Discrete Mathematics

Dr Kieran Murphy

Department of Department of Computing and Mathematics
INSTITUTION.
(kmurphy@wit.ie)

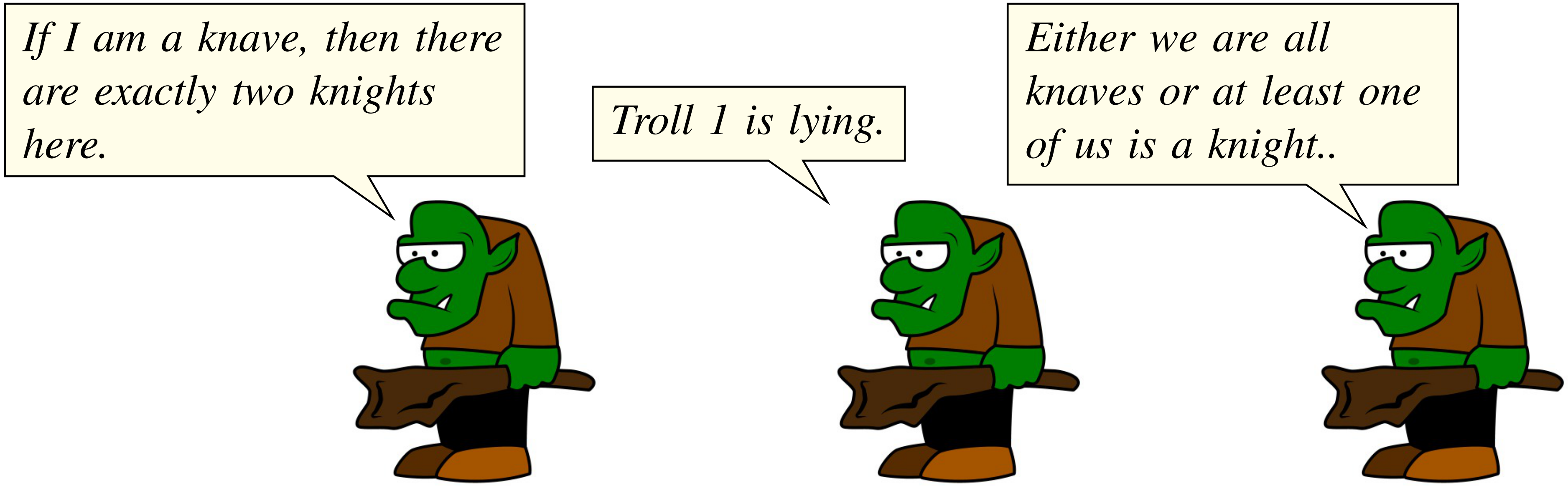
Autumn Semester, 2021

RESOURCE OUTLINE LABEL

- Propositions and fundamental logical operators (AND, OR and NOT).
- Evaluating logical expression using truth tables.
- Satisfiability, Tautologies and Contradictions.

Thought for the day . . .

While walking through a fictional forest, you encounter three identical trolls guarding a bridge. Each troll is either a knight, who always tells the truth, or a knave, who always lies. The trolls will not let you pass until you correctly identify each as either a knight or a knave. Each troll makes a single statement:



Which troll are knights? and which are knaves?

Outline

Logic

Logic is "science of reasoning"

- Allows us to represent knowledge in precise, unambiguous way.
- Allows us to make valid inferences using a set of consistent rules.
- Roots of logic date back to the ancient Greeks, e.g., Aristotle.
- Greeks were interested in valid logical inference rules, such as syllogisms:

  "All men are mortal.
  Socrates is a man.
  Therefore, Socrates is mortal."

The Partially Examined Life podcast: `www.partiallyexaminedlife.com`
The Fallacy-a-Day Podcast: `http://fallacyaday.com`

Propositional Logic I

- The building blocks of propositional logic are propositions

Definition 1 (Proposition)

A proposition (statement) is a sentence that is either **True** or **False**.

