Propositional Logic

Benjamin Spector

Novemebr 2020

1 Syntax

Vocabulary

1. Atomic sentences (atoms for short): $p_1, p_2, \ldots$
2. Logical connectives: $\neg, \wedge, \vee, \rightarrow$ ($\wedge$, $\vee$, and $\rightarrow$ are the 'binary connectives')
3. Parentheses: (,)

Grammar

1. $At \rightarrow p_1|p_2|\ldots$
2. $S \rightarrow At|\neg S|(S \wedge S)|(S \vee S)|(S \rightarrow S)$

The well-formed formulae of propositional logic are all the strings of the symbols generated by this grammar. From now on, we simply say 'formula' or 'sentence' to refer to well-formed formulae.

Exercise : For each string, determine whether it is a formula

1. $\neg(p_7)$
2. $\neg(p_8 \vee p_9)$
3. $(p_8 \wedge (p_1 \vee p_7))$
4. $(p_1 \vee p_2 \vee p_3)$
5. $\neg p_7$
6. $\neg p_7 \wedge p_1$
7. $(p_1 \rightarrow (p2 \rightarrow (p_3 \rightarrow p_4))$
8. $(p_1 \rightarrow (p_2 \rightarrow (p_3 \rightarrow p_4)))$

Convention: omit most external parentheses.

The induction principle

(1) A set of strings is *closed* under logical connectives if:
for any two strings ϕ and ψ in Σ, $\ulcorner\neg\phi\urcorner$, $\ulcorner(\phi \wedge \psi)\urcorner$, $\ulcorner(\phi \vee \psi)\urcorner$, $\ulcorner(\phi \rightarrow \psi)\urcorner$ also belong to Σ.

(2) The set of well-formed formulae is the smallest set Σ of strings which contains all atoms and is closed under logical connectives.

(3) A consequence: If a set of formulae includes all atoms and is closed under logical connectives, it is the set of all well-formed formulae

Inductive proofs

In order to show that a certain property P holds of all formulae, it is sufficient to show that it holds of all the atoms, and that, for every ϕ, ψ, if it holds of ϕ and ψ, then it holds of $\ulcorner\neg\phi\urcorner$, $\ulcorner(\phi \wedge \psi)\urcorner$, $\ulcorner(\phi \vee \psi)\urcorner$, $\ulcorner(\phi \to \psi)\urcorner$.

Exercise

Prove that every formula contains an equal number of left parentheses and right parentheses.

Exercise

Given a sentence S, let $a(S)$ be the number of atoms that occur in S, and let $b(S)$ be the number of binary connectives that occur in S. Prove that for any sentence S, $a(S) = b(S) + 1$.

Exercise (harder)

Given a formula ϕ, a subformula of ϕ is a substring of ϕ which is itself a formula.

Let ϕ be a formula which contains no negation. Let n be the number of occurrences of atomic sentences in a sentence ϕ. Prove (by induction) that the number of subformulae of ϕ is $2n - 1$.

2 Semantics of Propositional Logic

2.1 Truth tables

A	¬ A
1	0
0	1

A	B	(A ∧ B)
1	1	1
1	0	0
0	1	0
0	0	0

A	B	(A ∨ B)
1	1	1
1	0	1
0	1	1
0	0	0

A	B	A → B
1	1	1
1	0	0
0	1	1
0	0	1

Computing the truth-table of a complex formula
On the board.

Discussion: material implication

Exercise: exclusive disjunction

Let us add the following exclusive disjunction connective to our language, $\veebar$, with the following truth-table.

A	B	A $\veebar$ B
1	1	0
1	0	1
0	1	1
0	0	0

1. Show that $(a \veebar (b \veebar c))$ is true if and only if exactly 1 or exactly 3 of $\{a, b, c\}$ are true.

2. Show that a sentence of the form $(\alpha_i \veebar (\alpha_2 \veebar (\ldots \veebar \alpha_n) \ldots)$, where $\alpha_1, \ldots \alpha_n$ are atomics sentences, is true if and only if an odd number of atomic sentences in it are true.

