Groups and Rings

Carter Aitken

2025-05-05

Abstract

Contents

1 Proposition **3**

2 Logical Arguments **3**

3 Propositional Logic **4**

3.1 Notations and Lp 4

Notation: Symbols 4

Definition: Well Formed Expressions (WFE) 4

Notation: Operator Priority 5

4 Translating English into Prop Logic **6**

4.1 Examples 6

Lemma: Balanced Paranthesis 6

5 Semantics **8**

5.1 Semantics of Lp formulas 8

Definition: Truth Evaluation 9

Notation: Eval Function 10

Definition: Satisfiable under t 11

Definition: Unsatisfiable 11

Definition: Tautology 12

Notation: Tautology 12

Definition: Satisfiable set of Formulas 12

Definition: Unsatisfiable set of Formulas 12

Example: Infinite Sigma 12

Definition: Arguement . . . 13
Definition: Maximally Satisfiable . . . 13
Example: The Infinite Atomic Set is Maximally Satisfiable . . . 13
Definition: Uniquely Satisfiable . . . 13
Theorem: Uniquely Satisfiable iff Maximally Satisfiable . . . 13
Definition: Models . . . 14
Definition: Logical Equivalence . . . 14
Example: Important Logical Equivalences . . . 16
Theorem: Replaceabiliy . . . 17
Example: Pierce's Law . . . 17
5.2 Normal Forms . . . 17
Definition: Literals . . . 17
Definition: Conjuctive Clause . . . 17
Definition: Disjunction Clause . . . 17
Definition: Conjunctive Normal Form (CNF) . . . 17
Definition: Disjunctive Normal Form (DNF) . . . 17
Example: into DNF . . . 18
Example: into DNF . . . 18
5.3 Essential Laws of Propositional Calculus . . . 18
Definition: Boolean Algebra . . . 19
5.4 Amount of Connectives per Formula . . . 20
Theorem: Formula to CNF . . . 20
Definition: Adequate Set of Connectives . . . 21
Theorem: Std Connectives are Adequate . . . 21
Theorem: And and Neg are Ade . . . 21
Theorem: NOR is adequate . . . 22
Theorem: NAND is ade . . . 22
Theorem: And isn't Ade . . . 22

1 Proposition

An atomic prop cannot be broek down into smaller propositions.

A compound proposition is composed of atomics props.

Atomic

- I am graduating.
- I am applying for grad school.

Compound

- I am not graduating
- I am graduating implies im applying for grad school

2 Logical Arguments

An **argument** is a set of props, consiting of zero or more premises.

Premises: If I am applying for grad schools, then I must be graduating. I am graduating.

Conclusion: I am applying for grad school.

If the concl doesn't follow from prem then the argument is invalid.

3 Propositional Logic

3.1 Notations and Lp

Notation 3.1: Symbols.

- ***Proposition Symbols.*** *Used for atomic formulas. We'll use lowercase letters,* $\{a, b, c, \ldots\}$.
- ***Connections.*** $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$.
- ***Parems.*** *Denotes order.*

Let Lp be the language of propositional logic.

$$\wedge \wedge \vee \leftarrow \text{(not legal)}$$

$$(p \leftarrow \text{(not legal)}$$

We defined a tokenizer. We'll now define the ***parser****.*

Definition 3.1: Well Formed Expressions (WFE).

