COMP30026
Models of Computation

Cezary Kaliszyk and William Umboh

Lecture 2

Propositional Logic

Overview

Last time

- Overview of the subject
- A few problems solvable with logic

Today

- Formal propositional logic
- Soon: Mechanized proof

Propositional = Boolean Logic

Until the mid-19th century, “logic” meant Aristotelian logic.

George Boole took an algebraic view of logic.

Deep connection between logic and arithmetic.

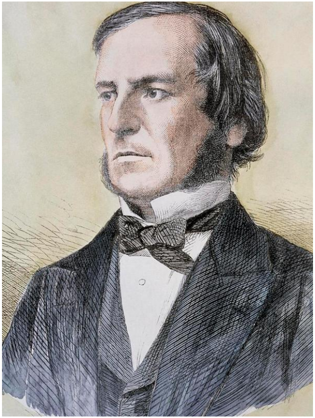


Figure: George Boole, circa 1864

Intro Puzzle

Heidi, Dina and Louise are being questioned by their aunt.

Here is what they say:

> *Heidi: "Dina and Louise had equal share in it; if one is guilty, so is the other."*
> *Dina: "If Heidi is guilty, then so am I."*
> *Louise: "Dina and I are not both guilty."*

Their aunt, knowing that they are honest kids, realises that they cannot tell a lie.

Has she got sufficient information to decide who (if any) are guilty?

Definition

propositional formulas are built from

- atoms $\quad P, \; Q, \; R, \; P_1, \; P_2, \; \ldots$

Definition

propositional formulas are built from

- atoms $P,\ Q,\ R,\ P_1,\ P_2,\ \ldots$
- bottom $\perp$

Definition

propositional formulas are built from

- atoms $P,\ Q,\ R,\ P_1,\ P_2,\ \ldots$
- bottom $\bot$
- top $\top$

Definition

propositional formulas are built from

- atoms $P,\ Q,\ R,\ P_1,\ P_2,\ \ldots$
- bottom $\bot$
- top $\top$
- negation $\neg$ $\neg P$ "not P"

Definition

propositional formulas are built from

atoms	$P,\ Q,\ R,\ P_1,\ P_2,\ \ldots$		
bottom	$\bot$		
top	$\top$		
negation	$\neg$	$\neg P$	"not P"
conjunction	$\wedge$	$P \wedge Q$	"P and Q"

Definition

propositional formulas are built from

atoms	$P,\ Q,\ R,\ P_1,\ P_2,\ \ldots$		
bottom	$\bot$		
top	$\top$		
negation	$\neg$	$\neg P$	"not P"
conjunction	$\land$	$P \land Q$	"P and Q"
disjunction	$\lor$	$P \lor Q$	"P or Q"

Definition

propositional formulas are built from

atoms	$P,\ Q,\ R,\ P_1,\ P_2,\ \ldots$		
bottom	$\bot$		
top	$\top$		
negation	$\neg$	$\neg P$	"not P"
conjunction	$\wedge$	$P \wedge Q$	"P and Q"
disjunction	$\vee$	$P \vee Q$	"P or Q"
implication	$\rightarrow$	$P \rightarrow Q$	"if P then Q"

Definition

propositional formulas are built from

atoms	$P,\ Q,\ R,\ P_1,\ P_2,\ \ldots$		
bottom	$\bot$		
top	$\top$		
negation	$\neg$	$\neg P$	"not P"
conjunction	$\wedge$	$P \wedge Q$	"P and Q"
disjunction	$\vee$	$P \vee Q$	"P or Q"
implication	$\rightarrow$	$P \rightarrow Q$	"if P then Q"

according to following:

$$\varphi \ ::= \ P \mid \bot \mid \top \mid (\neg \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi)$$

Some Well-Formed Formulas

$$P \tag{1}$$

$$(P \rightarrow Q) \tag{2}$$

$$(P \vee \neg P) \tag{3}$$

$$\neg(P \wedge \neg P) \tag{4}$$

$$(P \rightarrow \neg P) \tag{5}$$

$$(((P \rightarrow Q) \rightarrow P) \rightarrow P) \tag{6}$$

Some Well-Formed Formulas

$$P \tag{1}$$

$$(P \rightarrow Q) \tag{2}$$

$$(P \vee \neg P) \tag{3}$$

$$\neg(P \wedge \neg P) \tag{4}$$

$$(P \rightarrow \neg P) \tag{5}$$

$$(((P \rightarrow Q) \rightarrow P) \rightarrow P) \tag{6}$$

Quiz

- Can we express an “if and only if” with what we have?
- $P \longleftrightarrow Q$

Notational Conventions

- Omit outer parentheses

Notational Conventions

- Omit outer parentheses
- Negation binds stronger than $\wedge, \vee$

Notational Conventions

- Omit outer parentheses
- Negation binds stronger than $\land, \lor$
- $\land, \lor$ bind stronger than implication

