Negation
  $\neg\neg A \leftrightarrow A$
- DeMorgan's Laws
  $\neg(A \vee B) \leftrightarrow (\neg A \wedge \neg B)$
  $\neg(A \wedge B) \leftrightarrow (\neg A \vee \neg B)$
- Distributive Laws
  $(A \wedge B) \vee C \leftrightarrow (A \vee C) \wedge (B \vee C)$
  $(A \vee B) \wedge C \leftrightarrow (A \wedge C) \vee (B \wedge C)$
- Associative Laws
  $(A \vee B) \vee C \leftrightarrow A \vee (B \vee C)$
  $(A \wedge B) \wedge C \leftrightarrow A \wedge (B \wedge C)$
- Commutative Laws
  $A \vee B \leftrightarrow B \vee A$
  $A \wedge B \leftrightarrow B \wedge A$

Here, the connective $\leftrightarrow$ is used as an abbreviation: in particular, $A \leftrightarrow B$ means $(A \rightarrow B) \wedge (B \rightarrow A)$.

Exercise: Using truth tables, establish that these laws are in fact tautologies.

4 Logical Entailment

A knowledge-based agent is one equipped with a *knowledge base*—simply a set of sentences called **axioms**, because they are accepted as facts. The **logical entailment** problem is the following: given knowledge base KB and formula A, does the KB **semantically entail** A? Semantic entailment is defined as follows: in all interpretations in which the formulas of KB hold true, does A also hold true? If KB semantically entails A, this fact is denoted KB $\models A$.

A logic is **decidable** iff there exists an effective procedure for solving the logical entailment problem. Propositional logic is decidable. One effective procedure that decides logical entailment in the propositional case is enumeration of all possible interpretations using truth tables. But the complexity of this procedure is exponential in the number of propositional variables. Can we do better? Not unless P=NP, because A is valid iff $\neg A$ is not satisfiable, and satisfiability is NP-complete.

Example: Today we have class. If we have class, then it is either Tuesday or Thursday.[1] If it is Thursday, then we can relax; the weekend is approaching. But if it is Tuesday, then we can study or we can relax (if we are lazy). If we study, then afterwards, we can relax. Can we relax?

Let C represent "we have class today;" let T represent "it is Tuesday;" let H represent "it is Thursday;" let R represent "we can relax;" let S represent "we can study." Now the following list of axioms comprises our knowledge base: C, $C \rightarrow T \vee H$, $H \rightarrow R$, $T \rightarrow R \vee S$, and $S \rightarrow R$.

5 Logical Inference

An alternative approach to solving the logical entailment problem, and one that can be generalized to apply to more expressive logics than merely propositional logic (e.g., first-order logic), is to use proof-theoretic procedures. Here, we present two proof theories for propositional logic, a **Hilbert-style** calculus, which has many logical axioms but very few rules of inference, and a **Gentzen-style** calculus, which has few logical axioms but many rules of inference.

The following are examples of logical axioms, or **schema**: for all formulas A, B, and C,

1. $A \rightarrow (B \rightarrow A)$
2. $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
3. $(\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A)$

A **rule of inference** is a set of premises and a conclusion: *e.g.*, if $A \rightarrow B$ and A, then B. In this example, $A \rightarrow B$ and A are premises and B is the conclusion. A concrete instance of this inference rule is the following: given premises "if Socrates is a man, then Socrates is mortal," and "Socrates is a man," conclude "Socrates is mortal." This type of inference is called **modus ponens**.

A Hilbert-style proof is sequence of statements in a logical language such that each statement is either an instance of a logical axiom or an immediate consequence of some rule of inference applied to previous statements in the sequence. A **theorem** is the last statement in the sequence. Hilbert-style proofs are often cumbersome.[2] For example, the following is a Hilbert-style derivation:

1.	$P \rightarrow Q$	Assumption
2.	$Q \rightarrow R$	Assumption
3.	$(Q \rightarrow R) \rightarrow (P \rightarrow (Q \rightarrow R))$	Axiom 1
4.	$P \rightarrow (Q \rightarrow R)$	Modus Ponens 2,3
5.	$(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \rightarrow Q) \rightarrow (P \rightarrow R))$	Axiom 2
6.	$(P \rightarrow Q) \rightarrow (P \rightarrow R)$	Modus Ponens 4,5
7.	$P \rightarrow R$	Modus Ponens 1,6

[1] Possibly both.
[2] Recall high-school geometry.

