COMP2111 Week 3
Term 1, 2019
Propositional Logic

Summary of topics

- Well-formed formulas (SYNTAX)
- Boolean Algebras
- Valuations (SEMANTICS)
- CNF/DNF
- Proof
- Natural deduction

Summary of topics

- Well-formed formulas
- Boolean Algebras
- Valuations
- CNF/DNF
- Proof
- Natural deduction

Well-formed formulas

Let $\text{PROP} = \{p, q, r, \ldots\}$ be a set of propositional letters. Consider the alphabet

$$\Sigma = \text{PROP} \cup \{\top, \bot, \neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,)\}.$$

The **well-formed formulas** (wffs) over PROP is the smallest set of words over Σ such that:

- $\top$, $\bot$ and all elements of PROP are wffs
- If φ is a wff then $\neg\varphi$ is a wff
- If φ and ψ are wffs then $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$, and $(\varphi \leftrightarrow \psi)$ are wffs.

Examples

The following are well-formed formulas:

- $(p \wedge \neg\top)$
- $\neg(p \wedge \neg\top)$
- $\neg\neg(p \wedge \neg\top)$

The following are **not** well-formed formulas:

- $p \wedge \wedge$
- $p \wedge \neg\top$
- $(p \wedge q \wedge r)$
- $\neg(\neg p)$

Conventions

To aid readability some conventions and binding rules can and will be used.

- Parentheses omitted if there is no ambiguity (e.g. $p \wedge q$)
- $\neg$ binds more tightly than $\wedge$ and $\vee$, which bind more tightly than $\rightarrow$ and $\leftrightarrow$ (e.g. $p \wedge q \rightarrow r$ instead of $((p \wedge q) \rightarrow r)$

Other conventions (rarely used/assumed in this course):

- $'$ or $\bar{\cdot}$ for $\neg$
- $+$ for $\vee$
- $\cdot$ or juxtaposition for $\wedge$
- $\wedge$ binds more tightly than $\vee$
- $\wedge$ and $\vee$ associate to the left: $p \vee q \vee r$ instead of $((p \vee q) \vee r)$
- $\rightarrow$ and $\leftrightarrow$ associate to the right: $p \rightarrow q \rightarrow r$ instead of $(p \rightarrow (q \rightarrow r))$

Conventions

To aid readability some conventions and binding rules can and will be used.

- Parentheses omitted if there is no ambiguity (e.g. $p \wedge q$)
- $\neg$ binds more tightly than $\wedge$ and $\vee$, which bind more tightly than $\rightarrow$ and $\leftrightarrow$ (e.g. $p \wedge q \rightarrow r$ instead of $((p \wedge q) \rightarrow r)$

Other conventions (rarely used/assumed in this course):

- $'$ or $\bar{\cdot}$ for $\neg$
- $+$ for $\vee$
- $\cdot$ or juxtaposition for $\wedge$
- $\wedge$ binds more tightly than $\vee$
- $\wedge$ and $\vee$ associate to the left: $p \vee q \vee r$ instead of $((p \vee q) \vee r)$
- $\rightarrow$ and $\leftrightarrow$ associate to the right: $p \rightarrow q \rightarrow r$ instead of $(p \rightarrow (q \rightarrow r))$

Parse trees

The structure of well-formed formulas (and other grammar-defined syntaxes) can be shown with a **parse tree**.

Example

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

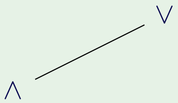
$$\vee$$

Parse trees

The structure of well-formed formulas (and other grammar-defined syntaxes) can be shown with a **parse tree**.

Example

$$((P \land \neg Q) \lor \neg(Q \to P))$$

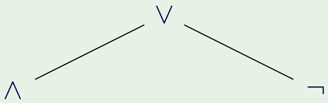


Parse trees

The structure of well-formed formulas (and other grammar-defined syntaxes) can be shown with a **parse tree**.

Example

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

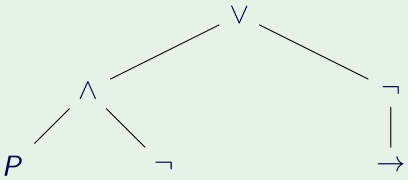


Parse trees

The structure of well-formed formulas (and other grammar-defined syntaxes) can be shown with a **parse tree**.

Example

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

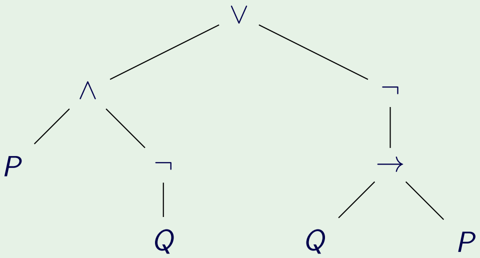


Parse trees

The structure of well-formed formulas (and other grammar-defined syntaxes) can be shown with a **parse tree**.