3. Discussion: is there a connective in English or French which expresses exclusive disjunction?

2.2 Formal Definition

A *valuation* v is a function from formulae to $\{0, 1\}$ such that:

(4) For any two formulas ϕ and ψ

a. $v(\neg\phi) = 1 - v(\phi)$
b. $v((\phi \wedge \psi) = min(v(\phi), v(\psi))$
c. $v((\phi \vee \psi) = max(v(\phi), v(\psi))$
d. $v((\phi \rightarrow \psi)) = max(1 - v(\phi), \psi)$

Theorem: If two valuations v_1 and v_2 give the same value to every atomic sentence, then they are identical. In other terms, a valuation is fully determiner by the values it assigns to atomic sentences.

Proof.
Let v_1 and v_2 be two valuations such that for every atom p, $v_1(p) = v_2(p)$. We prove by induction that for every formulae ϕ, $v_1(\phi) = v_2(\phi)$.

1. Base case: for every atomic sentence p, $v_1(p) = v_2(p)$. True by hypothesis.

2. Inductive hypothesis: Assume that for ϕ and ψ, we have $v_1(\phi) = v_2(\phi)$ and $v_1(\psi) = v_2(\psi)$.

3. Inductive step

   (a) $v_1(\neg\phi) = 1 - v_1(\phi) = 1 - v_2(\phi) = v_2(\phi)$
   (b) $v_1((\phi \wedge \psi)) = min(v_1(\phi), v_1(\psi)) = min(v_2(\phi), v_2(\psi)) = v_2((\phi \wedge \psi))$
   (c) Similarly for the two other connectives.

Remarks on notations

$v(\phi) = 1$ is often notated as $v \vDash \phi$, $[\![\phi]\!]^v = 1$, or $\phi(v) = 1$.

Valuations and propositions

One can look at each valuation as representing a possible state of the world.

For every sentence, we can look at the *set of valuations that it makes true*. We can think as such sets as being the *meaning* of a propositional logic sentence. I will call such sets *propositions*

(5) If ϕ is a sentence, the *proposition* expressed by ϕ, noted $[\phi]$, is the set of all valuations v such that $v(\phi) = 1$

(6) Let W be the set of all valuations. Then, for any ϕ, ψ:

a. $[\neg\phi] = W - [\phi]$.

b. $[\phi \wedge \psi] = [\phi] \cap [\psi]$

c. $[\phi \vee \psi] = [\phi] \cup [\psi]$

d. $[\phi \rightarrow \psi] = (W - [\phi]) \cup [\psi]$

Discussing material implication again.

3 Satisfiability, Logical Consequence, Tautologies and Contradiction

Satisfiability

A valuation v *satisfies* a set of sentence Σ if for every sentence ϕ in Σ, $v(\phi) = 1$.

A set of formulae Σ is said to be **satisfiable** (or **consistent**[1]) if there is a valuation v that satisfies it.

A set of formulae is said to be **contradictory** (or **inconsistent**) if it is not satisfiable.[2]

Exercises

1. (a) Is the set $\{p \vee q, \neg p\}$ satisfiable?
   (b) Is the set $\{p \vee q, \neg p, \neg(q \vee r)\}$ satisfiable?
   (c) Is the set $\{p, p \rightarrow \neg p\}$ satisfiable?
   (d) Is the set $\{p \rightarrow q, q \rightarrow \neg p\}$ satisfiable?.
   (e) Is the set $\{p, p \rightarrow q, q \rightarrow \neg p\}$ satisfiable?.

2. Find a set Σ of 5 formulae such that a) Σ is contradictory, but b) every proper subset of Σ with 4 members is satisfiable.

