CS262 Logic and Verification
Lecture 2: Basics of propositional logic

Propositional logic

A formal system for reasoning about propositions

Proposition = **Formula** = **Statement**

Which of the following are propositions?

- It is raining.
- Every natural number has a successor.
- Have a nice day.
- You may have soup or melon for your starter.
- Will it never end?

Those for which it makes sense to ask: is it true or false?

Atomic vs. compound propositions

Consider the following propositions:

- It is raining.
- Either it is raining or I will hang the washing outside, and if I hang the washing outside then it will dry more quickly.
- If it is raining then I take my umbrella. I have not taken my umbrella. Therefore it is not raining.

The form of an argument

Consider the following two propositions:

- If it is raining then I take my umbrella. I have not taken my umbrella. Therefore it is not raining.
- If I get all the questions right then I get full marks. I have not got full marks. Therefore I have not got all the questions right.

It's the **form** of the argument that matters rather than the meanings of the sentences.

Our propositional language

Atomic formulas are the most basic propositions (indecomposable)

E.g: $p =$ "It is raining"
$q =$ "I take my umbrella"

Logical connectives/operators allow us to build more complicated propositions

$\neg$	negation	"not p"	"It is not raining."
$\wedge$	conjunction	"p and q"	"It is raining and I take my umbrella"
$\vee$	disjunction	"p or q"	"It is raining or I take my umbrella"
$\rightarrow$	implication	"p implies q"	"If it is raining then I take my umbrella"

More binary connectives in the exercises. Will use $\circ \in \{\wedge, \vee, \rightarrow, \ldots\}$ as general notation.

Syntax of propositional formulas

Propositional formulas are exactly those that can be constructed by a finite number of applications of the following rules:

- Propositional variables $p, q, r, \ldots$ and $\top$ and $\bot$ are **atomic formulas**.
- If X is a formula then so is $\neg X$.
- If X and Y are formulas then so is $(X \circ Y)$, where $\circ$ is any binary connective (e.g., $\circ = \{\wedge, \vee, \rightarrow, \ldots\}$).

Examples:

- $((p \vee \neg p) \wedge (\neg p \vee \bot))$
- $\neg\neg\neg\top$

Counterexamples:

- $p \vee q$
- $p \wedge q \rightarrow r$

Conventions

May leave out brackets where there is no ambiguity:

- $p \lor q = (p \lor q)$
- $p \land q \land r = (p \land q) \land r = p \land (q \land r)$
- $p \lor q \lor r = (p \lor q) \lor r = p \lor (q \lor r)$

But the following is ambiguous:

- $p \to q \to r$, because $(p \to q) \to r \neq p \to (q \to r)$

Parse tree

We can inductively construct a **parse tree** for any propositional formula:

- For any atomic formula X, the parse tree has a single node labeled X.
- The parse tree for $\neg X$ has a node labeled $\neg$ as the root, and the parse tree for X as the unique subtree of the root.
- The parse tree for $(X \circ Y)$ has a node labeled $\circ$ as the root, and the parse trees for X and Y as its left and right subtree.

Example: $((p \wedge \neg q) \rightarrow (r \vee (q \rightarrow \neg p)))$

Each subtree is a subformula.

Exercise

Draw parse trees for the following formulas:

- $(p \wedge (\neg\top \rightarrow r))$
- $(p \rightarrow ((q \vee r) \rightarrow \neg\neg\neg r))$

Induction and recursion

Based on our inductive definition, we can define recursive functions on formulas.

Example:

$$\begin{aligned} \deg(A) &= 0 \text{ if } A \text{ is an atomic formula,} \\ \deg(\neg X) &= 1 + \deg(X) \text{ if } X \text{ is a formula,} \\ \deg(X \circ Y) &= \deg(X) + \deg(Y) + 1 \text{ if } X \text{ and } Y \text{ are formulas.} \end{aligned}$$

We can also argue about formulas by induction.

Theorem: The degree of a formula equals the number of inner nodes of the parse tree.

Proof:

Semantics of propositional logic

- In propositional logic, formulas can have one of two possible values **True=T** or **False=F**
- A **truth table** lists each possible combination of truth values for the variables of a Boolean function (=truth function) $f : \{T, F\}^n \rightarrow \{T, F\}$, and specifies the function value in each case.
- n variables; gives a table with 2^n rows
- order of rows is irrelevant

Truth tables of connectives

Each connective is defined by its truth table:

Unary connective

X	$\neg X$
T	F
F	T

Binary connectives

X	Y	$X \wedge Y$	$X \vee Y$	$X \rightarrow Y$	$\ldots$
T	T	T	T	T	
T	F	F	T	F	
F	T	F	T	T	
F	F	F	F	T	

Nullary connectives $\top$="top" and $\bot$="bottom"

	$\top$
	T

	$\bot$
	F

Valuations

A **valuation** is a mapping v from the set of propositional formulas to the set of truth values $\{T, F\}$ satisfying the following conditions:

- $v(\top) = T$; $v(\bot) = F$,
- $v(\neg X) = \neg v(X)$,
- $v(X \circ Y) = v(X) \circ v(Y)$.

To examine all interesting valuations for a given formula, it is enough to specify v for each variable. The value of all subformulas can then be deduced bottom-up (in the parse tree).

Example: if v is a valuation such that $v(p) = T$ and $v(q) = F$,
and $X = \neg p \vee (\neg q \rightarrow p)$
then $v(\neg p) = F$, $v(\neg q) = T$, $v(\neg q \rightarrow p) = T$, and $v(X) = T$

The truth table of a formula lists all possible valuations.

Truth table of compound formulas

Example 1: $(p \wedge q) \vee (\neg p \wedge \neg q)$

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

Truth table of compound formulas

Example 2: $p \to (q \wedge r)$

p	q	r	p	$\to$	$(q \wedge r)$
T	T	T		T	T
T	T	F		F	F
T	F	T		F	F
T	F	F		F	F
F	T	T		T	T
F	T	F		T	F
F	F	T		T	F
F	F	F		T	F

Tautologies, contradictions

A formula that evaluates to T under all valuations is called a **tautology**.

A formula that evaluates to F under all valuations is called a **contradiction**.

A formula that evaluates to T for some valuation is called **satisfiable**.

Propositional consequence

We say that a formula X is a **consequence** of a set S of formulas, denoted $S \models X$, provided that X maps to T under every valuation that maps every member of S to T.

X is a tautology if and only if $\emptyset \models X$, which we write as $\models X$.

Examples:

- $\{p, p \rightarrow q\} \models q$
- $\{p, q, r\} \models p$
- $\models (p \vee \neg p)$

Counterexamples:

- $p \vee q \not\models q$
- $p \rightarrow q \not\models p$

Limitations of truth tables

Truth tables are great, but there are also disadvantages:

- Space requirements: 2^n rows for n variables
- Often wasteful; e.g.: $p \vee q \vee r \vee s \vee t \vee \neg p$ (look at binary decision diagrams)
- Symmetry or structure of formulas is disguised; e.g.: $f(x_1, \ldots, x_n) = T$ iff and only if and odd number of $x_i = T$