Propositional Logic and Semantics

English is naturally *ambiguous*. For example, consider the following *employee (non)recommendations* and their *ambiguity* in the English language:

- *"I can assure you that no person would be better for the job."*
- *"All in all, I cannot say enough good things about this candidate or recommend him too highly."*

Goal: We want to be able to write *formal boolean expressions* such that there is no *ambiguity*.

For example, $p \to q \to r$ means $(p \to q) \to r$ or $p \to (q \to r)$?

Propositional Formulas

- Formal expressions involving *conjunctions* and *propositional variables*.
- We denote this *set* by $\mathcal{F}_{\mathcal{PV}}$ or simply $\mathcal{F}$, and define $\mathcal{F}$ *inductively*.

Slight Diversion - Defining Sets Inductively

Defining Sets Inductively

What does the following *definition* construct?

Let X be the smallest set such that:

Basis: $0 \in X$

Inductive Step: if $x \in X$ then $x + 1 \in X$.

X is $\mathbb{N}$

Q: How could we define the integers, $\mathbb{Z}$?

Let $\mathbb{Z}$ be the smallest set containing:

Basis: $0 \in \mathbb{Z}$

Inductive Step: if $x \in \mathbb{Z}$ then $x+1 \in \mathbb{Z}$ and $x-1 \in \mathbb{Z}$.

Q: How abou the *rationals*, $\mathbb{Q}$?

Basis: $0 \in \mathbb{Q}$

Inductive Step: if $x, y \in \mathbb{Q}$

1. $x+1 \in \mathbb{Q}$
2. $x-1 \in \mathbb{Q}$
3. $\frac{x}{y} \in \mathbb{Q}$ where $y \neq 0$.

Q: How abou the *language of arithmetic*, $\mathcal{L}A$?

Let $\mathcal{LA}$ be the smallest set such that:

Basis: $\mathbb{Q} \in \mathcal{LA}$

Inductive Step: Suppose that $x, y \in \mathcal{LA}$ then

1. $(x+y) \in LA$ $\quad (5+2)10$
2. $(x-y) \in LA$
3. $(x*y) \in LA$
4. $(x \div y) \in LA$

Why define sets by induction?

Consider the following conjecture:

Let e be an element of $\mathcal{LA}$.

variables eg. x, y,

Let $vr(e)$ represent the number of characters in e.

Let $op(e)$ represent the number of *operations*, ie., characters from $\{+, -, *, \div\}$ in e.

CLAIM 1: *Let P(e) be "vr(e) = op(e) + 1". Then* $\forall e \in \mathcal{LA}$, $P(e)$.

We can *prove* this using a special version of induction called *structural induction.*

CLAIM 1: *Let P(e) be "vr(e) = op(e) + 1". Then* $\forall e \in \mathcal{LA}, P(e)$.

We can *prove* this using a special version of induction called *structural induction.*

Proof. STRUCTURAL INDUCTION on e:

1. **Basis:** Suppose $e \in \mathbb{Q}$, then $vr(e) = 1$, $op(e) = 0$ so $P(e)$ holds.

2. **Induction Step:** Assume that $P(e_1)$ and $P(e_2)$ are *true* for arbitrary expressions in $\mathcal{LA}$. Let $e = e_1 \oplus e_2$ where $\oplus \in \{+, -, *, \div\}$.

   $op(e_1)$ $op(e_2)$

   Then, $vr(e) = vr(e_1) + vr(e_2)$

   by Structural induction → $= op(e_1)+1+op(e_2)+1$

   $= op(e)+1$ ← by def'n.

∎

Let's define Propositional Logic Using Structural Induction:

$\mathcal{F}_{\mathcal{PV}}$ is the smallest set such that:

Base Case:

- **true** and **false** belong to $\mathcal{F}_{\mathcal{PV}}$, and if $p \in PV$ then $p \in \mathcal{F}_{\mathcal{PV}}$.

Induction Step: If p and $q \in \mathcal{F}_{\mathcal{PV}}$, then so are

- **NEGATION**: $\neg p$
- **CONJUNCTION**: $(p \wedge q)$
- **DISJUNCTION**: $(p \vee q)$
- **CONDITIONAL**: $(p \to q)$
- **BICONDITIONAL**: $(p \leftrightarrow q)$

A formula in $\mathcal{F}_{\mathcal{PV}}$ is *uniquely* defined, i.e., there is no *ambiguity*. (see the **Unique Readibility Theorem** in the notes.)