1. *a propositional symbol is a well formed expression.*

$$p \leftarrow \text{(WFE)}$$

2. *If* $A \in \text{Form}(\text{Lp}) \implies (\neg A) \in \text{Form}(\text{Lp})$.
3. *If* $A, B \in \text{Form}(\mathcal{L}^{\text{p}}) \implies (A \wedge B) \in \text{Form}(\mathcal{L}^{\text{p}})$.
4. *If* $A, B \in \text{Form}(\mathcal{L}^{\text{p}}) \implies (A \vee B) \in \text{Form}(\mathcal{L}^{\text{p}})$.
5. *If* $A, B \in \text{Form}(\mathcal{L}^{\text{p}}) \implies (A \rightarrow B) \in \text{Form}(\mathcal{L}^{\text{p}})$.
6. *If* $A, B \in \text{Form}(\mathcal{L}^{\text{p}}) \implies (A \leftrightarrow B) \in \text{Form}(\mathcal{L}^{\text{p}})$.

```
data WFE t = PropSym t |
             ExprBin (t -> t -> t) (WFE t) (WFE t) |
             ExprUn (t -> t) (WFE t)

eval :: WFE t -> t
eval wfe = case wfe of
  PropSym t -> t
```

```
  ExprBim fn l r -> fn (eval l) (eval r)
  ExprUn fn u -> fn (eval u)


neg :: Bool -> Bool
neg pred = if pred then False else True

conj :: Bool -> Bool -> Bool
conj a b = if a then b else False

inj :: Bool -> Bool -> Bool
inj a b = (neg a) `conj` (neg b)
     -- = (conj `on` neg) a b

xor :: Bool -> Bool -> Bool
xor a b = (a `ing` b) `conj` ((neg . conj) a b)
```

To make inline Lp work, we need to establish operator prior and associativity.

Notation 3.2: Operator Priority. *Operator prior is as follows:*

1. $\neg$
2. $\wedge \leftarrow$ (left assoc)
3. $\vee \leftarrow$ (left assoc)
4. $\implies \leftarrow$ (right assoc)
5. $\iff \leftarrow$ (left assoc)

$$((\neg p) \vee q) = \neg p \vee q$$

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

$$p \to q \to r = p \to (q \to r)$$

$$p \wedge q \wedge r = (p \wedge q) \wedge r$$

4 Translating English into Prop Logic

4.1 Examples

$$\begin{aligned}
s &:= \text{I am applying to grad schools} \\
j &:= \text{I am applying to jobs} \\
g &:= \text{I am graduating} \\
\text{s or j} &= s \vee j \\
\text{i am either S or J but not S and J} &= (s \vee j) \wedge \neg(s \wedge j) \\
&= s \iff \neg j \\
&= (s \implies \neg j) \wedge (\neg j \implies s) \\
(a \vee b) \wedge \neg(a \wedge b) &:= a \oplus b \\
\text{s if g} &= g \implies s \\
\text{s only if g} &= s \implies g \\
\text{storm} \implies \text{rain} &= \text{rain if storm} \\
&= \text{it's raining if it's storming} \\
&= \text{storm only if rain} \\
&= \text{it's storming only if it's raining} \\
\text{g is sufficient for s} &= g \implies s \\
\text{g is necessary for s} &= s \implies g \\
\text{Although g, i am not j} &= g \wedge \neg j \\
\oplus = \neg \circ \leftrightarrow &
\end{aligned}$$

Lemma 4.1: Balanced Paranthesis. *Every formula in* $\mathrm{Form}(\mathcal{L}^{\mathrm{p}})$ *has balanced paranths.*

Proof. Let A be an arbitrary formula in $\mathrm{Form}(\mathcal{L}^{\mathrm{p}})$. The following proof is by **structural induction**. Let $R(A)$ be the property that $LP(A) = RP(A)$. Letting $LP(A)$ be the number of Left paranthesis' in A. Let $RP(A)$ be the number of Right paranthesis' in A.
Base Case: A is atomic, $A = p$ for some prop p.

$$LP(A) = RP(A) = 0$$

Inductive Case 1: $A = \neg B$ for some $B \in \mathrm{Form}(\mathcal{L}^{\mathrm{p}})$. Our IH says $LP(B) = RP(B)$.

$$LP(A) = LP((\neg B)) = 1 + LP(B) = 1 + RP(B) = RP((\neg B)) = RP(A)$$

Inductive Case 2: Let $(\diamond)$ be a generic binary operator $(\diamond) : (\mathcal{L}^{\mathrm{p}}) \times (\mathcal{L}^{\mathrm{p}}) \to (\mathcal{L}^{\mathrm{p}})$. $A = (B \diamond C)$, for some $B, C \in \mathrm{Form}(\mathcal{L}^{\mathrm{p}})$, with $LP(B) = RP(B)$ and $LP(C) = RP(C)$ by IH.

