# COMP1215 - Logic

Dominik Tarnowski

October 2019

## Contents

# 1 Proposition

**Definition 1.1.** Proposition A statement that we assume to be true

**Example 1.1.** Invalid propositions:

1. Is it raining today?

2. $x^2 = 25$

3. go eat something.

Propositions must be either true or false. Items 1 and 3 aren't propositions, and number 2 can either be true or false.

**Example 1.2.** Valid propositions are either true or false:

1. It's raining today

2. 1+2=3

3. The sun is not yellow

## 1.1 Propositional Variables

Propositional variables are used to to represent prepositions. By convention, we use the following variables as propositional variables:

$$p, q, r, s$$

# 2 Logical Operators

Propositions are combined together using logical operators. These are basically mathematical ways of writing logical statements such as:

1. If it's raining today, I will wear a coat.
2. I will drink water unless I have some lemonade left in the fridge.

## 2.1 Negation ($\neg$, NOT p)

**Definition 2.1.** Negation. Inverts the variable. If $p = T$, then $\neg p = F$ and etc.

The most basic logical operator is negation. It is a **unary** operator, meaning it only takes one variable. We can negate a proposition $p$ by writing:

$$\neg p$$

And here is the truth table for negation:

| $p$ | $\neg p$ |
|---|---|
| F | T |
| T | F |

## 2.2 Conjuction ($p \wedge q$, p AND q)

**Definition 2.2.** Conjunction. Two or more events or things occurring at the same time.

Given propositions $p$ and $q$, the statement:

$$p \wedge q$$

Is only true if both $p$ and $q$ are true.
The truth table for conjunction looks like this:

| $p$ | $q$ | $p \wedge q$ |
|---|---|---|
| F | F | F |
| F | T | F |
| T | F | F |
| T | T | T |

## 2.3 Disjunction ($p \vee q$, p OR q)

**Definition 2.3.** Disjunction. One or both of the things occurring at the same time.

Given propositions $p$ and $q$, the statement:

$$p \lor q$$

Is true if either of the propositions is true.

| $p$ | $q$ | $p \lor q$ |
|---|---|---|
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | T |

### 2.4 Implication ($p \to q$, if p then q)

Implications is slightly harder to understand. Let's consider $p \to q$, read as "p implies q".

1. We assume implication until proven otherwise, so when $p = False$, the implication is true.
2. $p$ implies $q$ if $p$ is true and $q$ is true.

| $p$ | $q$ | $p \to q$ |
|---|---|---|
| F | F | T |
| F | T | T |
| T | F | F |
| T | T | T |

## 3 Falsity / Contradiction ($\bot$)

todo: more
This is a constant symbol.

## 4 Syntactic Order

| Operation | Precedence | Associativity |
|---|---|---|
| $\neg$ | 2 | - |
| $\land$ | 1 | left |
| $\lor$ | 1 | left |
| $\to$ | 0 | right |

**Example 4.1.**

$$p \land \neg q \land r \to s = (((p \land (\neg q)) \land r) \to s)$$

**Example 4.2.**

$$p \rightarrow \neg q \rightarrow r = p \rightarrow ((\neg q) \rightarrow r)$$

# 5 Semantics

**Definition 5.1.** Semantics. Meaning of the formula.

## 5.1 Semantics as a function

$$Let\ 2 = \{False, True\}$$

$$Let\ \neg : 2 \rightarrow 2$$

$$Let\ \vee : 2 \times 2 \rightarrow 2$$

$$Let\ \wedge : 2 \times 2 \rightarrow 2$$

$$Let\ \rightarrow : 2 \times 2 \rightarrow 2$$

Suppose that **V** is the set of propositional variables (p,q,r,s,...), and $\sigma$ is some function that takes a formula and returns its true value:

$$[\![-]\!]_\sigma : F \rightarrow 2$$

This means that anything placed in $\sigma(...)$ function will be evaluated to $True$ or $False$.

**Example 5.1.** Evaluate $[\![\bot]\!]_\sigma$.

$$[\![\bot]\!]_\sigma = F$$

**Example 5.2.** Evaluate $[\![p \wedge \neg q]\!]_\sigma$, given that $\sigma(p) = T, \sigma(q) = F$.

$$[\![p \wedge \neg q]\!]_\sigma = \wedge([\![p]\!]_\sigma, \neg([\![q]\!]_\sigma)) = \wedge(\sigma(p), \sigma(\neg q)) = \wedge(T, T) = T$$

# 6 Tautuology / Valid Formulae

**Definition 6.1.** Tautology. A formula that evaluates to True for all possible values of the propositional variables.

**Example 6.1.** Here is an example of a tautology:

| $p$ & $q$ & $(p \rightarrow q) \vee (q \rightarrow p)$ | heightT | T |
|---|---|---|
| T | | |
| T | F | T |
| F | T | T |
| F | F | T |

### 6.1 Satisfiable Formula

**Definition 6.2.** Satisfiable Formula. A formula that evaluates to True at least once.

### 6.2 Logically Equivalent

**Definition 6.3.** Logically Equivalent. Two formulas are equivalent when they evaluate to same values given the same inputs.

## 7 Formal Proof

1. A collection of rules is used to prove things formally
2. there are many proof systems, however we will be covering Natural Deduction.