- Examples:

"*Java is a programming language.*"	**True**
"*Cork is the capital of Ireland.*"	**False**
"$1 + 2 = 3$"	**True**
"*Today is Tuesday.*"	depends
"*The universe is fine-tuned.*"	unknown (at present)

- Examples of sentences that are not propositions/statements:
  - "*How are you?*" — A question cannot be assign a **True/False** value.
  - "*Stop sleeping in class!*" — An order cannot be assign a **True/False** value.
  - "*Correct horse battery staple.*" — Not a sentence.
  - "*This sentence is false.*" — Pathological example.

Propositional Variables, Truth Value

Given a proposition we are interested in knowing its truth value.

Definition 2 (Truth Value)

The truth value of a proposition identifies whether a proposition is true (written **True** or **T** or 1) or false (written **False** or **F** or 0)

Question

What is truth value of "*Tuesday in the day after Sunday*" ? **F**

Notation

- Variables that represent propositions are called propositional variables.
- Denote propositional variables using lower-case letters, such as p, p_1, p_2, q, r, s, $\ldots$
- Truth value of a propositional variable is either **T** or **F**.

Compound vs Atomic Propositions

- Propositional logic allows constructing more complex propositions from atomic ones.
- More complex propositions formed using logical connectives (also called boolean connectives or logical operators).
- The three basic logical connectives:

Connective	Symbol	Python
conjunction (AND)	$\wedge$	**and**
disjunction (OR)	$\vee$	**or**
negation (NOT)	$\neg$	**not**

- Propositions formed using these logical connectives called compound propositions; otherwise called atomic propositions.

$$\underbrace{\underbrace{\text{Today is wet}}_{\text{atomic}} \text{ and } \underbrace{\text{I am hungry}}_{\text{atomic}}}_{\text{compound}}$$

Exercise

Classify each of the sentences below as an atomic statement, a compound statement, or not a statement at all.

1. The sum of the first 100 odd positive integers.
2. Everybody needs somebody sometime.
3. Waterford will win the All-Ireland or I'll eat my hat.
4. Go to your room!
5. Every natural number greater than 1 is either prime or composite.
6. This sentence is false.

Negation (NOT)

- Negation of a proposition, p, written, $\neg p$, represents the proposition:
  "*It is not the case that p.*"
- What is the relationship between the truth value of p and $\neg p$?

  If p is **T**, then $\neg p$ is **F** and vice versa.

- In simple English, what is $\neg p$ if p stands for ...

p	$\neg p$
"*Today is Tuesday.*"	"*Today is not Tuesday.*"
"$1 + 1 = 2$"	"$1 + 1 \neq 2$"

- Properties of NOT
  - $\neg\neg p = p$

Conjunction (AND)

- Conjunction of two propositions, p and q, written as $p \wedge q$, is the proposition:

  "*p and q*"

- What is the relationship between the truth value of p and of q and the truth value of $p \wedge q$?

$$p \wedge q = \begin{cases} \mathbf{T} & \text{if both } p \text{ is } \mathbf{T} \text{ and } q \text{ is } \mathbf{T} \\ \mathbf{F} & \text{otherwise} \end{cases}$$

Example

What is the conjunction and the truth value of $p \wedge q$ for ...

- $p =$ "*It is a autumn semester*", $q =$ "*Today is Thursday*"
- $p =$ "*It is Tuesday*", $q =$ "*It is morning*"

Disjunction (OR)

- Disjunction of two propositions, p and q, written as $p \vee q$, is the proposition

$$\text{“}\textit{p or q}\text{”}$$

- What is the relationship between the truth value of p and of q and the truth value of $p \vee q$?

$$p \vee q = \begin{cases} \mathbf{T} & \text{if either } p \text{ is } \mathbf{T} \text{ or } q \text{ is } \mathbf{T}\text{, or both are } \mathbf{T} \\ \mathbf{F} & \text{otherwise} \end{cases}$$

Example

What is the disjunction and the truth value of $p \vee q$ for …

- $p =$ “*It is a autumn semester*”, $q =$ “*Today is Thursday*”
- $p =$ “*It is Friday*”, $q =$ “*It is morning*”

Python Implementation I

Python supports the fundamental logical connectives (programmers call them “logical operators”)

Logical Connective	Math	Python Operator
conjunction (AND)	$\wedge$	**and**
disjunction (OR)	$\vee$	**or**
negation (NOT)	$\neg$	**not**

Outline

Propositional Formulas and Truth Tables

- A propositional formula is logical expression constructed from atomic and compound propositions and logical connectives.
- A truth table for a propositional formula, A, shows the truth value of A for every possible value of its constituent atomic propositions.