$$LP(A) = LP((B \diamond C)) = 1 + LP(B) + LP(C)$$

$$= 1 + RP(B) + RP(C) = RP((B \diamond C)) = RP(A)$$

So by the principal of structural induction, $R(A)$ holds. $\square$

thm: for any $A \in \mathrm{Form}(LP)$, $LP(A) = RP(A)$, proven above

```
# Machine Proof of Above in Roc
Inductive formula : Type :=
  | Atom : string  -> formula
  | Not  : formula -> formula
  | And  : formula -> formula -> formula
  | Or   : formula -> formula -> formula
  | Imp  : formula -> formula -> formula
  | Iff  : formula -> formula -> formula

Fixpoint lparans (f : formula) : nat :=
  mathc f with
  | Atom _ => 0
  | Not f1 => lparans f1 + 1
  | And f1 f2 => lparans f1 + lparns f2 + 1
  | And f1 f2 => lparans f1 + lparns f2 + 1
  | And f1 f2 => lparans f1 + lparns f2 + 1
```

```
  | And f1 f2 => lparans f1 + lparns f2 + 1



Fixpoint rparans (f : formula) : nat :=
  mathc f with
  | Atom _ => 0
  | Not f1 => rparans f1 + 1
  | And f1 f2 => rparans f1 + lparns f2 + 1
  | And f1 f2 => rparans f1 + lparns f2 + 1
  | And f1 f2 => rparans f1 + lparns f2 + 1
  | And f1 f2 => rparans f1 + lparns f2 + 1

Theorem lparans_eq_rparens : forall f : formula, lparens f = rparens.
Proof.
  induction f.
  - (* Atom *) simpl. reflexivity.
  - (* Not *) simpl. rewrite. IHf. reflexivity.
  - (* And *) simpl. rewrite. IHf1. IHf2. reflexivity.
  - (* And *) simpl. rewrite. IHf1. IHf2. reflexivity.
  - (* And *) simpl. rewrite. IHf1. IHf2. reflexivity.
  - (* And *) simpl. rewrite. IHf1. IHf2. reflexivity.
Qed.

Theorem lparens_eq_rparens' : forall f : formula, lparens f = rparens f.
Proof.
  Induction f; reflexivity.
Qed.
```

5 Semantics

5.1 Semantics of Lp formulas

What does p mean?

$$\begin{bmatrix} p & q \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} p & \neg p \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p & q & p \wedge q \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} p & q & \neg p & \neg p \wedge q \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$(\neg) : \{0,1\} \to \{0,1\}$$

$$(\diamond) : \{0,1\} \times \{0,1\} \to \{0,1\}$$

Definition 5.1: Truth Evaluation. *A* ***truth evaluation*** *is a mapping from proposition symbols to truth values.*

$$t : \mathrm{Atom}(\mathcal{L}^{\mathrm{p}}) \to \{0,1\}$$

Definition 5.2. *Evaluation of formula* $A \in \mathrm{Form}(\mathcal{L}^{\mathrm{p}})$ *under a truth evaluation* t.

Notation 5.1: Eval Function. A^t

Case 1: $A = p$, $p \in \mathrm{Atom}(\mathcal{L}^{\mathrm{p}})$. *Then* $A^t = p^t = t(p)$.
Case 2: $A = \neg B$. *Then* $A^t = (\neg B)^t$. *Note that* $\neg(B^t)$ *is wrong, because* $\neg$ *is from syntax, and* 0 *is from semantics. So*

$$(\neg B)^t = \begin{cases} 0 & : B^t = 1 \\ 1 & : B^t = 0 \end{cases}$$

Case 3: $A = B \wedge C$.

$$A^t = (B \wedge C)^t = \begin{cases} 1 & : B^t = 1 \text{ and } C^t = 1 \\ 0 & : \text{otherwise} \end{cases}$$

Case 4: $A = B \vee C$.

$$A^t = (B \vee C)^t = \begin{cases} 1 & : B^t = 1 \text{ or } C^t = 1 \\ 0 & : \text{otherwise} \end{cases}$$

Case 5: $A = B \to C$.

$$A^t = (B \to C)^t = \begin{cases} 1 & : B^t = 0 \text{ or } C^t = 1 \\ 0 & : \text{otherwise} \end{cases}$$

$$\begin{bmatrix} p & q & p \to q \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Case 6: $A = B \leftrightarrow C$.

$$A^t = (B \leftrightarrow C)^t = \begin{cases} 1 & : B^t = C^t \\ 0 & : \text{otherwise} \end{cases}$$

