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, $p \rightarrow q \rightarrow r$ means $(p \rightarrow q) \rightarrow r$ or $p \rightarrow (q \rightarrow r)$?

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

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

   by Structural induction → $= \underbrace{op(e_1) + 1 + op(e_2)}\, + 1$

   $= op(e) + 1$ ← by def'n.

$\square$

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**: $(p \rightarrow q)$
- **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,

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

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.,

   $(3 \times 2) + 5$

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

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 {**true**, **false**}.

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

Q: What does $p_1 \rightarrow p_2$ *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 $\rightarrow$ by showing when $p_1 \rightarrow p_2$ is true and when it is false.

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

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

$$(\neg x \wedge \neg y \wedge \neg z) \text{ or } (\neg x \wedge \neg y \wedge z) \text{ or } (\neg x \wedge y \wedge z)$$

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*?