3. Consider the infinite set of formulae Σ defined as follows:
   $\Sigma = \{p_1 \rightarrow p_2, p_2 \rightarrow p_3, p_3 \rightarrow p_4, \ldots, p_i \rightarrow p_{i+1}, \ldots\}$.
   Characterize all the valuations that satisfy Σ.

Logical Consequence

A formula ϕ is a logical consequence of a set of formulae Σ if every valuation v that satisfies Σ is such that $v(\phi) = 1$.

Notation: $\Sigma \vDash \phi$

One also says that Σ *entails* ϕ.
Fact: $\Sigma \vDash \phi$ if and only if $\Sigma \cup \{\neg\phi\}$ is contradictory.

[1]The notions of 'consistency' and 'satisfiability' are in fact distinct. Satisfiability is a semantic notion: a set of formulae is satisfiable is there is a valuation that makes it true. Consistency is a *proof-theoretic* notion. We can define various *proof procedures* for propositional logic, i.e. rules of symbolic manipulations that apply to a set of premises and yield conclusions. A set is *consistent* relative to a proof procedure if a contradiction cannot be proved from it. For well-designed proof systems, in the case of propositional logic, *consistency* turns out to be equivalent to *satisfiable* - this result is known as a *completeness theorem.*

[2]See previous footnote.

Exercise

Determine whether the following statements are true

1. $p \vDash p \vee q$
2. $p, (q \rightarrow p) \vDash q$
3. $p, \neg p \vDash q$
4. $(p \rightarrow q), \neg q \vDash \neg p$
5. $(p \rightarrow q), \neg q \vDash p$
6. $(p \wedge q) \rightarrow r \vDash p \rightarrow q$
7. $(p \vee q) \rightarrow r \vDash p \rightarrow r$

Some properties of the logical consequence relation

- Reflexivity
- Transitivity
- Monotonicity.

Tautologies, validities, contradictions

ϕ is said to be logically valid, or a *tautology*, if and only if for every valuation v, $v(\phi) = 1$.

Notation: $\vDash \phi$

A tautology is entailed by the empty set (which justified the notation).
Example: $(p \vee \neg p)$

ϕ is a *contradiction* if for every valuation v, $v(\phi) = 0$.
Example: $(p \wedge \neg p)$

Often, the propositional language is enriched by two additional atomic sentences standing for 'contradiction' and 'tautology' whose interpretation does not vary across valuations:

Tautology symbol: $\top$
Contradiction symbol: $\bot$

When the language is so enriched, valuations are defined as functions v from atomic sentences to $\{0, 1\}$ such that $v(\top) = 1$ and $v(\bot) = 0$.

Exercise

For each formula above, detemine whether it is a tautology, a contradiction, or neither.

(7)
a. $(p \vee \neg p) \vee q$
b. $(p \vee \neg p)$
c. $\neg(p \wedge q) \rightarrow \neg q$
d. $\neg(p \vee q) \rightarrow \neg q$
e. $((p \vee q) \rightarrow r) \rightarrow (p \rightarrow r)$
f. $((p \wedge q) \rightarrow r) \rightarrow (p \rightarrow r)$
g. $\neg p \rightarrow (p \rightarrow q))$

h. $(p \to \neg p) \to p$
i. $(p \to \neg p) \to \neg p$

A useful technique: see what it would take for the sentence to be *false*, and what it would take for it to be *true*.

Equivalence

ϕ and ψ are equivalent if $\phi \vDash \psi$ and $\psi \vDash \phi$.

Equivalently: ϕ and ψ are equivalent if $[\phi] = [\psi]$

Note that: $(\top \wedge \phi)$ and $(\bot \vee \phi)$ are equivalent to ϕ, for any ϕ.

Notation: $\phi \equiv \psi$

Material Equivalence sign

The signs $\leftrightarrow$ is typically an *object language connective* that can be added to the logical vocabulary, together with the following semantic rule: $v(\phi \leftrightarrow \psi) = 1$ iff $v(\phi) = v(\psi)$. We then have:

Be careful not to confuse this object-language connective with the meta-language sign for equivalence.