Theorem 5.1. *For all* $A \in \mathrm{Form}(\mathcal{L}^{\mathrm{p}}_{\neg,\vee,\wedge})$ *and* $\forall t$, $\Delta(A)^t = (\neg A)^t$ *where*

$$\Delta(A) := \begin{cases} \neg p & : \textit{if } A = p \textit{ for some } p \in \mathrm{Atom}(\mathcal{L}^{\mathrm{p}}_{\neg\vee\wedge}) \\ \neg\Delta(B) & : A = \neg B,\ B \in \mathrm{Form}(\mathcal{L}^{\mathrm{p}}_{\neg\vee\wedge}) \\ \Delta(B) \vee \Delta(C) & : A = B \wedge C \\ \Delta(B) \wedge \Delta(C) & : A = B \vee C \end{cases}$$

$$\Delta : \mathrm{Form}(\mathcal{L}^{\mathrm{p}}_{\neg\vee\wedge}) \to \mathrm{Form}(\mathcal{L}^{\mathrm{p}}_{\neg\vee\wedge})$$

Example.

$$\Delta(\neg p \wedge q) = \neg\neg p \vee \neg q$$

Proof. Let $R(A)$ be the property that $\Delta(A)^t = (\neg A)^t$.

Case 1: $A = p$.

$$\Delta(A)^t = \Delta(p)^t = (\neg p)^t$$

$$(\neg A)^t = (\neg p)^t \implies R(A) \text{ holds}$$

Case 2: $A = \neg B$.

$$\Delta(A)^t = \Delta(\neg B)^t = (\neg\Delta(B))^t$$

$$\text{IH: } \Delta(B)^t = (\neg B)^t$$

$$= \begin{cases} 1 & : \Delta(B)^t = 0 \\ 0 & : \Delta(B)^t = 1 \end{cases}$$

$$= \begin{cases} 1 & : (\neg B)^t = 0 \\ 0 & : (\neg B)^t = 1 \end{cases}$$

So by IH, case 2 holds. □

Recap: $t : \mathrm{Atom}(\mathcal{L}^{\mathrm{p}}) \to \{0, 1\}$.
$A^t :=$ truth value of A under thrugh valuation t

Definition 5.3: Satisfiable under t. *A formula A is* ***satisfiable*** *if there exists t such that $A^t = 1$.*

Definition 5.4: Unsatisfiable. *a formula A is unstatisfiable if for all t, $A^t = 0$.*

Example 5.1.

$$p \leftarrow \text{(satis)}$$

$$p \wedge \neg p \leftarrow \text{(unsatis)}$$

$$p \vee \neg p \leftarrow \text{(satis)}$$

$$\textit{in some sense, } p \vee \neg p = 1$$

Definition 5.5: Tautology. *A formula is a **tautology** if* $\forall t,\ A^t = 1$.

Example 5.2. $p \vee \neg p$

Notation 5.2. *a unsatisfiable formula under* t *is called a **contradiction**.*

Notation 5.3: Tautology. *If* A *is a tautology, then* $A \models\!\mid \tau$.

Definition 5.6: Satisfiable set of Formulas. *a set* $\Sigma :\subseteq (\mathcal{L}^{\mathrm{p}})$ *is called **satisfiable** if* $\exists t\ s/t\ \forall A \in \Sigma,\ A^t = 1$.

A truth evaluation is basically defining the variable to true or false, then evaluating the formula.

Example 5.3. *Let* $\mathbb{S}$ *be the satifiable adjective.*

$$\{p\} \leftarrow \mathbb{S}$$

$$\{p, \neg p\} \leftarrow \neg\mathbb{S}$$

Definition 5.7: Unsatisfiable set of Formulas. *A set* $\Sigma \leftarrow \neg\mathbb{S}$ *when* $\forall t$,

$$\exists A \in \Sigma,\ A^t = 0$$

Example 5.4. $\emptyset$*? It's satisfiable.*

Example 5.5: Infinite Sigma. $\Sigma := \{p | p \in \mathrm{Atom}(\mathcal{L}^{\mathrm{p}})\}$. *Define* $t : t(p_i) = 1$.