Q: What happens when a *propositional formula* is quite *complex*? such as,

$$(((p \wedge y) \vee (q \to (r \wedge t))) \wedge \neg(s \wedge (u \vee (v \vee (x \vee z)))))$$

This has lead to *conventions* that define an *informal* notation that uses less brackets.

Bracketing Conventions

1. drop the *outer most parenthesis* e.g,

   $(x \vee y)$ it's ok to do $x \vee y$

2. give $\wedge$ and $\vee$ precedence over $\rightarrow$ and $\leftrightarrow$ (like $\times, +$ *vs.* $<, =$ in *arithmetic*) eg.,

   $(x \wedge y) \rightarrow (z \vee p)$ equiv $x \wedge y \rightarrow z \vee p$

3. give $\wedge$ precedence over $\vee$ (similar to $\times$ *vs.* $+$ in *arithmetic*) eg.,

   $(3 \times 2) + 5$

   $p \wedge q \vee r$ $\quad (p \wedge q) \vee r$

4. *group* from the *right* when the *same connective* appears *consecutively*, eg.,

   $p \rightarrow q \rightarrow r$ equiv $p \rightarrow (q \rightarrow r)$

Q: Using these *conventions*, how can

$$(((p \wedge y) \vee (q \rightarrow (r \wedge t))) \wedge \neg(s \wedge (u \vee (v \vee (x \vee z)))))$$

be *simplified*?

$(p \wedge y \vee (q \rightarrow r \wedge t)) \wedge \neg(s \wedge (u \vee v \vee x \vee z))$

The Meaning of τ

Q: What is the difference between a *propositional formula* and a *propositional statement*?

propositional formula is syntactic.

once we give variables a value of true or false we have a semantic statement.

Therefore we need a method to determine the *truth value* of a *statement* from the *truth values* assigned to the *propositional variables*.

- Let τ be a *truth assignment*, ie., a function.

$$\tau : PV \to \{\mathbf{true}, \mathbf{false}\}.$$

- If $p \in PV$ and τ assigns **true** to p, then we write

$$\tau(p) = \mathbf{true}.$$

- How does τ affect a *propositional statement*?

→ gives the statement meaning.

- We need a *function* that behaves like τ, but *operates* on *propositional statements*.

- Let $\underbrace{\tau^* : \mathcal{F}_{\mathcal{PV}} \to \{\mathbf{true}, \mathbf{false}\}}$. What does this mean?

τ^* takes as input a propositional statement and return true or false.

We formally define τ^* using *structural induction*:

Let $\mathbf{Q}, \mathbf{P} \in \mathcal{F}_{\mathcal{PV}}$.

Base Case: $P \in PV$. What is $\tau^*(P)$?

$\tau(P)$.

Inductive Step

Now we assume that $\mathbf{P}, \mathbf{Q} \in \mathcal{F}_{\mathcal{PV}}$ and that $\tau^*(\mathbf{P})$ and $\tau^*(\mathbf{Q})$ return a value from $\{\mathbf{true}, \mathbf{false}\}$. Then:

$$\tau^*(\neg Q) = \begin{cases} \mathbf{true}, & \textit{if } \tau^*(Q) \text{ is false.} \\ \mathbf{false}, & \textit{otherwise} \end{cases}$$

$$\tau^*(Q \wedge P) = \begin{cases} \mathbf{true}, & \textit{if } \tau^*(Q) = \tau^*(P) = \text{true} \\ \mathbf{false}, & \textit{otherwise} \end{cases}$$

$$\tau^*(Q \vee P) = \begin{cases} \mathbf{false}, & \textit{if } \tau^*(Q) = \tau^*(P) = \text{false.} \\ \mathbf{true}, & \textit{otherwise} \end{cases}$$

Semantics

- **Satisfies** If $\tau^*(Q) = \mathbf{true}$, then we say that τ^* *satisfies* Q.
- **Falsifies** If $\tau^*(Q) = \mathbf{false}$, then we say that τ^* *falsifies* Q.
- We can determine which *truth assignments* of the propositional variables *satisfy* a particular *propositional statement* using a *truth table*.

Truth Tables

We will use $\{0,1\}$ to represent $\{\mathbf{true}, \mathbf{false}\}$.

p_1	p_2	$\neg p_1$	$\neg p_2$	$p_1 \wedge p_2$	$p_1 \vee p_2$	$p_1 \to p_2$	$p_1 \leftrightarrow p_2$
0	0					1	
0	1					1	
1	0					0	
1	1					1	

Q: What does $p_1 \to p_2$ *really* mean?

means that if p_1 is true it requires p_2 to be true.