### 7.1 Notation

Let's that $\Gamma$ is a set of formulas and $\phi$ is a formula. When we can prove $\phi$ as the conclusion, given formulas in $\Gamma$, we write:

$$\Gamma \vdash \phi$$

When we can prove $\phi$ without any assumptions, we write the following (shorthand for $\emptyset \vdash \phi$):

$$\vdash \phi$$

### 7.2 Difference between Proof and Formal Proof

So far, we proved things by logically describing the process of prooving something. Although for many cases, this is enough to prove it, logicians use formal proof systems in order to prove more complex concepts.

### 7.3 Semantic Entailment

Suppose that $\Gamma$ is the set of formulas, let $\phi$ be a tautology.
If $\Gamma$ semantically entails $\phi$, it means that if all the formulas of $\Gamma$ evaluate to $T$, then $\phi$ also evaluates to $T$. This is denoted with the following syntax:

$$\Gamma \vDash \phi$$

### 7.4 Soundness

Suppose that $\Gamma$ is the set of all formulas, let $\phi$ be a tautology.

$$\Gamma \vdash \phi \ implies \ \Gamma \vDash \phi$$

This means that whenever $\Gamma$ proves $\phi$, $\Gamma$ semantically entails $\phi$.

### 7.5 Completeness

Completeness also refers to the set of formulas and a tautology and is the opposite of soundness:

$$\Gamma \vDash \phi \textit{ implies } \Gamma \vdash \phi$$

# 8 Natural Deduction

Natural deduction is a type of a formal proof system.

## 8.1 Introduction

How does natural deduction work? It's based on syntax trees, but upside down. todo: diagram

## 8.2 Conjunction Rules

### 8.2.1 ∨ Introduction

$$\frac{\phi \qquad \psi}{\phi \vee \psi}$$

The leaf nodes from the top are always the assumptions (from Γ), and the very bottom is the final tautology.

### 8.2.2 ∨ Elimination

Given a conjunction, we can infer/eliminate any one of the variables:

$$\frac{\phi \vee \psi}{\phi}$$

or we can do this:

$$\frac{\phi \vee \psi}{\psi}$$

## 8.3 Implication Rules

### 8.3.1 → Introduction

$$\begin{array}{c} [\phi] \\ \vdots \\ \psi \\ \hline \phi \to \psi \end{array}$$

todo: explain syntax

### 8.3.2 → Elimination

$$\frac{\phi \qquad \phi \to \psi}{\psi}$$

## 8.4 Negation Rules

### 8.4.1 ¬ Introduction

$$\begin{array}{c}[\phi] \\ \vdots \\ \dfrac{\bot}{\neg\phi}\end{array}$$

### 8.4.2 ¬ Elimination

$$\frac{\phi \qquad \neg\phi}{\bot}$$

## 8.5 Disjunction

### 8.5.1 ∧ Introduction

$$\frac{\psi}{\phi \wedge \psi}$$

$$\frac{\phi}{\phi \wedge \psi}$$

## 8.6 ∧ Elimination

$$\dfrac{\begin{array}[b]{c}[\phi] \\ \vdots \\ \chi\end{array} \qquad \begin{array}[b]{c}[\psi] \\ \vdots \\ \chi\end{array} \qquad \phi \wedge \psi}{\chi}$$

## 8.7 Rules for False

$$\frac{\bot}{\phi}$$

As you can see, we can derrive anything from ⊥.

$$\begin{array}{c}[\neg\phi] \\ \vdots \\ \dfrac{\bot}{\phi}\end{array}$$

# 9 Predicate logic

Predicate logic is a much richer language that allows us to do more things. In propositional logic, we would need to use words to describe some things that predicate logic has syntax for, e.g:

$$\text{for all } x_1 \text{ and } x_2 \text{ in } A, \text{ if } f(x_1) = f(x_2) \text{ then } x_1 = x_2$$

Could be written in predicate logic by using the following syntax:

$$\forall x. \forall y. (f(x_1) = f(x_2)) \rightarrow (x_1 = x_2)$$

Where $\forall$ means for all elements in a world.
Where $\exists$ means there exists an element in the world.

## 9.1 Semantics

Given a set $A$, called the domain:

1. for each constant $a$, $a \in A$
2. for each n-ary function $f$, $f : A^n \rightarrow A$
3. for each n-ary relation $R$, $R \subseteq A^n$

Given a function $\sigma$, from variables to elements of $A$:
If $A$ **is a model for** $\phi$, we write:

$$A \vDash_\sigma \phi$$

## 9.2 Godel's incompleteness theorem

Let's start with the basics. In mathematics, we have something called **axioms**. Axioms are **statements that we can assume to be true without having to prove them formally**, for example:

1. if a=b and b=c, then a=c,
2. given any two points, you can always draw a line between them

Godel proved the following statement

> If axioms don't contradict each other and are "listable", some statements are true but cannot be proved

### 9.2.1 But what does that mean?

It means that there will always be some problems that no matter how powerful a computer might be, given a valid set of axioms, it will not be able to decide whether they are true or false. These types of problems are very complicated, hence I did not include an example here.

More importantly, in this context, sometimes it is impossible to get a sound and complete system of axioms that will prove all the statements.

**Example 9.1.** There is no finite set of rules that are both sound and complete for all the true statements about natural numbers.