Definition 5.8: Arguement. *Consists of a set of premises and a conclusion. The arguement is valid when the conclusion follows from the premises.*
A formula A is a tautological consequence of $\Sigma \subseteq \mathrm{Form}(\mathcal{L}^{\mathrm{p}})$ if $\forall t,\ \Sigma^t = 1 \implies A^t = 1$.

$$\Sigma^t := \begin{cases} 1 & \forall A \in \Sigma,\ A^t = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\Sigma^t = \bigwedge_{A \in \Sigma} A^t$$

This is saying $\Sigma \implies A$

Example 5.6. $\Sigma = \{p \to q, p\}$ $A = q$. *So* $\Sigma \implies A$. *Hypothetical syllagism.*

Notation 5.4. $\Sigma \models A$, *means A is a logical consequence of Σ.*

Example 5.7. $\{p\} \not\models \neg p$

Definition 5.9: Maximally Satisfiable. *a set $\Sigma \subseteq \mathrm{Form}(\mathcal{L}^{\mathrm{p}})$ is maximally satisfiable if $\forall A \in \mathrm{Form}(\mathcal{L}^{\mathrm{p}})$, $\Sigma \models A$ xor $\Sigma \models \neg A$.*
Note that $\Sigma \models \neg A \not\equiv \Sigma \not\models A$.

Example 5.8: The Infinite Atomic Set is Maximally Satisfiable. $\{p\} \models p$, $\{p\} \not\models \neg p$, $\{p\} \not\models q$, $\{p\} \not\models \neg q$. *So $\{p\}$ isn't maximally satisfiable.*

$$\{p_1, p_2, \dots\} := \Sigma$$

$$t(p_i) := 1 \implies \Sigma^t = 1$$

$$\implies \Sigma \models A \text{ xor } \Sigma \models \neg A$$

Definition 5.10: Uniquely Satisfiable. *Σ is uniquely satisfiable if*

$$\exists! t \ s/t \ \Sigma^t = 1$$

Theorem 5.2: Uniquely Satisfiable iff Maximally Satisfiable. *Suppose Σ is satisfiable. Then Σ is uniquely satisfiable **iff** Σ is maximally satisfiable.*

Proof. ($\implies$). Assume Σ maximimally satisfiable. Assume by contradiction that there is t_1, t_2 s/t $\Sigma^{t_1} = 1$ and $\Sigma^{t_2} = 1$, so Σ isn't uniquely satisfiable.
So $t_1 \neq t_2 \implies \exists p$ s/t $t_1(p) \neq t_2$.
$t_1(p) = 1 \iff t_2(p) = 0$. Let $t_1(p) = 1$ and $t_2(p) = 0$.
We know $\Sigma \models p \operatorname{xor} \Sigma \models \neg p$.
Case 1: $\Sigma \models p$. $p^{t_1} = 1$ and $p^{t_2} = 1$ which is a contradiction.
Case 2: $\Sigma \models \neg p$. $(\neg p)^{t_1} = 1$ and $(\neg p)^{t_2} = 1$ which is a contradiction. $\square$

Example 5.9. *If* $\Sigma \leftarrow \neg\mathbb{S} \implies \Sigma \models A$ *holds when?* $\forall t$, $\Sigma^t = 0$. *So* $\Sigma = 1$ *is always false. So that always implies* $A^t = 1$. *So it always holds. YAY.*

Definition 5.11: Models. $\Sigma \models A$ ***iff*** $\Sigma^1 = 1 \implies A^t = 1$.

Example 5.10. $A \leftarrow \neg\mathbb{S} \implies \Sigma \implies A$ *only sometimes*

Example 5.11. $A \leftarrow \tau \implies \Sigma \models A$ *always.*

Example 5.12. $\Sigma \models A \implies \Sigma \cup \{\neg A\}$ *is never satis.*

$$\Sigma^t = 1 \implies A^t = 1$$

$$(\Sigma \cup \{\neg A\})^t = 1?$$

$$\Sigma^t = 1 \implies \neg A = 0$$

$$\Sigma^t = 0 \implies \textit{already not satis}$$

Example 5.13. $\Sigma \leftarrow \mathbb{S}$, $\Sigma' \subseteq \Sigma \implies \Sigma' \leftarrow \mathbb{S}$ *If that* t *satis sigma, then it also satis sigma'.*

Example 5.14. $\Sigma \leftarrow \mathbb{S}$, $\Sigma' \supseteq \Sigma \implies \Sigma' \leftarrow$ *sometimes* $\mathbb{S}$

Definition 5.12: Logical Equivalence. *A is logically equivalent with B if*

$$A \models B \textit{ and } B \models A$$

$$A \models\!\!\dashv B$$

Logical Equivalence is a Equivalence Relation Proof. Lol nevermind. □

Example 5.15: Important Logical Equivalences.

