CSCE 222
Discrete Structures for Computing

Propositional Logic

Dr. Hyunyoung Lee

Based on slides by Andreas Klappenecker

Propositions

A proposition is a declarative sentence that is either true or false (but not both).

Examples:

- College Station is the capital of the USA.
- There are fewer politicians in College Station than in Washington, D.C.
- 1+1=2
- 2+2=5

Propositional Variables

A variable that represents propositions is called a propositional variable.

For example: p, q, r, ...

[Propositional variables in logic play the same role as numerical variables in arithmetic.]

Propositional Logic

The area of logic that deals with propositions is called propositional logic.

In addition to propositional variables, we have logical connectives (or operators) such as
not (for negation), and (for conjunction), or (for disjunction), exclusive or (for exclusive disjunction), conditional (for implication), and biconditional.

Formation Tree

Each logical connective is enclosed in parentheses, except for the negation connective ¬. Thus, we can associate a unique binary tree to each proposition, called the formation tree.

The formation tree contains all subformulas of a formula, starting with the formula at its root and breaking it down into its subformulas until you reach the propositional variables at its leafs.

Formation Tree

A formation tree of a proposition p has a root labeled with p and satisfies the following rules:

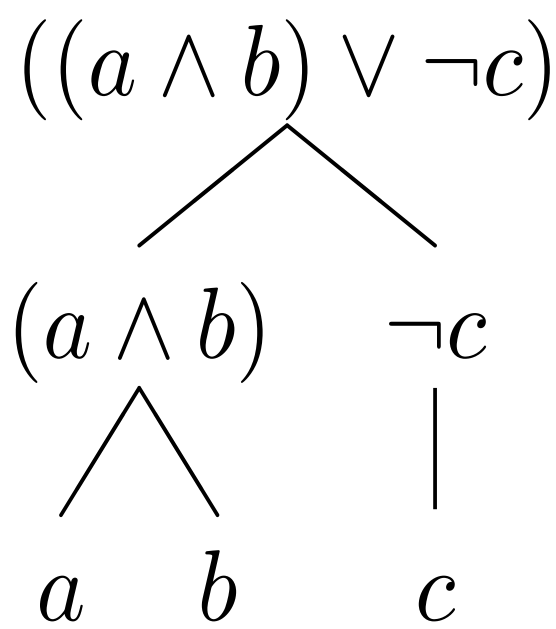
T1. Each leaf is an occurrence of a propositional variable in p.

T2. Each internal node with a single successor is labeled by a subformula ¬q of p and has q as a successor.

T3. Each internal node with two successors is labeled by a subformula aXb of p with X in $\{\wedge, \vee, \oplus, \rightarrow, \leftrightarrow\}$ and has a as a left successor and b as a right successor.

Example

The formation tree of the formula $((a \wedge b) \vee \neg c)$ is given by



Assigning Meanings to Formulas

We know that each formula corresponds to a unique binary tree.

We can evaluate the formula by

- giving each propositional variable a truth value
- defining the meaning of each logical connective
- propagate the truth values from the leafs to the root in a unique way, so that we get an unambiguous evaluation of each formula.

Semantics

Let B={t,f}. Assign to each connective x a function Mx: B->B that determines the semantics of x.

P	$M_{\neg}(P)$
f	t
t	f

P	Q	$M_{\wedge}(P,Q)$	$M_{\vee}(P,Q)$	$M_{\oplus}(P,Q)$	$M_{\rightarrow}(P,Q)$	$M_{\leftrightarrow}(P,Q)$
f	f	f	f	f	t	t
f	t	f	t	t	t	f
t	f	f	t	t	f	f
t	t	t	t	f	t	t

Summary

Informally, we can summarize the meaning of the connectives in words as follows:

1. The and connective ($a \wedge b$) is true if and only if both a and b are true.
2. The or connective ($a \vee b$) is true if and only if at least one of a, b is true.
3. The exclusive or ($a \oplus b$) is true if and only if precisely one of a, b is true.
4. The implication ($a \rightarrow b$) is false if and only if the premise a is true and the conclusion b is false.
5. The biconditional ($a \longleftrightarrow b$) is true if and only if the truth values of a and b are the same.

Equivalences and Applications

Remarks

- In our formal introduction of propositional logic, we used a strict syntactic structure with full parenthesizing (except negations).
- From now on, we will be more relaxed about the syntax and allow to drop enclosing parentheses.
- This can introduce ambiguity, which is resolved by introducing operator precedence rules (from highest to lowest) as follows. 1) negation, 2) and, 3) or, xor, 4) conditional, 5) biconditional

Tautologies

A proposition p is called a tautology if and only if in a truth table it always evaluates to true regardless of the assignment of truth values to its variables.

Example:

p	$\neg p$	$p \vee \neg p$
F	T	T
T	F	T

Example of a Tautology

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$	$(\neg p \vee q) \leftrightarrow (p \rightarrow q)$
F	F	T	T	T	T
F	T	T	T	T	T
T	F	F	F	F	T
T	T	F	T	T	T

Roughly speaking, this means that $\neg p \vee q$ has the same meaning as $p \rightarrow q$.

Logical Equivalence

Two propositions p and q are called logically equivalent if and only if $p \leftrightarrow q$ is a tautology.

We write $p \equiv q$ if and only if p and q are logically equivalent. We have shown that $(\neg p \vee q) \equiv (p \rightarrow q)$. In general, we can use truth tables to establish logical equivalences.

De Morgan's Laws (1)

Theorem: $\neg(p \wedge q) \equiv \neg p \vee \neg q$.

Proof: We can use a truth table:

p	q	$(p \wedge q)$	$\neg(p \wedge q)$	$\neg p$	$\neg q$	$\neg p \vee \neg q$
F	F	F	T	T	T	T
F	T	F	T	T	F	T
T	F	F	T	F	T	T
T	T	T	F	F	F	F

De Morgan's Laws (2)

Theorem: $\neg(p \vee q) \equiv \neg p \wedge \neg q$

Proof: Use truth table, as before. See Example 2 on page 27 of our textbook.

Commutative, Associative, and Distributive Laws

Commutative laws:

$$p \vee q \equiv q \vee p$$

$$p \wedge q \equiv q \wedge p$$

Associative laws:

$$(p \vee q) \vee r \equiv p \vee (q \vee r)$$

$$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$$

Distributive laws:

$$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$$

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

Double Negation Law

We have $\neg(\neg p) \equiv p$

p	$\neg p$	$\neg(\neg p)$
F	T	F
T	F	T

Logical Equivalences

You find many more logical equivalences listed in Table 6 on page 29.

You should very carefully study these laws.

Logical Equivalences in Action

Let us show that $\neg(p \rightarrow q)$ and $p \wedge \neg q$ are logically equivalent without using truth tables.

$$\begin{aligned} \neg(p \rightarrow q) &\equiv \neg(\neg p \vee q) \quad \text{(previous result)} \\ &\equiv \neg(\neg p) \wedge \neg q \quad \text{(de Morgan)} \\ &\equiv p \wedge \neg q \quad \text{(double negation law)} \end{aligned}$$

[Arguments using laws of logic are more desirable than truth tables unless the number of propositional variables is tiny.]