CS262 Logic and Verification

Lecture 3: Boolean algebra

Logical equivalence

Formulas are truth functions: each valuation maps the formula to T or F.

Formulas representing the same truth function are called **logically equivalent**.

Example:

p	q	$p \to q$	$\neg p$	$\vee$	q
T	T	T	F	T	
T	F	F	F	F	
F	T	T	T	T	
F	F	T	T	T	

We write: $p \to q = \neg p \vee q$

We can replace logically equivalent formulas by each other.

This also works for compound formulas: $X \to Y = \neg X \vee Y$

Laws of Boolean algebra

$X \vee X = X \quad X \wedge X = X$	Idempotence
$X = \neg\neg X$	Double negation
$X \to Y = \neg X \vee Y$	Replacing implication
$X \wedge Y = Y \wedge X \quad X \vee Y = Y \vee X$	Commutativity
$(X \wedge Y) \wedge Z = X \wedge (Y \wedge Z)$	Associativity
$(X \vee Y) \vee Z = X \vee (Y \vee Z)$	
$\neg(X \vee Y) = \neg X \wedge \neg Y$	De Morgan's Laws
$\neg(X \wedge Y) = \neg X \vee \neg Y$	
$X \wedge (Y \vee Z) = (X \wedge Y) \vee (X \wedge Z)$	Distributivity
$X \vee (Y \wedge Z) = (X \vee Y) \wedge (X \vee Z)$	
$(X \wedge Y) \to Z = X \to (Y \to Z)$	Exportation
$X \to Y = X \to (X \wedge Y)$	Absorption Law
$(X \to Y) \wedge (X \to \neg Y) = \neg X$	Contradiction
$X \vee \top = \top \quad X \wedge \top = X$	Neutral elements
$X \vee \bot = X \quad X \wedge \bot = \bot$	
$X \vee \neg X = \top \quad X \wedge \neg X = \bot$	

and many more...

Proofs

These laws can be verified by truth tables, or by deriving them from other laws.

Example: $(X \to Y) \wedge (X \to \neg Y) \equiv \neg X$ (contradiction law)

$$
\begin{aligned}
&(X \to Y) \wedge (X \to \neg Y) && \text{(replace implication)} \\
&= (\neg X \vee Y) \wedge (\neg X \vee \neg Y) && \text{(distributivity)} \\
&= \neg X \vee (Y \wedge \neg Y) && \text{(neutral elements)} \\
&= \neg X \vee \bot && \text{(neutral elements)} \\
&= \neg X
\end{aligned}
$$

Rewriting/simplifying formulas

Laws of Boolean algebra can be used to simplify complex formulas.

Example:

$$
\begin{aligned}
&((p \wedge q) \rightarrow (r \wedge s)) \vee (p \wedge q) && \text{(replace implication)}\\
&= \big(\neg(p \wedge q) \vee (r \wedge s)\big) \vee (p \wedge q) && \text{(commutativity)}\\
&= \big((r \wedge s) \vee \neg(p \wedge q)\big) \vee (p \wedge q) && \text{(associativity)}\\
&= (r \wedge s) \vee \big(\neg(p \wedge q) \vee (p \wedge q)\big) && \text{(neutral elements)}\\
&= (r \wedge s) \vee \top && \text{(neutral elements)}\\
&= \top
\end{aligned}
$$

This is called **equational reasoning**.

Problem: It is difficult to automate. Which rule to apply next?

Exercise

Prove the following useful equivalences (by equational reasoning using the aforementioned laws):

- $X \to Y = \neg Y \to \neg X$
- $(X \vee Y) \to Z = (X \to Z) \wedge (Y \to Z)$
- $X \to (Y \wedge Z) = (X \to Y) \wedge (X \to Z)$

Getting rid of logical operators

We know that $X \to Y = \neg X \vee Y$, so any occurence of $\to$ can be replaced.

Hence $\dot{\to}$ is not strictly necessary in our language.

Question: What is a **small/minimal set of connectives** that allows us to express *any* Boolean function?

This will be discussed in the exercises.