- ***Conj is Abelian:*** $A \wedge B \models\!\!\!\!\!= B \wedge A$
- ***Disj is Abelian:*** $A \vee B \models\!\!\!\!\!= B \vee A$
- ***Equi is Abelian:*** $A \leftrightarrow B \models\!\!\!\!\!= B \leftrightarrow A$
- $\diamond$ ***is Assoc:*** $A \diamond (B \diamond C) \models\!\!\!\!\!= (A \diamond B) \diamond C$ *for* $(\diamond) \in \{\wedge, \vee, \leftrightarrow\}$
- ***Dist of Disj:*** $(A \wedge B) \vee C \models\!\!\!\!\!= (A \vee C) \wedge (B \vee C)$
- ***Dist of Conj:*** $(A \wedge B) \wedge C \models\!\!\!\!\!= (A \wedge C) \vee (B \wedge C)$
- ***De Morgan's Disj:*** $\neg(A \vee B) \models\!\!\!\!\!= \neg A \wedge \neg B$
- ***De Morgan's Conj:*** $\neg(A \wedge B) \models\!\!\!\!\!= \neg A \vee \neg B$
- $\Delta(A) \models\!\!\!\!\!= \neg A$
- $A \models\!\!\!\!\!= \neg\neg A$
- $A \vee \neg A \models\!\!\!\!\!= \tau$
- $A \wedge \neg A \models\!\!\!\!\!= \mathcal{C}$.
- ***Iden of Disj:*** $A \vee \mathcal{C} \models\!\!\!\!\!= A$
- ***Iden of Conj:*** $A \wedge \tau \models\!\!\!\!\!= A$
- ***Domination of Conj:*** $A \wedge \mathcal{C} \models\!\!\!\!\!= \mathcal{C}$
- ***Domination of Disj*** $A \vee \tau \models\!\!\!\!\!= \tau$
- $A \to B \models\!\!\!\!\!= \neg A \vee B$
- ***Contrapositive Equiv:*** $A \to B \models\!\!\!\!\!= \neg B \to \neg A$
- $A \leftrightarrow B \models\!\!\!\!\!= (A \to B) \wedge (B \to A)$
- ***Absorption of Disj:*** $(A \wedge B) \vee A \models\!\!\!\!\!= A$
- ***Absorption of Conj:*** $(A \vee B) \wedge A \models\!\!\!\!\!= A$
- $(A \wedge B) \vee (A \wedge \neg B) \models\!\!\!\!\!= A$
- $(A \vee B) \wedge (A \vee \neg B) \models\!\!\!\!\!= A$

Theorem 5.3: Replaceabiliy. *If $B \models\!\mid C$, then replacing some occurances of B inside A gives a logically equivalent formula. We can do beta reduction.*

Example 5.16: Pierce's Law. $((A \to B) \to A) \to A$

$$
\begin{aligned}
&\models\!\mid ((\neg A \vee B) \to A) \to A \\
&\models\!\mid (\neg(\neg A \vee B) \vee A) \to A \\
&\models\!\mid ((A \wedge \neg B) \vee A) \to A \\
&\models\!\mid A \to A \\
&\models\!\mid \neg A \vee A \\
&\models\!\mid A \vee \neg A \\
&\models\!\mid 1_\tau
\end{aligned}
$$

5.2 Normal Forms

Definition 5.13: Literals. *a formula is a **literal** if it is a either p or $\neg p$ for some $p \in \text{Form}(\mathcal{L}^{\text{p}})$.*

Definition 5.14: Conjuctive Clause. *a formula is a **conjuective clause** is a conjunction of literals.*

Definition 5.15: Disjunction Clause. *a formula is a **disjunction clause** is a disjunction of literals.*

Example 5.17. *p is a disjunctive clause with 1 disjoint.*

Definition 5.16: Conjunctive Normal Form (CNF). *A formula is a (CNF) if it is a conjunction of disj clauses.*

Definition 5.17: Disjunctive Normal Form (DNF). *A formula is a (DNF) if it is a disjunction of conj clauses.*

Example 5.18. $(p \wedge \neg q) \vee \neg p$, $\neg p \vee p \vee q$

$\neg p$ is a literal, a disj clause, a conj clause, in CNF, in DNF.

