Groups and Rings

Carter Aitken

2025-05-05

Abstract

Contents

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* $A, B \in \text{Form}(\text{Lp}) \implies (A \wedge B) \in \text{Form}(\text{Lp})$.
4. *If* $A, B \in \text{Form}(\text{Lp}) \implies (A \vee B) \in \text{Form}(\text{Lp})$.
5. *If* $A, B \in \text{Form}(\text{Lp}) \implies (A \rightarrow B) \in \text{Form}(\text{Lp})$.
6. *If* $A, B \in \text{Form}(\text{Lp}) \implies (A \leftrightarrow B) \in \text{Form}(\text{Lp})$.

```
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* Form(Lp) *has balanced paranths.*

Proof. Let A be an arbitrary formula in Form(Lp). 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 $B \in$ Form(Lp). 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 $(\diamond) : (\text{Lp}) \times (\text{Lp}) \to (\text{Lp})$. $A = (B \diamond C)$, for some $B, C \in$ Form(Lp), 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. □

thm: for any $A \in \text{Form}(LP)$, $LP(A) = RP(A)$, 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.
```

4.2 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 4.1: Truth Evaluation. *A **truth evaluation** is a mapping from proposition symbols to truth values.*

$$t : \mathrm{Atom}(\mathrm{Lp}) \to \{0,1\}$$

Definition 4.2. *Evaluation of formula* $A \in \mathrm{Form(Lp)}$ *under a truth evaluation* t.

Notation 4.1: Eval Function. A^t

Case 1: $A = p$, $p \in \mathrm{Atom(Lp)}$. *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: $A = B \wedge C$.

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

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

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

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

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

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

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

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

Theorem 4.1. *For all* $A \in \mathrm{Form}(\mathrm{Lp}_{\neg,\vee,\wedge})$ *and* $\forall t$, $\Delta(A)^t = (\neg A)^t$ *where*

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

$$\Delta : \mathrm{Form}(\mathrm{Lp}_{\neg\vee\wedge}) \to \mathrm{Form}(\mathrm{Lp}_{\neg\vee\wedge})$$

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. □