Propositional Calculus (Part 1)

CS 4710 Course Notes

January 16, 2019

Introduction

"Rational construction" – the practice of constructing arguments with premises and a conclusion, where the connection between premises and conclusion is the "right sort."

Propositional calculus (also called "propositional logic" or "PC") is a basic model of rational persuasion, where we concern ourselves with premises that *logically imply* (i.e. guarantee the truth of) their conclusions.

As we will see, PC provides strong logical guarantees, but at the cost of what it can accomplish. There is often a tradeoff between the strength of a logical system's guarantees and the strength of the system itself (what it can prove).

Eventual goal: proofs of soundness and completeness

Our study of PC is aimed at proving two main "guarantees" of PC – soundness and completeness. For now we will state these informally.

Let "$P \vdash_{PC} Q$" be read as "Q is derivable from P in PC". Derivability, as we will see below, is a *syntactic* notion – it has to do with whether or not we can construct ("derive") Q from P with PC's inference rules, or "system of derivation".

"$P \models_{PC} Q$", on the other hand, is read "Q is entailed by P in PC". Entailment, in contrast to derivability, is a *semantic* notion – rather than dealing in symbols and inference rules, it has to do with being true in virtue of formal meaning (semantics).

Both $\vdash_{PC}$ and $\models_{PC}$ are different ways of filling out the somewhat vague notion of "follows from", one being tied to the syntax and rules of PC, and the other tied to its semantics. Soundness and completeness give us guarantees, in opposite directions, about the equivalence of these two notions. (Below we will use $\vdash$ in place of $\vdash_{PC}$ where the context is clear.)

Soundness. If $P \vdash Q \implies P \models Q$, then PC is *sound*.

Completeness. If $P \models Q \implies P \vdash Q$, then PC is *complete*.

For any system that is sound, derivability guarantees entailment. In other words, whatever we derive via syntactical rules is guaranteed to be true. For complete systems, entailment implies

derivability – all true statements are (eventually) recoverable by a finite derivation.

We will now proceed to develop the above, formally, for PC.

Set notation

Curly brackets "{" and "}" will be used to demarcate sets.

"ϕ" will be used to denote the empty set, such that $\phi = \{\}$.

"$\cup$" will denote set union, and "$\cap$" will denote intersection. For example, $\{0,1\} \cup \{1,2\} = \{0,1,2\}$, and $\{0,1\} \cap \{1,2\} = \{1\}$.

Language of propositional calculus (PC)

Def. 1 – A *statement* P is an assertion that can be true or false (informally speaking).

Def. 2 – An *argument* S consists of two finite sets of statements, called the *premises* and the *conclusion*. Alternatively, it can be viewed as a finite series of statements $P_1, P_2, \ldots, P_n, C$ where each P_i for $i \in \{1, \ldots, n\}$ is a premise, and C is the conclusion.

Def. 3 – An argument S is *valid* if its conclusion is true in **every case** its premises are true.

Notice the implicit modal notions in definition 3 – we are asked to imagine (or construct) eac phossible assignment of truth values to the premises. Validity is defined against a background of possibilities.

Notation – PC will include symbols for our truth values, true and false, namely "$\top$" and "$\bot$".

Def. 4 – The Principle of Bivalence. An *assignment* of a truth value to a statement must be one (and only one) member of $\{\top, \bot\}$.

Def. 5 – A *valuation* V is an assignment to each *atomic statement* in our *language*. We will define atomic statement and language below.

Def. 6 – A *language* L consists of a set of *atomic statements* (or "statement letters") A, a set of *logical operators* O, and a set of *formation rules* R.

A language tells us what our basic vocabulary is (the "atomic statements"), what connectives we have at our disposal (the operators), and what sort of recursive rule will generate the full set of statements.

Def. 7 – An *atomic statement* is a statement with no logical operators in it. It must be capable of being assigned any truth value (it must **not** be always true or always false under any assignment).

Def. 8 – A *logical operator* is a connective for statements. It has an arity (the number of

statements it functions over). For example, the connective "$\neg$" has arity one. If P is a statement, then $\neg P$ is an application of the operator $\neg$ to P.

We will now define the notion of a "well-formed formula" (WFF), which captures the notion of grammaticality for our system. WFF's are determined by the interaction of all components of a language – its atomic sentences, its operators, and its formation rules.

Let our set of operators be the single operator $O = \{\neg, \vee\}$, where $\vee$ has arity two. Let A refer to our set of atomic statements. We can lay out the formation rules (defining WFF's) in a recursive fashion:

Def. 9 – Well-formed formula.

1. If $x \in A$, x is a WFF.
2. If x and y are WFF's, then $\neg x$, $\neg y$, $x \vee y$, and $y \vee x$ are WFF's.
3. Nothing else is a WFF.

We say that WFF's are "closed under" the operators.

Semantics – truth tables

Our underlying semantics are "truth-functional" – in this section we'll get at what that means. Note above that given our definition of a language, we have a finite set of atomic statements **and** a finite set of truth values. This means that every possible valuation is enumerable; we call these "truth tables".

Example. Consider all of the possible valuations of a set of two statements, $S = \{A, B\}$.

A	B
$\top$	$\top$
$\top$	$\bot$
$\bot$	$\top$
$\bot$	$\bot$

For a set S write its cardinality (size) as $|S|$. Note that for a set of atomic statements S, the number of rows in our truth table is $2^{|S|}$. For a statement P and a valuation V, we write $[\![P]\!]^V$ to denote P's truth value under V. The principle of bivalence, then, can be restated as: for any WFF P and any assignment of meanings V, $[\![P]\!]^V \in \{\top, \bot\}$.

Truth-functionality and validity

For a WFF P, we say it is *complex* (or non-atomic) if $P \notin A$ (where A is our set of atomic statements). **Truth-functionality** says that the truth value of any complex statement is unambiguously determined by the truth values of its parts (the statements and operators it contains).

Def. 10 – An argument S is *valid* iff, for **every** valuation V and every $P \in S$, $[\![P]\!]^V = \top$.

Def. 11 – An argument S is *logically consistent* iff, for **some** valuation V and every $P \in S$,

$[\![P]\!]^V = \top$.

Def. 12 – An argument S is *logically inconsistent* iff it is **not** consistent. In other words, there does not exist a valuation V such that, for all $P \in S$, $[\![P]\!]^V = \top$.

Logical operators of PC

We will adopt the following logical operators in our treatment of PC: $\{\neg, \vee, \wedge, \rightarrow\}$, whose semantics are given in the truth table below.

P	Q	$P \wedge Q$	$P \vee Q$	$\neg P$	$P \rightarrow Q$
$\top$	$\top$	$\top$	$\top$	$\bot$	$\top$
$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$	$\top$	$\top$	$\top$
$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$

Expressivity and equivalence

Def. 12 – A set of operators O is *more expressive* than O' iff O' can be constructed using only the operators in O. This can be demonstrated using truth tables.

Example. Let $O' = \{\rightarrow\}$ and $O = \{\neg, \vee\}$. We can show O' is expressible by O in the following truth table, effectively showing that every statement including a $\rightarrow$ can be phrased equivalently with $\neg$ and $\vee$:

P	Q	$\neg P$	$\neg P \vee Q$	$P \rightarrow Q$
$\top$	$\top$	$\bot$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\top$	$\top$
$\bot$	$\bot$	$\top$	$\top$	$\top$

Note that P and Q are logically equivalent statements iff for every valuation V, $[\![P]\!]^V = [\![Q]\!]^V$.

Next session we'll cover the deduction theorem, soundness, completeness. If we have time, we will begin to cover predicate logic and model-theoretic semantics.