Example 5.19: into DNF. *Transform $p \leftrightarrow \neg q$ into (DNF).*

1. *Use a truth table.* $(p \wedge \neg q) \vee (\neg p \wedge q)$. $\begin{bmatrix} p & q & p \leftrightarrow \neg q \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$.

Transform $p \leftrightarrow \neg q$ into (CNF).

1. *Use a truth table of* $\neg(p \leftrightarrow \neg p)$. $\models\!\dashv (p \wedge q) \vee (\neg p \wedge \neg q)$. $\xrightarrow{\neg} (\neg p \vee \neg q) \wedge (p \vee q)$.

Example 5.20: into DNF. *Transform $p \leftrightarrow \neg q$ into (DNF).*

1. *Use a truth table.* $(p \wedge \neg q) \vee (\neg p \wedge q)$. $\begin{bmatrix} p & q & p \leftrightarrow \neg q \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$.
2. $p \iff \neg q \models\!\dashv (p \implies \neg q) \wedge (\neg q \implies p) \models\!\dashv (\neg p \vee q) \wedge (q \vee p)$ *is CNF*$\models\!\dashv$

5.3 Essential Laws of Propositional Calculus

We can do algebriac simplification over these formulas. I.e. boolean algebra.

Definition 5.18: Boolean Algebra.

1. *underlying set* B
2. *binary operations* $\{(+), (\cdot)\}$
3. *unary operator* $\{\overline{\cdot}\}$ *"complement"*
4. *nullary operator* $1 : B$ $0 : B$*. The functions that take no input and give 1*
5. *Laws:*
   (a) $x + 0 = x$
   (b) $x \cdot 1 = x$
   (c) $x + \overline{x} = 1$
   (d) $x \cdot \overline{x} = 0$
   (e) $(x + y) + z = x + (y + z)$
   (f) $(x * y) * z = x * (y * z)$
   (g) $x + y = y + x$
   (h) $x * y = y * x$

Example 5.21. *Any set* S *generates a boolean algebra.*

1. $B = \mathcal{P}(S)$
2. $+ = \cup$
3. $\cdot = \cap$
4. $\overline{A} = S \backslash A$
5. $0 = \emptyset$
6. $1 = S$
7. *Laws:*
   (a) $A \cup \emptyset = A$
   (b) $A \cap S = A$
   (c) $A \cup (S \backslash A) = S$
   (d) $A \cap (S \backslash A) = \emptyset$

Example 5.22. $\mathrm{Form}(\mathcal{L}^{\mathrm{p}})$ *generate a boolean algebra.* $B := \mathrm{Form}(\mathcal{L}^{\mathrm{p}})$

1. $+ = \vee$
2. $\cdot = \vee$
3. $\overline{A} = \neg A$
4. $0 =$ *any contradiction*
5. $1 =$ *any tautology*
6. *Laws:*
   (a) $A \vee 0 \models\!\!\!\dashv A$
   (b) $A \wedge 1 \models\!\!\!\dashv A$
   (c) $A \vee \neg A \models\!\!\!\dashv 1$
   (d) $A \wedge \neg A \models\!\!\!\dashv 0$
   (e) ...

Example 5.23. *Disjunctive Normal Form.*

$$xyz + x\overline{y}z + xy\overline{z}$$

$$\text{2 OR GATES, 2 NOT GATES, 6 AND GATES}$$

$$\models\!\!\!\dashv x(yz + \overline{y}z + y\overline{z})$$

$$\models\!\!\!\dashv x(yz + yz + \overline{y}z + y\overline{z}) \text{ since } b + b = b$$

$$\models\!\!\!\dashv x((y + \overline{y})z + y(z + \overline{z}))$$

$$\models\!\!\!\dashv x(y + z)$$

5.4 Amount of Connectives per Formula

Theorem 5.4: Formula to CNF. *Any Formula in* $(\mathcal{L}^{\mathrm{p}})$ *is logically eq to atleast one forumla in CNF (also DNF).*

Note 5.1. *We remove* $\implies$ *and* $\iff$ *using laws.*
Push $\neg$ *inside* $\wedge/\vee$ *using de Morgan's laws*
So we only need $\{\neg, \wedge, \vee\}$

How many connectives are there? A connective is a n-ary boolean function.