Example

$$((P \land \neg Q) \lor \neg(Q \to P))$$



Parse trees formally

Formally, we can define a parse tree as follows:
A parse tree is either:

- (B) A node containing $\top$;
- (B) A node containing $\perp$;
- (B) A node containing a propositional variable;
- (R) A node containing $\neg$ with a single parse tree child;
- (R) A node containing $\wedge$ with two parse tree children;
- (R) A node containing $\vee$ with two parse tree children;
- (R) A node containing $\rightarrow$ with two parse tree children; or
- (R) A node containing $\leftrightarrow$ with two parse tree children.

Summary of topics

- Well-formed formulas
- Boolean Algebras
- Valuations
- CNF/DNF
- Proof
- Natural deduction

Definition: Boolean Algebra

A *Boolen algebra* is a structure $(T, \vee, \wedge, ', 0, 1)$ where

- $0, 1 \in T$
- $\vee : T \times T \to T$ (called **join**)
- $\wedge : T \times T \to T$ (called **meet**)
- $' : T \to T$ (called **complementation**)

and the following laws hold for all $x, y, z \in T$:

commutative:
- $x \vee y = y \vee x$
- $x \wedge y = y \wedge x$

associative:
- $(x \vee y) \vee z = x \vee (y \vee z)$
- $(x \wedge y) \wedge z = x \wedge (y \wedge z)$

distributive:
- $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$
- $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$

identity: $x \vee 0 = x, \quad x \wedge 1 = x$

complementation: $x \vee x' = 1, \quad x \wedge x' = 0$

Examples of Boolean Algebras

The set of subsets of a set X:

- T : Pow(X)
- $\wedge$: $\cap$
- $\vee$: $\cup$
- $'$: ^{c}
- 0 : $\emptyset$
- 1 : X

Laws of Boolean algebra follow from Laws of Set Operations.

Examples of Boolean Algebras

The two element Boolean Algebra :

$$\mathbb{B} = (\{\texttt{true}, \texttt{false}\}, \&\&, \|, !, \texttt{false}, \texttt{true})$$

where $!, \&\&, \|$ are defined as:

- $!\texttt{true} = \texttt{false}$; $!\texttt{false} = \texttt{true}$,
- $\texttt{true}\ \&\&\ \texttt{true} = \texttt{true}$; ...
- $\texttt{true} \| \texttt{true} = \texttt{true}$; ...

NB

We will often use $\mathbb{B}$ for the two element set $\{true, false\}$. For simplicity this may also be abbreviated as $\{T, F\}$ or $\{1, 0\}$.

Examples of Boolean Algebras

Cartesian products of $\mathbb{B}$, that is n-tuples of 0's and 1's with Boolean operations, e.g. $\mathbb{B}^4$:

$$\begin{aligned}
\textit{join:} \quad & (1,0,0,1) \vee (1,1,0,0) = (1,1,0,1) \\
\textit{meet:} \quad & (1,0,0,1) \wedge (1,1,0,0) = (1,0,0,0) \\
\textit{complement:} \quad & (1,0,0,1)' = (0,1,1,0) \\
0\text{:} \quad & (0,0,0,0) \\
1\text{:} \quad & (1,1,1,1).
\end{aligned}$$

Examples of Boolean Algebras

Functions from any set S to $\mathbb{B}$; their set is denoted $\mathrm{Map}(S, \mathbb{B})$

If $f, g : S \longrightarrow \mathbb{B}$ then

- $(f \vee g) : S \longrightarrow \mathbb{B}$ is defined by $s \mapsto f(s) \parallel g(s)$
- $(f \wedge g) : S \longrightarrow \mathbb{B}$ is defined by $s \mapsto f(s) \,\&\&\, g(s)$
- $f' : S \longrightarrow \mathbb{B}$ is defined by $s \mapsto !f(s)$
- $0 : S \longrightarrow \mathbb{B}$ is the function $f(s) = \texttt{false}$
- $1 : S \longrightarrow \mathbb{B}$ is the function $f(s) = \texttt{true}$

Examples of Boolean Algebras

If $(T, \vee, \wedge, ', 0, 1)$ is a Boolean algebra, then the **dual algebra** $(T, \wedge, \vee, ', 1, 0)$ is also a Boolean Algebra. For example:

- T : $\text{Pow}(X)$
- $\wedge$: $\cup$
- $\vee$: $\cap$
- $'$: ^{c}
- 0 : X
- 1 : $\emptyset$

Every finite Boolean algebra satisfies: $|T| = 2^k$ for some k.
All algebras with the same number of elements are **isomorphic**, i.e. "structurally similar", written $\simeq$. Therefore, studying one such algebra describes properties of all.
The algebras mentioned above are all of this form

- n-tuples $\simeq \mathbb{B}^n$
- $\mathrm{Pow}(S) \simeq \mathbb{B}^{|S|}$
- $\mathrm{Map}(S, \mathbb{B}) \simeq \mathbb{B}^{|S|}$

NB

Boolean algebra as the calculus of two values is fundamental to computer circuits and computer programming.
Example: Encoding subsets as bit vectors.