C	T	H	S	R	C	$C \rightarrow T \vee H$	$H \rightarrow R$	$T \rightarrow R \vee S$	$S \rightarrow R$	R
T	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	T	T	T	F	T	T	F	T	F	F
T	**T**	**T**	**F**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	T	T	F	F	T	T	F	F	T	F
T	**T**	**F**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	T	F	T	F	T	T	T	T	F	F
T	**T**	**F**	**F**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	T	F	F	F	T	T	T	F	T	F
T	**F**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	F	T	T	F	T	T	F	T	F	F
T	**F**	**T**	**F**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
T	F	T	F	F	T	T	F	T	T	F
T	F	F	T	T	T	F	T	T	T	T
T	F	F	T	F	T	F	T	T	F	F
T	F	F	F	T	T	F	T	T	T	T
T	F	F	F	F	T	F	T	T	T	F
F	T	T	T	T	F	T	T	T	T	T
F	T	T	T	F	F	T	F	T	F	F
F	T	T	F	T	F	T	T	T	T	T
F	T	T	F	F	F	T	F	F	T	F
F	T	F	T	T	F	T	T	T	T	T
F	T	F	T	F	F	T	T	T	F	F
F	T	F	F	T	F	T	T	T	T	T
F	T	F	F	F	F	T	T	F	T	F
F	F	T	T	T	F	T	T	T	T	T
F	F	T	T	F	F	T	F	T	F	F
F	F	T	F	T	F	T	T	T	T	T
F	F	T	F	F	F	T	F	T	T	F
F	F	F	T	T	F	T	T	T	T	T
F	F	F	T	F	F	T	T	T	F	F
F	F	F	F	T	F	T	T	T	T	T
F	F	F	F	F	F	T	T	T	T	F

Figure 1: KB $\models R$: *i.e.*, R holds in all interpretations in which KB holds.

Gentzen (and later Prawitz) devised a formalism for expressing logical inference rules and proofs using tree (technically, graph) structures, rather than linear Hilbert-style derivations. This proof theory is called natural deduction. The rules of natural deduction come in introduction-elimination pairs. Those rules relevant to (classical) propositional logic are listed in Table 1.

Here is some evidence for the “naturality” of Gentzen and Prawitz’ system:

- The rule $\neg I$ can be understood as “proof by contradiction.”
- The rule $\vee E$ can be understood as case analysis.
- The rule $\rightarrow E$ is called modus ponens.

Introduction Rules	Elimination Rules
$\dfrac{}{\top}\ (\top\text{I})$	$\dfrac{\bot}{A}\ (\bot\text{E})$
$\dfrac{A \quad B}{A \wedge B}\ (\wedge\text{I})$	$\dfrac{A \wedge B}{A} \quad \dfrac{A \wedge B}{B}\ (\wedge\text{E})$
$\dfrac{A}{A \vee B} \quad \dfrac{B}{A \vee B}\ (\vee\text{I})$	$\dfrac{A \vee B \quad \begin{array}{c}[A]\\ \vdots \\ C\end{array} \quad \begin{array}{c}[B]\\ \vdots \\ C\end{array}}{C}\ (\vee\text{E})$
$\dfrac{\begin{array}{c}[A]\\ \vdots \\ B\end{array}}{A \to B}\ (\to\text{I})$	$\dfrac{A \quad A \to B}{B}\ (\to\text{E})$
$\dfrac{\begin{array}{c}[A]\\ \vdots \\ \bot\end{array}}{\neg A}\ (\neg\text{I})$	$\dfrac{\neg\neg A}{A}\ (\neg\text{E})$

Table 1: Rules of Natural Deduction.

- The rule $\neg E$ is called the law of double negation.
  Its presence indicates that the propostional logic under study is classical.[3]

The following is the above derivation carried out using natural deduction.

$$\dfrac{\dfrac{\dfrac{[P] \quad P \to Q}{Q}\ (\to \text{E}) \qquad Q \to R}{R}\ (\to \text{E})}{P \to R}\ (\to \text{I})$$

If there exists a derivation of theorem A from knowledge base KB using proof theory Π, we write KB $\vdash_{\Pi} A$. A proof theory Π is **sound** iff KB $\vdash_{\Pi} A$ implies KB $\models A$ (if A is provable, then it is true). A proof theory Π is **complete** iff KB $\models A$ implies KB $\vdash_{\Pi} A$ (if A is true, then it is provable). Both the Hilbert-style and the Gentzen-style calculi are sound and complete for propositional logic. Therefore, the logical entailment problem for propositional logic can be restated as a **logical inference** problem: e.g., given knowledge base KB and formula A, does KB $\vdash A$? For example, KB $\vdash_{\text{ND}} R$ (where ND denotes the proof theory of natural deduction), as follows:

[3] One alternative to classical logic is intuitionistic logic.

$$\dfrac{\dfrac{[R]}{R} \qquad \dfrac{[S] \quad S \to R}{R}\,(\to \mathrm{E}) \qquad \dfrac{\dfrac{[T] \quad T \to R \vee S}{R \vee S}\,(\to \mathrm{E}) \qquad \dfrac{\dfrac{[H] \quad H \to R}{R}\,(\to \mathrm{E})}{R \vee S}\,(\vee \mathrm{I}) \qquad \dfrac{C \quad C \to T \vee H}{T \vee H}\,(\to \mathrm{E})}{R \vee S}\,(\vee \mathrm{E})}{R}\,(\vee \mathrm{E})$$