1. $n = 1 : 4$ connectives
2. $n = 2 : 16$ connectives

We have 2^n possible inputs, considering a binary string as input. For each input, there are 2 choices. So we have 2^{2^n} choices.

Definition 5.19. *A set of logical connectives* S *is adequate if the connectives in* S *are capable of expressing any set of n-ary connectives.*

Definition 5.20: Adequate Set of Connectives. S *connectives is adequate if for any n-ary boolean function* $f : \{0,1\}^n \to \{0,1\}$ *and for any set of prop symbols* $p_1, p_2, \ldots, p_n$ *there is a* $A \in \text{From}(\mathcal{L}^{\text{p}})$ *using only connectives in* S *and prop symbols of* $p_1, \ldots, p_n$, *s/t*

$$f(p_1, \ldots, p_n) \models\!\!\!\!\!= A$$

Theorem 5.5: Std Connectives are Adequate. $\{\neg, \vee, \wedge\}$ *is ade.*

Theorem 5.6: And and Neg are Ade. $\{\neg, \wedge\}$ *is ade.*

Proof. We already know that

$$\{\neg, \vee, \wedge\} \leftarrow \text{(ade)}$$

$$\text{Show } \forall c \in \{\neg, \wedge, \vee\} \text{ can be expressed in } \{\neg, \wedge\}$$

$$\neg p \models\!\!\!\!\!= \neg p$$

$$p \wedge q \models\!\!\!\!\!= p \wedge q$$

$$p \vee q \models\!\!\!\!\!= \neg(\neg p \wedge \neg q)$$

$\square$

Theorem 5.7.

$$\{\neg, \vee\}$$

$$\{\neg, \implies\}$$

Theorem 5.8: NOR is adequate. *The non-disjunction, NOR is adequate alone.*

$$\begin{bmatrix} p & q & \mathrm{NOR}(p,q) \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Proof. We know $\{\neg, \vee\}$ is ade.

$$\neg p \models\models \mathrm{NOR}(p,p)$$

$$p \vee q \models\models \neg\neg(p \vee q) \models\models \neg \mathrm{NOR}(p,q) \models\models \mathrm{NOR}(\mathrm{NOR}(p,q), \mathrm{NOR}(p,q))$$

□

Theorem 5.9: NAND is ade. $\{\mathrm{NAND}\}$ *is ade.*

Proof. EX. □

Theorem 5.10: And isn't Ade. $\{\wedge\}$ *isn't ade.*

Proof. Assume $\{\wedge\}$ is ade, for the sake of contradiction. So that means every formula can be written in terms of only $\wedge$. So $\neg p \mathrel{\models\!\!\!\dashv} p_1 \wedge p_2 \wedge p_3 \wedge \cdots \mathrel{\models\!\!\!\dashv} A$, where $p_i = f_i(p)$ for some f_i. Build out of p.
Let t be a truth evaluation s/t $(\neg p)^t = 1$.

$$\implies p^t = 0$$

$$\text{We know } \neg p \mathrel{\models\!\!\!\dashv} A$$

$$\implies A^t \mathrel{\models\!\!\!\dashv} 1$$

$$\text{claim } A^t = 0$$

This is a contradiction.
Claim: For any $A \in \mathrm{Form}(\mathcal{L}^{\mathrm{p}}_{\wedge})$, if $p^t = 0$ then $A^t = 0$.

Proof. Induction on A. **Base Case:** $A = p$.

$$A^t = p^t = 0$$

Inductive Case: $A = B \wedge C$, for some $B, C \in \mathrm{Form}(\mathcal{L}^{\mathrm{p}}_{\wedge})$

$$\text{IH: } B^t = C^t = 0$$

$$A^t = (B \wedge C)^t = 0$$

So by induction, $A^t = 0$. So the claim holds, so the proof follows. □

□