Exercise: using Venn diagrams, remind yourself about the meaning of $\to$ by showing when $p_1 \to p_2$ is true and when it is false.

Example: Can we determine which *truth assignments* τ *satisfy* $(x \vee y) \to (\neg x \wedge z)$?

x	y	z	$x \vee y$	$\neg x \wedge z$	$(x \vee y) \to (\neg x \wedge z)$	τ
0	0	0	0	0	1	$\neg x \wedge \neg y \wedge \neg z$
0	0	1	0	1	1	$\neg x \wedge \neg y \wedge z$
0	1	0	1	0	0	
0	1	1	1	1	1	$\neg x \wedge y \wedge z$
1	0	0	1	0	0	
1	0	1	1	0	0	
1	1	0	1	0	0	
1	1	1	1	0	0	

So, $(x \vee y) \rightarrow (\neg x \wedge z)$ is *true* whenever

$(\neg x \wedge \neg y \wedge \neg z)$ or $(\neg x \wedge \neg y \wedge z)$ or $(\neg x \wedge y \wedge z)$

are true.

Therefore,

$$(x \vee y) \rightarrow (\neg x \wedge z) \Leftrightarrow (\neg x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge z)$$

A *formula* that is a *conjuction* or a bunch of $\wedge$s of *propositional variables* or their *negation* is called a

DNF:

A formula is in *Disjunctive Normal Form* if it is the *disjunction* ($\vee$) of *minterms*.

Example:

$$(\neg x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge z)$$

is the *DNF* of

$$(x \vee y) \rightarrow (\neg x \wedge z)$$

Q: What does the *DNF* construction tell us about *all boolean functions*?

Boolean Functions and Circuit Diagrams

Q: How are *DNF* formulas useful?

can create a parse tree which translates to a circuit diagram.

Suppose we have a *boolean function* $f(x, y, z)$, *equivalent* to the *DNF* formula:

$$(\neg x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge \neg z).$$

Then, we can convert $f(x, y, z)$ into a *parse tree*:

How can we create a *circuit diagram* from the *parse tree*?

Replace $\wedge, \vee, \neg$ with their respective gates.

Conjuctive Normal Form

Let's look at the truth table again:

x	y	z	$(x \vee y) \to (\neg x \wedge z)$	τ
0	0	0	1	
0	0	1	1	
0	1	0	0	
0	1	1	1	
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	0	

We used the *truth assignments* that make $(x \vee y) \to (\neg x \wedge z)$ true, to construct an equivalent formula in *DNF*.

Q: Can we use the *truth values* that make $(x \vee y) \to (\neg x \wedge z)$ *false* to also construct an *equivalent formula*?

Notice that $(x \vee y) \to (\neg x \wedge z)$ is *true* when:

¬ line 3 and $\neg(\neg x \wedge y \wedge \neg z) \wedge$

¬ line 5 and $\neg(x \wedge \neg y \wedge \neg z) \wedge$

¬ line 6 and $\neg(x \wedge \neg y \wedge z) \wedge$

¬ line 7 and $\neg(x \wedge y \wedge \neg z) \wedge \neg(x \wedge y \wedge z)$

¬ line 8

Therefore, $(x \vee y) \to (\neg x \wedge z) \Leftrightarrow$

$(x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee \neg y \vee \neg z)$.

So *truth tables* are great, right??

Q: How many *rows* would we need in a truth table if we have k *propositional variables*?

2^k

Goal: A *proof system* or *method to determine* whether a *propositional statement* is always true regardless of the truth assignments.

More Definitions

Tautology We say that a *propositional formula* P is a *tautology* if every truth assignment *satisfies* P.

Satisfiable We say that a *propositional formula* P is *satisfiable* if one truth assignment *satisfies* P.

Unsatisfiable We say that a *propositional formula* P is *unsatisfiable* if no truth assignment *satisfies* P.

Examples

Tautology:

$\neg p \vee p$

Satisfiable:

x

Unsatisfiable:

$\neg p \wedge p$

Logically Implies: P *logically implies* Q ***iff*** $P \to Q$ *is a tautology.*

Q: When is $P \to Q$ a *tautology*? when every truth assignment that satisfies P also satisfies Q.

We can denote "P *logically implies* Q" by $P \models Q$ or $P \Rightarrow Q$.

