Section 4

Normal forms, Boolean functions

Normal forms

Recall: $\to$ is 'superfluous' since it can be expressed using $\neg$ and $\vee$

$$\begin{aligned} A \to B &\equiv \neg A \vee B, \\ &\equiv \neg(A \wedge \neg B), \\ &\equiv \neg\neg(A \to B), \\ &\text{etc} \end{aligned}$$

Problem: Propositional formulas have *very many* equivalent forms

Solution: Transform formulas to **normal form**

Normal forms have many benefits:

- simplification of formulas $\rightsquigarrow$ improved intelligibility
- focusses considerations on a small subset of 'essential' connectives and 'patterns'

Flattening conjunctions and disjunctions

The associativity laws for $\wedge$ allow us to write formulas with a sequence of conjunctions without brackets

Example

Instead of

$$P_1 \wedge (P_2 \wedge (P_3 \wedge P_4))$$

we write

$$P_1 \wedge P_2 \wedge P_3 \wedge P_4$$

Similarly we flatten disjunctions

Example

We write

$$P_1 \vee P_2 \vee \neg Q \vee R$$

instead of

$$(P_1 \vee P_2) \vee (\neg Q \vee R)$$

Conjunctive normal form

Definition

A propositional formula is in **conjunctive normal form** (**CNF**),
if it is either

$\top$, $\bot$, or
a conjunction of disjunctions of propositional variables, negated propositional variables, $\top$ or $\bot$

Examples

$(P \vee \neg R) \wedge (Q \vee \neg R \vee \neg P) \wedge (Q \vee \neg R)$ in CNF
$(P \vee \neg R) \wedge (Q \vee \neg(R \wedge P))$
$(R \to P) \wedge (Q \vee \neg R \vee \neg P)$

Observations

Being in CNF means

- No connectives other than: $\top$, $\bot$, $\wedge$, $\vee$, $\neg$
- $\neg$ can only appear immediately in front of propositional variables
- No $\wedge$ inside $\vee$

An algorithm for transformation to CNF

Step 1: Eliminate all $\rightarrow$ and $\leftrightarrow$ connectives by using

$$\begin{aligned} A \rightarrow B &\equiv \neg A \vee B \\ A \leftrightarrow B &\equiv (\neg A \vee B) \wedge (A \vee \neg B) \end{aligned}$$

Step 2: Push $\neg$ inwards as far as possible by using the De Morgan's laws

$$\begin{aligned} \neg(A_1 \wedge A_2 \wedge \ldots \wedge A_n) &\equiv \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_n \\ \neg(A_1 \vee A_2 \vee \ldots \vee A_n) &\equiv \neg A_1 \wedge \neg A_2 \wedge \ldots \wedge \neg A_n \end{aligned}$$

Step 3: Eliminate all sequences of negations by using

$$\neg\neg A \equiv A$$

Step 4: Eliminate all conjunctions inside disjunctions by rewriting

$$(A^1_1 \wedge \ldots \wedge A^1_{m_1}) \vee (A^2_1 \wedge \ldots \wedge A^2_{m_2}) \vee \ldots \vee (A^n_1 \wedge \ldots \wedge A^n_{m_n})$$

$$\text{to} \quad \bigwedge_{i=1}^{k} \left(A^1_{i_1} \vee \ldots \vee A^n_{i_n}\right)$$

Remarks

Example for Step 4

Multiplying out $(A \wedge B) \vee (C \wedge D)$ gives

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

- We implicitly assume $\wedge$ and $\vee$ are associative
- The steps of the algorithm need to be applied repeatedly and exhaustively, until none of the steps can be applied any more

Theorem

Using the algorithm every propositional formula can be transformed into an equivalent formula in conjunctive normal form

Applying the algorithm

Example

$$\begin{aligned} & P \vee \neg(P \vee (Q \vee R)) && \\ \equiv\ & P \vee \neg(P \vee Q \vee R) && \text{associativity } \vee \\ \equiv\ & P \vee (\neg P \wedge \neg Q \wedge \neg R) && \text{Step 2, De Morgan} \\ \equiv\ & (P \vee \neg P) \wedge (P \vee \neg Q) \wedge (P \vee \neg R) && \text{Step 4} \end{aligned}$$