Negation

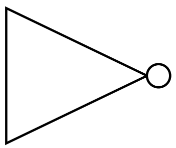
p	$\neg p$
F	**T**
T	**F**

Conjunction

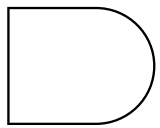
p	q	$p \wedge q$
F	**F**	**F**
F	**T**	**F**
T	**F**	**F**
T	**T**	**T**

Disjunction

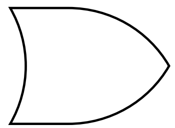
p	q	$p \vee q$
F	**F**	**F**
F	**T**	**T**
T	**F**	**T**
T	**T**	**T**



NOT



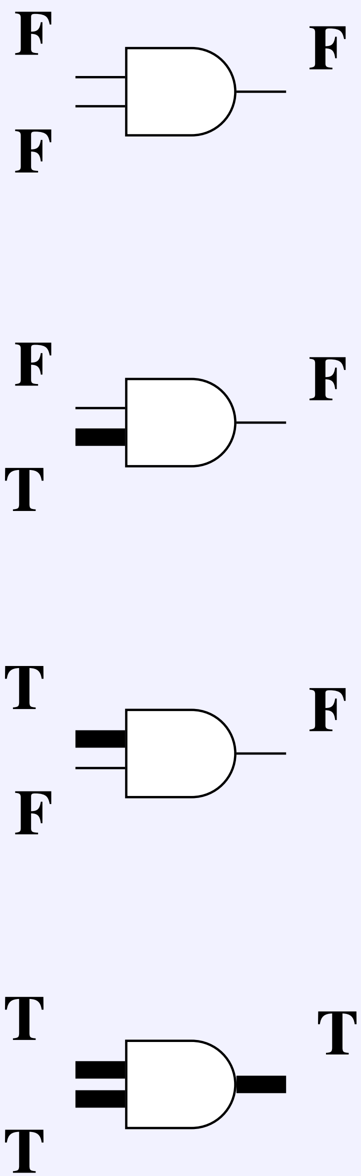
AND



OR

Truth tables and Logic Gates

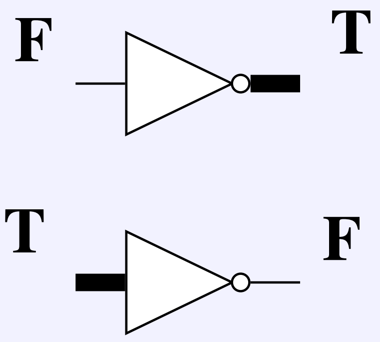
AND



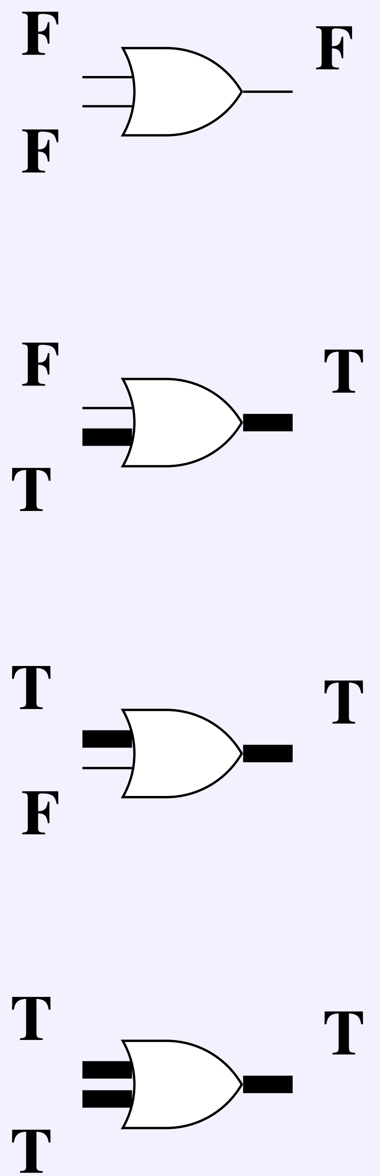
p	q	$p \wedge q$
F	**F**	**F**
F	**T**	**F**
T	**F**	**F**
T	**T**	**T**