Q: What is the difference between $P \Rightarrow Q$ and $P \to Q$?

semantics and talks about the meaning

syntax

Logically Equivalent: P *and* Q *are logically equivalent* ***iff*** $P \Rightarrow Q$ *and* $Q \Rightarrow P$.

We denote this $P \Leftrightarrow Q$

Q: How are "$P \Leftrightarrow Q$" and "$P \leftrightarrow Q$" related?

$P \Leftrightarrow Q$ if every truth assignment makes either P and Q or both P and Q false.

Some Logical Equivalences

Law of Double negation: $\neg\neg P \Leftrightarrow P$

De Morgan's Laws: $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$

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

Commutative Laws: $P \wedge Q \Leftrightarrow Q \wedge P$

$P \vee Q \Leftrightarrow Q \vee P$

Associative Laws: $(P \wedge Q) \wedge R \Leftrightarrow P \wedge (Q \wedge R)$

$(P \vee Q) \vee R \Leftrightarrow P \vee (Q \vee R)$

Distributive Laws: $P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$

$P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$

Identity Laws: $P \wedge (Q \vee \neg Q) \Leftrightarrow P$

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

$\rightarrow$ Law: $P \rightarrow Q \Leftrightarrow \neg P \vee Q$

$\leftrightarrow$ Law: $P \Longleftrightarrow Q \Leftrightarrow (\neg P \vee Q) \wedge (\neg Q \vee P)$

$P \rightarrow Q \wedge Q \rightarrow P \Longleftrightarrow P \wedge Q \vee \neg P \wedge \neg Q$

Propositional Logic Review

Idea Want a *formal* way to make inferences from *boolean statements*.

DEFINITIONS:

- **Syntax** The *symbols* that we use to *represent expressions*

  e.g., a programing language: a piece of code *compiles* if it has *proper syntax*.

- **Semantics** The *meaning* of what the *symbols represent.*

  e.g., a programming language: a piece of code meets its *specifications* if the *semantics* are correct.

- **Proposition** a *statement* that is a *sentence* that can be *evaluated* to **true** or **false**.

- **Propositional Variable** a *variable* that stands for or represents a *primitive proposition*, i.e., the *simplest propositions* we are considering.

  We denote the *set* of *propositional variables* as PV

- **Connectives** The *symbols*, $\{\vee, \wedge, \rightarrow, \leftrightarrow, \neg\}$, that we use to join *propositions* together to make *new propositions*.

Proving Two Formulas are Logically Equivalent

Example 1

$$(x \to y) \land (x \to z) \Leftrightarrow x \to (y \land z)$$

Proof.

$$\begin{aligned}(x \to y) \land (x \to z) &\Leftrightarrow \\ &\Leftrightarrow \\ &\Leftrightarrow \quad x \to (y \land z) \quad \to \text{ law}\end{aligned}$$

Example 2

$$(Q \to P) \land (\neg Q \to P) \Leftrightarrow P$$

Proof:

$$\begin{aligned}(Q \to P) \land (\neg Q \to P) &\Leftrightarrow (\neg Q \lor P) \land (Q \lor P) \quad (\to \times 2)\ (\neg\neg) \\ &\Leftrightarrow (P \lor \neg Q) \land (P \lor Q) \quad (\text{comm} \times 2) \\ &\Leftrightarrow P \lor (\neg Q \land Q) \quad (\text{dist.}) \\ &\Leftrightarrow P \quad (\text{Id}) \\ &\Leftrightarrow \\ &\Leftrightarrow\end{aligned}$$

Q: What did we just prove?

Proof by cases is a valid proof method.

Proving Two Formulas are *NOT* Equivalent

Q: How do we show that two formula are **not** *equivalent*?

Example 3

$$(y \to x) \land (z \to x) \overset{??}{\Leftrightarrow} (y \land z) \to x$$

Simplify a bit first:

$(\to \ x2)$

$$\begin{aligned}
(y \to x) \land (z \to x) &\Leftrightarrow (\neg y \lor x) \land (\neg z \lor x) \\
&\Leftrightarrow (x \lor \neg y) \land (x \lor \neg z) \ (\text{comm}) \\
&\Leftrightarrow \\
&\Leftrightarrow x \lor (\neg y \land \neg z) \Leftrightarrow (\neg y \land \neg z) \lor x
\end{aligned}$$

$\Leftrightarrow (y \lor z) \to x$ ($\to$ law)

Demorgan

Q: Is there a *truth assignment* that satisfies *only* one of $(y \land z) \to x$ and $(y \lor z) \to x$?