Further simplification

We can often significantly simplify the obtained formula by using

- the fundamental laws involving $\top$ and $\bot$
- idempotence, associativity and commutativity of $\wedge$ and $\vee$, and
- the absorption laws:

$$
\begin{aligned}
A \wedge (A \vee B) &\equiv A \\
A \vee (A \wedge B) &\equiv A
\end{aligned}
$$

Disjunctive normal form

Disjunctive normal forms have the same definition and properties as conjunctive normal forms; only the role of $\wedge$ and $\vee$ are reversed

Definition

A propositional formula is in **disjunctive normal form** (**DNF**), if it is either

$\top$, $\bot$, or

a disjunction of conjunctions of propositional variables, negated propositional variables, $\top$ or $\bot$

Example

$P \vee (\neg P \wedge \neg Q \wedge \neg R)$ in DNF

An algorithm for transformation to DNF

Step 1: Eliminate all $\rightarrow$ and $\leftrightarrow$ connectives by using the laws

$$\begin{aligned} A \rightarrow B &\equiv \neg A \vee B \\ A \leftrightarrow B &\equiv (\neg A \vee B) \wedge (A \vee \neg B) \end{aligned}$$

Step 2: Push $\neg$ inwards as far as possible by using De Morgan's laws

$$\begin{aligned} \neg(A_1 \wedge A_2 \wedge \ldots \wedge A_n) &\equiv \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_n \\ \neg(A_1 \vee A_2 \vee \ldots \vee A_n) &\equiv \neg A_1 \wedge \neg A_2 \wedge \ldots \wedge \neg A_n \end{aligned}$$

Step 3: Eliminate all sequences of negations by using the law

$$\neg\neg A \equiv A$$

Step 4: Eliminate all disjunctions inside conjunctions by rewriting

$$(A^1_1 \vee \ldots \vee A^1_{m_1}) \wedge (A^2_1 \vee \ldots \vee A^2_{m_2}) \wedge \ldots \wedge (A^n_1 \vee \ldots \vee A^n_{m_n})$$

$$\text{to} \quad \bigvee_{i=1}^{k} \left(A^1_{i_1} \wedge \ldots \wedge A^n_{i_n}\right)$$

Applying the algorithm

Steps 1–3 of the DNF algorithm are the same as in the CNF algorithm

Example

$$\begin{aligned} & P \vee \neg(P \vee (Q \vee R)) \\ \equiv\; & P \vee (\neg P \wedge \neg Q \wedge \neg R) \qquad \text{using Steps 1–3, as before} \end{aligned}$$

Finding disjunctive normal forms using truth tables

Let A be a formula with propositional variables $P_1, \ldots, P_n$

Algorithm 2 to compute DNF for A

Step 1: Compute the truth table for A

Step 2: For each row i in the truth table, let B_i be the formula

$$\tilde{P}_1 \wedge \ldots \wedge \tilde{P}_n$$

that corresponds exactly to the valuation of the propositional variables, if the entry for A is $\mathbf{1}$.
Otherwise, let $B_i = \perp$.

P_1	P_2	P_3	correspondent $\tilde{P}_1 \wedge \tilde{P}_2 \wedge \tilde{P}_3$
$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$P_1 \wedge P_2 \wedge P_3$
$\mathbf{1}$	$\mathbf{1}$	$\mathbf{0}$	$P_1 \wedge P_2 \wedge \neg P_3$
$\vdots$			
$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\neg P_1 \wedge \neg P_2 \wedge \neg P_3$

Then A has the disjunctive normal form $B_1 \vee \ldots \vee B_k$

Applying the algorithm

Finding the DNF for $P \leftrightarrow Q$:

P	Q	$P \leftrightarrow Q$
$\mathbf{1}$	$\mathbf{1}$	
$\mathbf{1}$	$\mathbf{0}$	
$\mathbf{0}$	$\mathbf{1}$	
$\mathbf{0}$	$\mathbf{0}$	

$P \leftrightarrow Q$ has as DNF

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

Remarks

- CNFs can also be directly constructed from truth tables (How is an exercise)
- These methods provide easy ways for computing DNFs and CNFs
- But, the complete truth table always needs to be constructed first – this requires exponential time

$$
\begin{aligned}
2^5 &= 32 \\
2^6 &= 64 \\
&\vdots \\
2^{10} &= 1024 \\
&\vdots
\end{aligned}
$$

Boolean functions

Recall, $\mathbb{B} = \{\mathbf{1}, \mathbf{0}\}$

Definition

A **Boolean function** f is a function from $\mathbb{B}^n$ to $\mathbb{B}$.

Each logical connective can be interpreted as a Boolean function

- $\neg$ can be interpreted as the function $f_\neg : \mathbb{B} \longrightarrow \mathbb{B}$ defined by:

x	$f_\neg$
1	**0**
0	**1**

$$
\begin{array}{c}
x \mapsto f_\neg(x) \\
\hline
\mathbf{1} \mapsto \mathbf{0} \\
\mathbf{0} \mapsto \mathbf{1}
\end{array}
$$

- $\wedge$ can be interpreted as the function $f_\wedge : \mathbb{B}^2 \longrightarrow \mathbb{B}$ defined by:

x_1	x_2	$f_\wedge$
1	**1**	**1**
1	**0**	**0**
0	**1**	**0**
0	**0**	**0**

$$
\begin{array}{ll}
(x_1, x_2) & \mapsto f_\wedge(x_1, x_2) \\
\hline
(\mathbf{1}, \mathbf{1}) & \mapsto \mathbf{1} \\
(\mathbf{1}, \mathbf{0}) & \mapsto \mathbf{0} \\
(\mathbf{0}, \mathbf{1}) & \mapsto \mathbf{0} \\
(\mathbf{0}, \mathbf{0}) & \mapsto \mathbf{0}
\end{array}
$$

Possible one-place Boolean functions

Conversely, each Boolean function can be viewed as defining a different logical connective (not all of these will be useful)

The possible one-place Boolean functions are:

x	f_1	f_2	f_3	f_4
1	**1**	**1**	**0**	**0**
0	**1**	**0**	**1**	**0**

We note that, for each $x \in \mathbb{B}$:

$$
\begin{aligned}
f_1(x) &= \mathbf{1} && \text{we write } f_1(x) = \top \\
f_4(x) &= \mathbf{0} && \text{we write } f_4(x) = \bot \\
f_3(x) &= \neg x && \\
f_2(x) &= x && \text{(identity function)}
\end{aligned}
$$

The possible two-place Boolean functions

x_1	x_2	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**
0	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**
0	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**

x_1	x_2	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
1	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**
0	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**
0	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**

We have, for each $(x_1, x_2) \in \mathbb{B}^2$:

$$
\begin{aligned}
f_1(x_1, x_2) &= \top & f_{16}(x_1, x_2) &= \bot \\
f_4(x_1, x_2) &= x_1 & f_6(x_1, x_2) &= x_2 \\
f_2(x_1, x_2) &= x_1 \vee x_2 & f_8(x_1, x_2) &= x_1 \wedge x_2 \qquad \ldots
\end{aligned}
$$

Boolean functions as propositional formulas

Theorem

Every Boolean function $f : \mathbb{B}^n \longrightarrow \mathbb{B}$ *can be expressed as a propositional formula using* $x_1, \ldots, x_n$ *as propositional variables and the connectives* $\neg$, $\wedge$ *and* $\vee$

Proof.

Use the truth table method for computing disjunctive normal forms $\square$

Boolean functions are useful for:

- answering questions relating to the adequacy of sets of connectives
- designing electronic switching circuits, see COMP12111