Notational Conventions

- Omit outer parentheses
- Negation binds stronger than $\land, \lor$
- $\land, \lor$ bind stronger than implication
- $\rightarrow$ is right-associative: $P \rightarrow Q \rightarrow R$ denotes $P \rightarrow (Q \rightarrow R)$

Notational Conventions

- Omit outer parentheses
- Negation binds stronger than $\land, \lor$
- $\land, \lor$ bind stronger than implication
- $\to$ is right-associative: $P \to Q \to R$ denotes $P \to (Q \to R)$
- Warning: $P \land Q \lor R$ is AMBIGUOUS

Examples

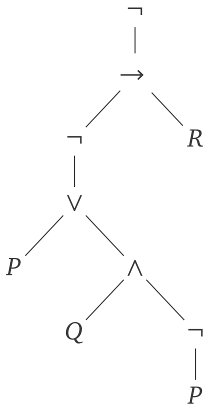
formula $\neg(\neg(P \vee (Q \wedge \neg P)) \to R)$

parse tree

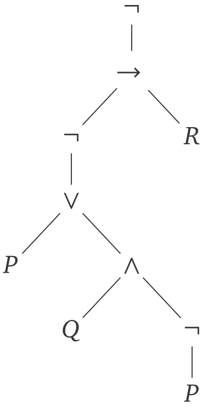
Examples

formula $\neg(\neg(P \vee (Q \wedge \neg P)) \rightarrow R)$

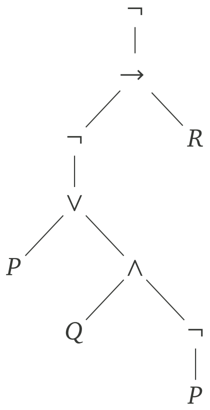
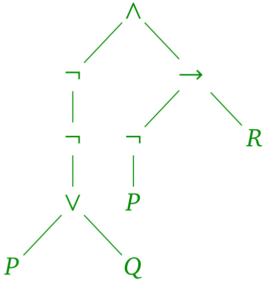
parse tree



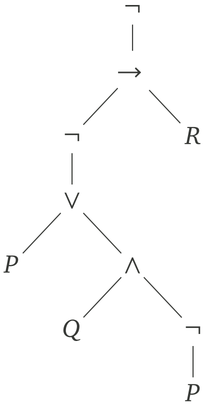
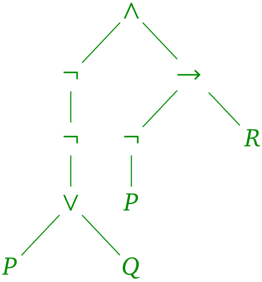
Examples

formula	$\neg(\neg(P \vee (Q \wedge \neg P)) \rightarrow R)$	$\neg\neg(P \vee Q) \wedge (\neg P \rightarrow R)$
parse tree		

Examples

formula	$\neg(\neg(P \vee (Q \wedge \neg P)) \rightarrow R)$	$\neg\neg(P \vee Q) \wedge (\neg P \rightarrow R)$
parse tree		

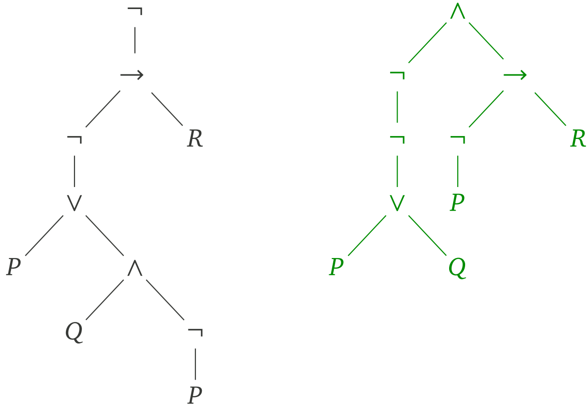
Examples

formula	$\neg(\neg(P \vee (Q \wedge \neg P)) \to R)$	$\neg\neg(P \vee Q) \wedge (\neg P \to R)$
parse tree		

$\neg\neg P \vee Q \wedge \neg P \to R$

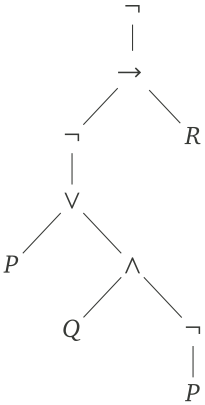
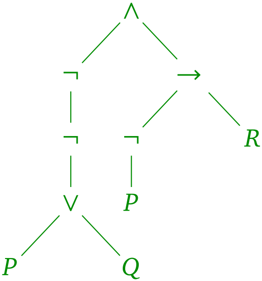
Examples

formula	$\neg(\neg(P \lor (Q \land \neg P)) \to R)$	$\neg\neg(P \lor Q) \land (\neg P \to R)$
parse tree		



$\neg\neg P \lor Q \land \neg P \to R$

Examples

formula	$\neg(\neg(P \vee (Q \wedge \neg P)) \rightarrow R)$	$\neg\neg(P \vee Q) \wedge (\neg P \rightarrow R)$
parse tree		

$\neg\neg P \vee Q \wedge \neg P \rightarrow R$ $\qquad$ $\neg\neg(P \vee Q) \wedge \neg P \rightarrow R$

Boolean Semantics: Connectives

Definition (Truth function)

A function from truth values to truth values.

