COMP9020 Week 7
Term 3, 2019
Logic II: Boolean Functions and Beyond

Boolean Functions

Propositional formulas map **valuations** to $\mathbb{B}$.

$$x \wedge \neg y$$

x	y	$x \wedge \neg y$
T	T	F
T	F	T
F	T	F
F	F	F

Boolean Functions

Propositional formulas map **valuations** to $\mathbb{B}$.

$$x \wedge \neg y$$

x	y	$x \wedge \neg y$
T	T	F
T	F	T
F	T	F
F	F	F

A valuation $v : \textsc{Prop} \to \mathbb{B}$ can be seen as a sequence of $|\textsc{Prop}|$ elements of $\mathbb{B}$. For example:

$$x \mapsto \texttt{true}, y \mapsto \texttt{false}, z \mapsto \texttt{true} \qquad \text{vs} \qquad (\texttt{true}, \texttt{false}, \texttt{true})$$

Boolean Functions

Propositional formulas map **valuations** to $\mathbb{B}$.

$$x \wedge \neg y$$

x	y	$x \wedge \neg y$
T	T	F
T	F	T
F	T	F
F	F	F

A valuation $v : \textsc{Prop} \to \mathbb{B}$ can be seen as a sequence of $|\textsc{Prop}|$ elements of $\mathbb{B}$. For example:

$$x \mapsto \texttt{true}, y \mapsto \texttt{false}, z \mapsto \texttt{true} \qquad \text{vs} \qquad (\texttt{true}, \texttt{false}, \texttt{true})$$

Propositional formulas can be viewed as mapping sequences of elements of $\mathbb{B}$ to $\mathbb{B}$.

Boolean Functions

Definition

An n-ary Boolean function is a map $f : \mathbb{B}^n \rightarrow \mathbb{B}$.

Question

How many unary Boolean functions are there?
How many binary functions? n-ary?

Question

What connectives do we need to express all of them?

Summary of topics

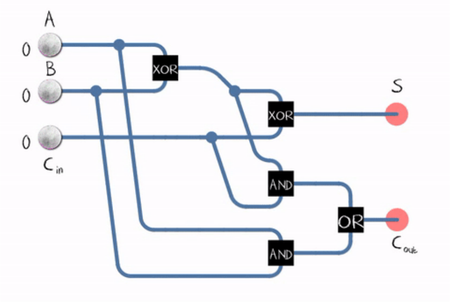
- Applications and notation
- CNF and DNF
- Karnaugh Maps
- Boolean Algebras
- Other logics

Summary of topics

- Applications and notation
- CNF and DNF
- Karnaugh Maps
- Boolean Algebras
- Other logics

Applications: Digital Circuits

Digital circuits are (sequences of) Boolean functions.



Applications

- Finding satisfying/unsatisfying assignments
- Minimizing expression size
- Defining more expressive logic

Syntax, revisited

To aid readability, we will adopt the following syntax and rules for this lecture:

- $\top$: 1
- $\bot$: 0
- $\neg P$: P' or $\overline{P}$
- $P \wedge Q$: $P \cdot Q$ or PQ (binds tighter than $\vee$)
- $\vee$ and $\cdot$ are associative and commutative

Observe that using $\overline{(\cdot)}$ obviates the need for some parentheses.

Example

$$\overline{A}BC + A\overline{B}C + AB\overline{C}$$

compared to

$$(((\neg A \wedge B) \wedge C) \vee ((A \wedge \neg B) \wedge C)) \vee ((A \wedge B) \wedge \neg C)$$

Summary of topics

- Applications and notation
- CNF and DNF
- Karnaugh Maps
- Boolean Algebras
- Other logics

Conjunctive and Disjunctive normal form

Definition

- A **literal** is an expression p or $\overline{p}$, where p is a propositional atom.

- A propositional formula is in CNF (conjunctive normal form) if it has the form
$$\bigwedge_i C_i$$
where each **clause** C_i is a disjunction of literals e.g. $p \vee q \vee \overline{r}$.
- A propositional formula is in DNF (disjunctive normal form) if it has the form
$$\bigvee_i C_i$$
where each clause C_i is a conjunction of literals e.g. $p \wedge q \wedge \overline{r}$.

CNF and DNF

- CNF and DNF are named after their top level operators; no deeper nesting of $\wedge$ or $\vee$ is permitted.
- CNF: Product of sums. DNF: Sum of products.
- We can assume in every clause (disjunct for the CNF, conjunct for the DNF) any given variable (literal) appears only once; preferably, no literal and its negation together.
  - $x \vee x = x, \quad x \wedge x = x$
  - $x \wedge \overline{x} = 0, \quad x \vee \overline{x} = 1$
  - $x \wedge 0 = 0$, $x \wedge 1 = x$, $x \vee 0 = x$, $x \vee 1 = 1$
- A preferred form for an expression is DNF, with as few terms as possible. In deriving such minimal simplifications the two basic rules are **absorption** and **combining the opposites**.