Substitution theorems

1. Substitution *salva-veritate* preserves truth-value. Let F be a formula and p be an atom and σ be a function that maps every formula ϕ to the formula ϕ' that results from replacing every occurrence of p in ϕ with F. Then, for every valuation v and every sentence ϕ, if $v(F) = v(p)$, then $v(\sigma(\phi)) = v(\phi)$.

2. Uniform substitution preserves equivalence. Let F be a formula and p be an atom and σ be a function that maps every formula ϕ to the formula ϕ' that results from replacing every occurrence of p in ϕ with F. Then, if $\phi \equiv \psi$, then $\sigma(\phi) \equiv \sigma(\psi)$

Importantly, the reverse statement is not generally true: starting from two non-equivalent sentences ϕ and ψ, it can be that $\sigma(\phi)$ and $\sigma(\psi)$ are equivalent.

<u>Exercise:</u> Find an example of two sentences which are not equivalent but become equivalent after uniform subsitution of a certain atom with a given formula.

Strong compositionality

Compositionality: the meaning of a complex sentence is a function of the meaning of its parts and the way they are combined.

Strong Compositionality: the meaning of a constituent is a function of the meaning of its immediate subconstuents and the way they are combined.
Strong Compositionality amounts to the substituability of synonyms.

The substitution theorem establishes that propositional logic satisfies strong compositionality, where synonymy is viewed as equivalence.

Theorems and metatheorems

When we talk about a 'theorem of propositional logic', we typically think of a propositional logic statement that is logically valid. In general, a 'theorem' within a formalized system is a well-formed formula of some formal language which follows from the *axioms* of the formal system (a distinguished set of formulae) together with some *inference rules.*

The substitution theorem above is not a 'theorem of propositional logic', it is a theorem *about* propositional logic - a *metatheorem.*

4 Some important semantic properties

4.1 The deduction theorem

The sign '$\rightarrow$' is an **object-language** sign, while the sign $\models$ is a **meta-language** sign. There is however an interesting relationship between the two.
$\Sigma, \phi \models \psi$ if and only if $\Sigma \models (\phi \rightarrow \psi)$

Likewise, if we add $\leftrightarrow$ to our language:

$\phi \equiv \psi$ iff $\models (\phi \leftrightarrow \psi)$

4.2 Useful facts

Ex Falso Quodlibet Sequitur

Every formula is entailed by the contradiction:

For every ϕ, $(p \wedge \neg p) \models \phi$, $\bot \models \phi$.

Everything entails the tautology

The tautology is entailed by every formula (or set of formulae):

For every set of formulae Σ, $\Sigma \models (p \vee \neg p)$, $\Sigma \models \top$.

Contraposition

$\phi \models \psi$ if and only if $\neg\psi \models \neg\phi$

Modus Ponens

$(\phi \rightarrow \psi), \phi \models \psi$

Using the deduction theorem and Modus Ponens to prove validities
Application to $((a \vee b) \rightarrow c) \rightarrow (a \rightarrow c)$

4.3 Some equivalences

De Morgan's laws

(8) For every formulae ϕ and ψ
  a. $\neg(\phi \vee \psi) \equiv \neg\phi \wedge \neg\psi$
  b. $\neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi$

Distributivity

(9) a. $A \lor (B \land C) \equiv (A \lor B) \land (A \lor C)$
b. $A \land (B \lor C) \equiv (A \land B) \lor (A \land C)$

Duality

Let ϕ be a formula whose only binary connectives are $\land$ and $\lor$. The *dual* of ϕ, which will note ϕ^D is the formula obtained from ϕ by replacing every occurence of $\land$ by $\lor$, and vice-versa.

Theorem.
For any two sentences ϕ and ψ whose only binary connectives are $\land$ and $\lor$,
$\phi \equiv \psi$ if and only if $\phi^D \equiv \psi^D$.

