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

```
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 $LP(A)$ 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 $LP(B) = RP(B)$.

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

$\square$