$\tau(x) = \text{FALSE} \quad \tau(y) = \text{TRUE} \quad \tau(z) = \text{FALSE}$

Back to Stuctural Induction...

We have already seen that $p \to q \Leftrightarrow \neg p \vee q$.

Q. Can all *propositional formulas* be rewritten using just $\vee$ and $\neg$?

An Example

Recall the definition of $\mathcal{F}$. $\mathcal{F}$ be the smallest set such that:

Basis: The set of *propositional variables* belong to $\mathcal{F}$, e.g., $P, Q, R, \ldots \in \mathcal{F}$

Induction Step: If P, Q belong to $\mathcal{F}$ then

1. $(P \vee Q) \in \mathcal{F}$
2. $(P \wedge Q) \in \mathcal{F}$
3. $(P \to Q) \in \mathcal{F}$
4. $\neg P \in \mathcal{F}$

CLAIM: Let $\mathcal{F}$ be as defined above. If $R \in \mathcal{F}$ then R can be *constructed* using only 4. and 1. above. I.e.,

> "*$\forall R \in \mathcal{F}$, there exists a logically equivalent formula in $\mathcal{F}$ constructed using only the operators $\neg$ and $\vee$.*"

CLAIM: "*$\forall R \in \mathcal{F}$, there exists a logically equivalent formula in $\mathcal{F}$ constructed using only the operators $\neg$ and $\vee$.*"

Proof. Structural induction on $R \in \mathcal{F}$.

Basis: Just $P \in F$ where P is a prop. var.

Inductive Step: If R is not a propositional variable, then R is constructed from one of the 4 rules.

Case 1. R is $(P \vee Q)$ done bc by structural induction P, Q can be written only with $\neg, \vee$

Case 2. R is $(P \wedge Q)$.

What is $(P \wedge Q)$ logically equivalent to in terms of $\vee$ and $\neg$?

$P \wedge Q \iff \neg(\neg P \vee \neg Q)$ since P, Q by struct. ind. can be written using only $\neg, \vee$ ...

Case 3. R is $(P \rightarrow Q)$.

What is $(P \rightarrow Q)$ logically equivalent to in terms of $\vee$ and $\neg$?

$P \rightarrow Q \iff \neg P \vee Q$

Case 4. R is $\neg P$.

Therefore, by structural induction the claim holds. $\square$

Q. What does this tell you about $\wedge, \vee, \rightarrow$ and $\neg$?

A. We don't actually need all the connectives.

Proving an item does *NOT* belong to a set

Consider the following *set* $\mathcal{H}$ defined by *induction*:

$\mathcal{H}$ is the smallest set such that:

Basis: The set of *propositional variables* belongs to $\mathcal{H}$

Induction Step: if $P \in \mathcal{H}$ and $Q \in \mathcal{H}$ then

1. $P \vee Q \in \mathcal{H}$

2. $P \wedge Q \in \mathcal{H}$

Q: Can all *propositional formulas* belong to $\mathcal{H}$?

Q: How do we prove that an item *does not belong* to an *inductively defined set*?

prove that a property holds for all items in the set and show that the item does not have the property.

Q: Suppose that all *propositional variables* are assigned a *value* of *true*. What does this tell you about *every* item in $\mathcal{H}$?

Every formula in H must be true.

Q: How does this help us?

$\neg P$ cannot be expressed in H.

Consider again $\neg P$. If $P \Leftrightarrow T$, what is the *truth value* of $\neg P$?

Let's prove our claim:

CLAIM 3: *$\forall h \in \mathcal{H}$, if every propositional variable in h has value true, then h is true.*

Proof. By structural induction on $R \in \mathcal{H}$.

Basis: Propositional variables. if P is true then the formula P is true.

Inductive Step: Assume that $P, Q \in \mathcal{H}$ satisify the claim.

Case 1: $R \leftrightarrow P \vee Q$: $T \vee T \Leftrightarrow T$ so R evaluates to T.

Case 2: $R \leftrightarrow P \wedge Q$: $T \wedge T \Leftrightarrow T$ so R evaluates to True.

$\square$

CLAIM 4: $\neg P \notin \mathcal{H}$.

Proof. if $\neg P \in H$ then $\neg P$ must satisfy Claim 3 which says that if $P \Leftrightarrow T$ then $\neg P$ is true which is a contradiction.

$\square$

Q: What does CLAIM 4 tell us about $\wedge$ and $\vee$ with respect to the *set* of all *propositional formulas*?