Informal proof

In the truth-table for disjunction, reversing all the 1s and 0s yields the truth-table for conjunction. And in the truth-table for conjunction, reversing all the 1s and 0s yields the truth-table for disjunction (except for the order of rows, which is irrelevant).

Let ϕ and ψ be formulae whose only binary connectives are $\lor$ and $\land$. The truth table for their duals ϕ^D and ψ^D can be obtained from those of ϕ and ψ by simply reversing all the 1s and 0s, and vice versa. So if ϕ and ψ are equivalent, i.e. if their truth-tables have the same last column, so are ϕ^D and ψ^D (and vice-versa).

Generalized De Morgan's laws

Let ϕ be a sentence of the form $\neg\psi$. Now define ϕ' as the sentence obtained from ψ by replacing each atomic sentence in ψ by its negation, and each occurrence of $\land$ by $\lor$, and vice versa. Then $\phi \equiv \phi'$.

Informal proof on the board.

(10) a. $\neg(a \lor (\neg b \land \neg(c \land \neg d)))$
b. $\neg a \land (\neg\neg b \lor \neg(\neg c \lor \neg\neg d))$
c. $\neg a \land (\neg\neg b \lor (\neg\neg c \land \neg\neg\neg d))$
d. $\neg a \land (b \lor (c \land \neg d))$

5 Expressive power of propositional logic, Functional Completeness

5.1 Interdefinability of connectives

Suppose we eliminate $\to$ from the language of propositional logic $\mathcal{L}$. Then the resulting language $\mathcal{L}'$. would remain *as expressive* as before in the following sense: for every sentence in $\mathcal{L}$, there is an equivalent sentence in $\mathcal{L}'$. This is so because for any two formulae ϕ and ψ, $(\phi \to \psi) \equiv (\neg\phi \lor \psi)$.

Likewise, the languages whose only connectives are $\{\neg, \lor\}$ or $\{\neg\land\}$ are as expressive as $\mathcal{L}'$.

Exercise: show this!

5.2 Scheffer stroke

Let us introduce a new connective, known as the Scheffer stroke (or 'nand'), noted |, defined by the following truth-table:

A	B	A | B
1	1	0
1	0	1
0	1	1
0	0	1

Exercise
1. Paraphrase $A|B$ using plain English.
2. Show that the language whose only connective is '|' is as expressive as the one based on $\{\neg, \vee, \wedge, \rightarrow\}$.
3. Consider now a connective that could be paraphrased in English by 'Neither A nor B'. Write a truth-table for it. Show that the language whose only connective is the one you have just defined is as expressive as the one based on $\{\neg, \vee, \wedge, \rightarrow\}$.

5.3 Boolean functions and functional completeness

A truth table for a complex formula ϕ which contains, say, 3 atoms, has the following format (ignoring intermediate lines):

p_1	p_2	p_3	ϕ
1	1	1	0
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	1

We can view such a table as representing a *function* which takes as an argument an ordered set of truth-values and return a truth-value. Here, the function f would be defined by:

– $f(1,1,1) = f(1,0,0) = f(0,1,0) = f(0,0,1) = 0$
– $f(1,1,0) = f(1,0,1) = f(0,1,1) = f(0,0,0) = 1$.

Now we can ask: is there a formula that has such a truth-table. The answer is **yes!**.

Building a formula whose truth-table is the one above

1. for each line of the truth-table which has 1 on the last column, write the conjunction of three statements which together fully describe the line. For instance, the 3rd line would go to $(p_1 \wedge (\neg p_2 \wedge p_3))$
2. Then combine each of these conjunctive sentences with disjunction

Convention: omit parentheses when they do not matter: $(A \wedge (B \wedge C)) \rightsquigarrow (A \wedge B \wedge C)$

We get: $(p_1 \wedge p_2 \wedge \neg p_3) \vee (p_1 \wedge \neg p_2 \wedge p_3) \vee (\neg p_1 \wedge p_2 \wedge p_3) \vee (\neg p_1 \wedge \neg p_2 \wedge \neg p_3)$

Now, this formula is true if and only if one of the conjunctive statement is true, i.e. if the truth-values of p_1, p_2, p_3 are assigned as in one of the lines in which the truth table yields 1.