Fact

1. *Absorption:* $x \vee (x \wedge y) \equiv x$
2. *Combining the opposites:* $(x \wedge y) \vee (x \wedge \overline{y}) \equiv x$

Theorem

For every Boolean expression ϕ, there exists an equivalent expression in conjunctive normal form and an equivalent expression in disjunctive normal form.

Proof.

We show how to apply the equivalences already introduced to convert any given formula to an equivalent one in CNF, DNF is similar. $\square$

Step 1: Remove $\rightarrow$ and $\leftrightarrow$

Using the equivalences

$$\begin{aligned} x \rightarrow y &\equiv \overline{x} \vee y, \text{ and} \\ x \leftrightarrow y &\equiv (\overline{x} \vee y)(x \vee \overline{y}) \end{aligned}$$

we first eliminate all occurrences of $\rightarrow$ and $\leftrightarrow$.

Step 2: Push Negations Down

Using **De Morgan's** laws and the **double negation** rule

$$\overline{x \vee y} \equiv \overline{x} \wedge \overline{y}$$
$$\overline{x \wedge y} \equiv \overline{x} \vee \overline{y}$$
$$\overline{\overline{x}} \equiv x$$

we push negations down towards the atoms until we obtain a formula that is formed from literals using only $\wedge$ and $\vee$.

Step 3: Use Distribution to Convert to CNF

Use the distribution rule:

$$x \vee (y_1 y_2 \cdots y_n) = (x \vee y_1)(x \vee y_2) \cdots (x \vee y_n)$$

to "push" $\vee$ down the parse tree until we obtain a CNF formula.

Example

Example

Convert $\neg(\neg p \wedge ((r \wedge s) \to q))$ to CNF

$$\begin{aligned}\neg(\neg p \wedge ((r \wedge s) \to q)) &\equiv \neg(\neg p \wedge (\neg(r \wedge s) \vee q)) \\ &= \overline{\overline{p}(\overline{rs} \vee q)}\end{aligned}$$

Example

Example

Convert $\neg(\neg p \wedge ((r \wedge s) \rightarrow q))$ to CNF

$$
\begin{aligned}
\neg(\neg p \wedge ((r \wedge s) \rightarrow q)) &\equiv \neg(\neg p \wedge (\neg(r \wedge s) \vee q)) \\
&= \overline{\overline{p}(\overline{rs} \vee q)} \\
&\equiv \overline{\overline{p}} \vee \overline{\overline{rs} \vee q} \\
&\equiv p \vee \overline{\overline{rs}}\,\overline{q} \\
&\equiv p \vee rs\overline{q}
\end{aligned}
$$

Example

Example

Convert $\neg(\neg p \wedge ((r \wedge s) \to q))$ to CNF

$$\begin{aligned}
\neg(\neg p \wedge ((r \wedge s) \to q)) &\equiv \neg(\neg p \wedge (\neg(r \wedge s) \vee q)) \\
&= \overline{\overline{p}(\overline{rs} \vee q)} \\
&\equiv \overline{\overline{p}} \vee \overline{\overline{rs} \vee q} \\
&\equiv p \vee \overline{\overline{rs}}\,\overline{q} \\
&\equiv p \vee rs\overline{q} \\
&\equiv (p \vee r)(p \vee s\overline{q}) \\
&\equiv (p \vee r)(p \vee s)(p \vee \overline{q}) \qquad \text{CNF}
\end{aligned}$$

Canonical Form DNF

Given a Boolean expression E, we can construct an equivalent DNF E^{dnf} from the lines of the truth table where E is true: Given an assignment v from $\{x_1 \ldots x_i\}$ to $\mathbb{B}$, define the literal

$$\ell_i = \begin{cases} x_i & \text{if } v(x_i) = \texttt{true} \\ \overline{x_i} & \text{if } v(x_i) = \texttt{false} \end{cases}$$

and a product $T_v = \ell_1 \ell_2 \cdots \ell_n$.

Example

If $v(x_1) = \texttt{true}$ and $v(x_2) = \texttt{false}$ then $T_v = x_1 \overline{x_2}$

The **canonical DNF** of E is

$$E^{dnf} = \bigvee_{v(E)=\texttt{true}} T_v$$

Example

If E is defined by

x	y	E
F	F	T
F	T	F
T	F	T
T	T	T

then $E^{dnf} = (\overline{x}\,\overline{y}) \vee (x\overline{y}) \vee (xy)$
Note that this can be simplified to either

$$\overline{y} \vee (xy)$$

or

$$(\overline{x}\,\overline{y}) \vee x$$

Canonical CNF

After pushing negations down, the negation of a DNF is a CNF (and vice versa).

