CS262 Logic and Verification

Lecture 3: Boolean algebra

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

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

We can replace logically equivalent formulas by each other.

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

Laws of Boolean algebra

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

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)

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

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

Example:

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

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):

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

Getting rid of logical operators

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

Hence $\to$ 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.