It is clear that the procedure above will work for *any* truth-table. Therefore, for any truth-table (reduced to its first columns corresponding to atoms and the final column), there is a formula that corresponds to it.

Boolean function associated with a formula

An n-ary Boolean function is a function that takes as arguments an ordered set of n truth-values (sequence of 0s and 1s), and returns a truth-value.

If ϕ is a formula in which n atoms occur. Let us arrange the atoms that occur in increasing order of indices: $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$.

Then the Boolean function associated with ϕ is the unique n-ary Boolean function f_ϕ such that for every valuation v, $f_\phi(v(\alpha_1), v(\alpha_2), \ldots, v(\alpha_n)) = v(\phi)$

Functional Completeness

A systematic generalization of the above informal reasoning (showing how to build a formula corresponding to a truth-table) yields the following result:

(11) **Functional Completeness of Propositional Logic**.
For every n, for every n-ary boolean function f, there exists a formula ϕ such that $f_\phi = f$.

Functionally Complete Sets of connectives

A set of connectives (associated with a semantics) is functionally complete if the propositional language based on them enjoys functional completeness. For instance, the language whose only connective is $|$ is functionally complete.

Exercises

A.

1. Consider the language based on $\{\wedge, \vee\}$
   Show that for any formula ϕ, the valuation that maps every atom to 1 maps ϕ to 1/

2. Is this language functionally complete?

B. [Difficult]

1. Consider the language based on $\{\neg, \veebar\}$ (remember that $\veebar$ is exclusive disjunction).
   Show that for every formula ϕ, the last column of its truth-table contains an even number of 1s.

2. Deduce from this that this language is not functionally complete.

5.4 Disjunctive and Conjunctive Normal Forms

Disjunctive Normal Forms

The set of **Disjunctive Normal Forms** (DNF) is a subset of the language of propositional logic, namely the one generated by the following grammar:

1. $L \rightarrow At | \neg At$

2. $D \rightarrow L | (L \wedge D)$

3. $S \rightarrow D | (D \vee S)$

Example (with some parentheses eliminated): $(p_1 \wedge \neg p_2) \vee (\neg p_2 \wedge p_7 \wedge p_1 6) \vee p_8$.

Theorem
Every sentence of propositional logic is equivalent to a Disjunctive Normal Form.

Conjunctive Normal Forms

The set of **Conjunctive Normal Forms**(CNF) is a subset of the language of propositional logic, namely the one generated by the following grammar:

1. $L \to At | \neg At$
2. $D \to L | (L \vee D)$
3. $S \to D | (D \wedge S)$

Theorem
Every sentence of propositional logic is equivalent to a Conjunctive Normal Form.

Proof
Let ϕ be a sentence. By the DNF theorem above, the sentence $\neg\phi$ is equivalent to a DNF ψ. By the generalized De Morgan's law, $\neg\psi$ is equivalent to a CNF. Since $\neg\psi$ is equivalent to ϕ, ϕ is equivalent to a CNF.

5.5 Expressive limitations of propositional logic

Despite the fact that every Boolean function is expressible in propositional logic, it is not the case that every *proposition* is expressible in propositional logic.

More specifically, it is not the case that for every set of valuations V, there exists a set of sentences Σ (even an infinite one) which is satisfied by all valuations of V and no other valuation.

An example of such a proposition would be the one that can be expressed in the metalanguage as *At least one of the atomic sentences is true*. It can be proved (but we don't do it here) there is no set of formulae which are satisfied by all valuations that make at least one atom true and no others.