$\Rightarrow$ Given an expression E, we can obtain an equivalent CNF by finding a DNF for $\neg E$ and then applying De Morgan's laws.

$\Leftrightarrow$ Look at rows in the truth table of E that contain `false`. Compute E^{dnf}. Swap $\vee$ and $\wedge$ and *negate* the literals.

Example

If E is defined by

x	y	E
F	F	F
F	T	F
T	F	T
T	T	F

then $E^{cnf} = (x \vee y)(x \vee \overline{y})(\overline{x} \vee \overline{y})$.

Exercise

Exercises

$\boxed{10.2.3}$ Find the canonical DNF form of each of the following expressions in variables x, y, z

- xy
- $\overline{z}$
- $xy + \overline{z}$
- 1

Exercise

Exercises

$\boxed{10.2.3}$ Find the canonical DNF form of each of the following expressions in variables x, y, z

- xy?
- $\overline{z}$?
- $xy + \overline{z}$?
- 1?

Summary of topics

- Applications and notation
- CNF and DNF
- Karnaugh Maps
- Boolean Algebras
- Other logics

Karnaugh Maps

For up to four variables (propositional symbols) a diagrammatic method of simplification called **Karnaugh maps** works quite well. For every propositional function of $k = 2, 3, 4$ variables we construct a rectangular array of 2^k cells. We mark the squares corresponding to the value `true` with eg "$+$" and try to cover these squares with as few rectangles with sides 1 or 2 or 4 as possible.

Example

	yz	$y\bar{z}$	$\bar{y}\,\bar{z}$	$\bar{y}z$
x	$+$	$+$		$+$
$\bar{x}$	$+$		$+$	$+$

For optimisation, the idea is to cover the $+$ squares with the minimum number of rectangles. One *cannot* cover any empty cells.

- The rectangles can go 'around the corner'/the actual map should be seen as a *torus*.
- Rectangles must have sides of 1, 2 or 4 squares (three adjacent cells are useless).

Example

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x	$+$	$+$		$+$
$\bar{x}$	$+$		$+$	$+$

For optimisation, the idea is to cover the $+$ squares with the minimum number of rectangles. One *cannot* cover any empty cells.

- The rectangles can go 'around the corner'/the actual map should be seen as a *torus*.
- Rectangles must have sides of 1, 2 or 4 squares (three adjacent cells are useless).

Example

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x	$+$	$+$		$+$
$\bar{x}$	$+$		$+$	$+$

$$E = (xy) \vee$$

Canonical form would consist of writing all cells separately (6 clauses).

For optimisation, the idea is to cover the $+$ squares with the minimum number of rectangles. One *cannot* cover any empty cells.

- The rectangles can go 'around the corner'/the actual map should be seen as a *torus*.
- Rectangles must have sides of 1, 2 or 4 squares (three adjacent cells are useless).

Example

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x	$+$	$+$		$+$
$\bar{x}$	$+$		$+$	$+$

$$E = (xy) \vee (\bar{x}\bar{y}) \vee$$

Canonical form would consist of writing all cells separately (6 clauses).

For optimisation, the idea is to cover the $+$ squares with the minimum number of rectangles. One *cannot* cover any empty cells.

- The rectangles can go 'around the corner'/the actual map should be seen as a *torus*.
- Rectangles must have sides of 1, 2 or 4 squares (three adjacent cells are useless).

Example

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x	$+$	$+$		$+$
$\bar{x}$	$+$		$+$	$+$

$$E = (xy) \lor (\bar{x}\bar{y}) \lor z$$

Canonical form would consist of writing all cells separately (6 clauses).

Exercise

Exercise

10.6.6(c)

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	+	+		+
$w\bar{x}$	+	+	+	+
$\bar{w}\bar{x}$			+	+
$\bar{w}x$	+			+

Exercise

Exercise

$\boxed{10.6.6(c)}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	$+$	$+$		$+$
$w\bar{x}$	$+$	$+$	$+$	$+$
$\bar{w}\bar{x}$			$+$	$+$
$\bar{w}x$	$+$			$+$

?

Summary of topics

- Applications and notation
- CNF and DNF
- Karnaugh Maps
- Boolean Algebras
- Other logics

Boolean Algebras II: The bigger picture

Proofs in Set Theory and Logical Equivalence proofs “look” the same.

Question

Is there an underlying reason?

Definition: Boolean Algebra

A *Boolean 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

Example

The set of subsets of a set X:

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

Laws of Boolean algebra follow from Laws of Set Operations.

Examples of Boolean Algebras

Example

The two element Boolean Algebra :

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

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

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

Examples of Boolean Algebras

Example

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

Example

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 $s \mapsto f(s) \,\|\, g(s)$
- $(f \wedge g) : S \longrightarrow \mathbb{B}$ 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 $f(s) = \texttt{true}$

Proofs in Boolean Algebras

Show an identity holds using the laws of Boolean Algebra, then that identity holds **in all Boolean Algebras**.