Boolean Semantics: Connectives

Definition (Truth function)

A function from truth values to truth values.

Boolean truth values: **t** and **f**
(also written **1** and **0**, $\top$ and $\bot$).

Boolean Semantics: Connectives

Definition (Truth function)

A function from truth values to truth values.

Boolean truth values: **t** and **f**

(also written **1** and **0**, $\top$ and $\bot$).

Usually presented as a truth table:

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
0	**0**	**1**	**0**	**0**	**1**	**1**
0	**1**	**1**	**0**	**1**	**1**	**0**
1	**0**	**0**	**0**	**1**	**0**	**0**
1	**1**	**0**	**1**	**1**	**1**	**1**

Boolean Semantics: Letters

Propositional letters are Boolean variables.

Boolean Semantics: Letters

Propositional letters are Boolean variables.

Definition (Truth assignment)

A function from propositional letters to truth values.

Boolean Semantics: Letters

Propositional letters are Boolean variables.

Definition (Truth assignment)

A function from propositional letters to truth values.

Usual notation:

$$v = \{P \mapsto \mathbf{1}, Q \mapsto \mathbf{0}\}.$$

Boolean Semantics: Letters

Propositional letters are Boolean variables.

Definition (Truth assignment)

A function from propositional letters to truth values.

Usual notation:

$$v = \{P \mapsto \mathbf{1}, Q \mapsto \mathbf{0}\}.$$

We then have:

$$\begin{aligned} v(P) &= \mathbf{1} \\ v(Q) = v(R) = \cdots = v(Z) &= \mathbf{0}. \end{aligned}$$

Truth of a Formula

Let $v = \{P \mapsto \mathbf{1}, Q \mapsto \mathbf{0}\}$.

Poll: Which of these formulas are true under v?

1. $P \land Q$
2. $(P \lor Q) \land (P \lor R)$
3. $P \rightarrow Q$
4. $\neg P \rightarrow \neg Q$

Truth of a Formula

Let $v = \{P \mapsto \mathbf{1}, Q \mapsto \mathbf{0}\}$.

Poll: Which of these formulas are true under v?

1. $P \land Q$
2. $(P \lor Q) \land (P \lor R)$
3. $P \rightarrow Q$
4. $\neg P \rightarrow \neg Q$

Shorthand: "$v \vDash \phi$" means "ϕ is true under v".

Truth Tables for Formulas

P	Q	R	$((P$	$\land$	$Q)$	$\lor$	$R)$
0	0	0	0	0	0	**0**	0
0	0	1	0	0	0	**1**	1
0	1	0	0	0	1	**0**	0
0	1	1	0	0	1	**1**	1
1	0	0	1	0	0	**0**	0
1	0	1	1	0	0	**1**	1
1	1	0	1	1	1	**1**	0
1	1	1	1	1	1	**1**	1

Poll

Which of these have the same truth tables?

1. $P \rightarrow Q$ and $\neg Q \rightarrow \neg P$
2. $(P \rightarrow Q) \wedge (P \rightarrow R)$ and $P \rightarrow (Q \wedge R)$
3. $(P \rightarrow R) \wedge (Q \rightarrow R)$ and $(P \wedge Q) \rightarrow R$

Hint: $P \rightarrow Q$ has the same truth table as $\neg P \vee Q$.

Logical Equivalence

Definition

Formulas are *logically equivalent* iff they have equal truth values under **every** truth assignment.

Shorthand: "$F \equiv G$" means "F is logically equivalent to G".

Warning: "→" is weird!

Often read as "implies", but causality is not required!

A	B	$A \rightarrow B$
0	**0**	**1**
0	**1**	**1**
1	**0**	**0**
1	**1**	**1**

1. If there are no bugs, then the program runs correctly.

Warning: "$\rightarrow$" is weird!

Often read as "implies", but causality is not required!

A	B	$A \rightarrow B$
0	**0**	**1**
0	**1**	**1**
1	**0**	**0**
1	**1**	**1**

1. If there are no bugs, then the program runs correctly.
2. If Melbourne is in Queensland, then Brisbane is in Victoria.

Modus Ponens

$$\begin{array}{l} P \rightarrow Q \\ P \\ \hline Q \end{array}$$

A rule is sound if every model of the premises is a model of the conclusion.

Challenge: prove that modus ponens is sound.