Example

In all Boolean Algebras

$$x \wedge x = x$$

for all $x \in T$.

Proof:

$$\begin{aligned} x &= x \wedge 1 && \text{[Identity]} \\ &= x \wedge (x \vee x') && \text{[Complement]} \\ &= (x \wedge x) \vee (x \wedge x') && \text{[Distributivity]} \\ &= (x \wedge x) \vee 0 && \text{[Complement]} \\ &= (x \wedge x) && \text{[Identity]} \end{aligned}$$

Duality

Definition

If E is an expression defined using variables (x, y, z, etc), constants (0 and 1), and the operations of Boolean Algebra ($\wedge$, $\vee$, and $'$) then dual(E) is the expression obtained by replacing $\wedge$ with $\vee$ (and vice-versa) and 0 with 1 (and vice-versa).

Definition

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

Theorem (Principle of duality)

If you can show $E_1 = E_2$ using the laws of Boolean Algebra, then $dual(E_1) = dual(E_2)$.

Duality

Example

We have shown $x \wedge x = x$.

By duality: $x \vee x = x$.

Summary of topics

- Applications and notation
- CNF and DNF
- Karnaugh Maps
- Boolean Algebras
- Other logics

Limitations to Propositional Logic

Propositional logic is unable to capture several useful phenomena:

- Spatial/temporal dependence (e.g. P holds **after** Q holds)
- Belief and knowledge (e.g. I know that you know that X holds)
- Relationships between propositions (e.g. "The sky is blue" and "my eyes are blue")
- Quantification (e.g. "All men are mortal")

Beyond Propositional Logic

Modal logic: Introduce **modalities** to capture statement qualifying.

Example

Temporal logic:

- $\mathcal{F}\,\varphi$: φ will be true at some point in the future
- $\mathcal{G}\,\varphi$: φ will be true at all points in the future
- $\varphi\,\mathcal{U}\,\psi$: φ will be true until ψ holds

Beyond Propositional Logic

First order logic/Predicate logic: Add relations (predicates) and quantifiers to capture relationships between propositions.

Example

- P: All men are mortal:
- Q: Socrates is a man:
- R: Socrates is mortal:

In propositional logic, there is no connection between P, Q and R: it is not the case that $P, Q \models R$.

Beyond Propositional Logic

First order logic/Predicate logic: Add relations (predicates) and quantifiers to capture relationships between propositions.

Example

- P: All men are mortal: $\forall x\, \texttt{Man}(x) \rightarrow \texttt{Mortal}(x)$
- Q: Socrates is a man:
- R: Socrates is mortal:

In propositional logic, there is no connection between P, Q and R: it is not the case that $P, Q \models R$.

Beyond Propositional Logic

First order logic/Predicate logic: Add relations (predicates) and quantifiers to capture relationships between propositions.

Example

- P: All men are mortal: $\forall x \, \texttt{Man}(x) \rightarrow \texttt{Mortal}(x)$
- Q: Socrates is a man: $\texttt{Man(Socrates)}$
- R: Socrates is mortal:

In propositional logic, there is no connection between P, Q and R: it is not the case that $P, Q \models R$.

Beyond Propositional Logic

First order logic/Predicate logic: Add relations (predicates) and quantifiers to capture relationships between propositions.

Example

- P: All men are mortal: $\forall x\, \texttt{Man}(x) \rightarrow \texttt{Mortal}(x)$
- Q: Socrates is a man: $\texttt{Man(Socrates)}$
- R: Socrates is mortal: $\texttt{Mortal(Socrates)}$

In propositional logic, there is no connection between P, Q and R: it is not the case that $P, Q \models R$.

In first-order logic you can show $P, Q \models R$.

Beyond Propositional Logic

First order logic/Predicate logic: Add relations (predicates) and quantifiers to capture relationships between propositions.

Example

- P: All men are mortal: $\forall x\, \texttt{Man}(x) \rightarrow \texttt{Mortal}(x)$
- Q: Socrates is a man: $\texttt{Man(Socrates)}$
- R: Socrates is mortal: $\texttt{Mortal(Socrates)}$

In propositional logic, there is no connection between P, Q and R: it is not the case that $P, Q \models R$.

In first-order logic you can show $P, Q \models R$.

Second order logic: Add quantification of relations.

Limitations

More expressive logics require more complex semantics.

- Logical equivalence harder to show
- Entailment harder to show
- Connections between different concepts not so straightforward

Example

In Temporal Logic, a valuation is a function $v : \textsc{Prop} \times \mathbb{N} \to \mathbb{B}$ – i.e. truth tables that change over time.