NOT

p	$\neg p$
F	**T**
T	**F**



OR

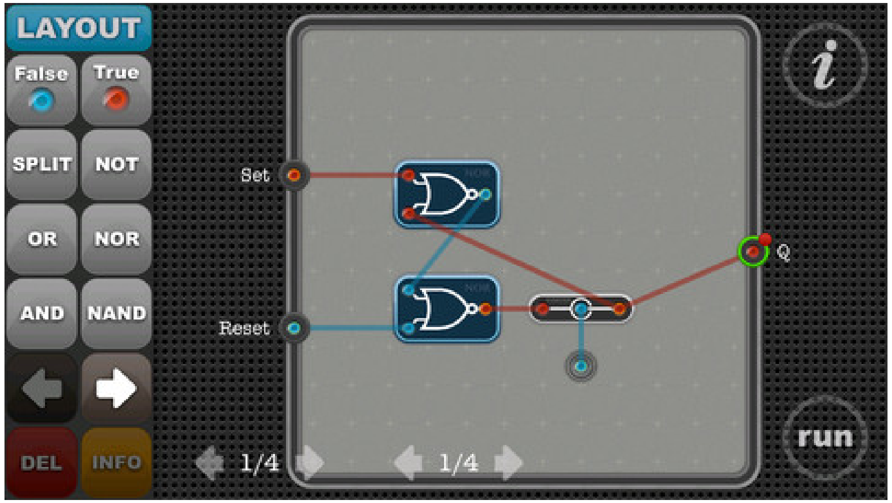


p	q	$p \vee q$
F	**F**	**F**
F	**T**	**T**
T	**F**	**T**
T	**T**	**T**

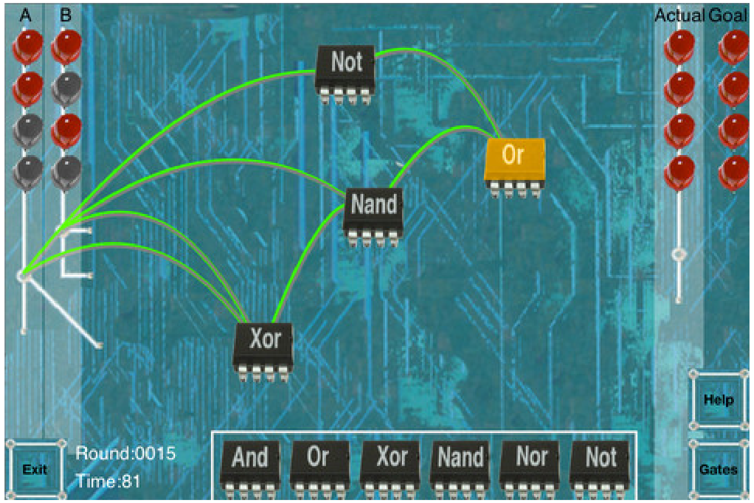
Other Resources

iPad/iPhone Apps (assume similar on Android)

Circuit Coder



Boolean Master



Videos

- `https://class.coursera.org/cs101/lecture/17`
  Part of the Computer Science 101 by Nick Parlante on coursera.

Constructing Truth Tables

Useful strategy for constructing truth tables for a formula:

STEP 1 Identify the constituent atomic propositions of the formula.

STEP 2 Identify compound propositions in within the formula in increasing order of complexity, including the formula itself.

STEP 3 Construct a table enumerating all combinations of truth values for atomic propositions.

STEP 4 Fill in values of compound propositions for each row.

Examples

Construct truth tables for the following formulas:

1. $(p \vee q) \wedge \neg p$
2. $(p \wedge q) \vee (\neg p \wedge \neg q)$
3. $(p \vee q \vee \neg r) \wedge r$

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

STEP 1 Identify the constituent atomic propositions … p and q

STEP 2 Identify compound propositions …

STEP 3 Enumerate all combinations of truth values for atomic propositions …

STEP 4 Fill in values of compound propositions for each row …

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

Example 2: $(p \wedge q) \vee (\neg p \wedge \neg q)$

STEP 1 Identify the constituent atomic propositions … p and q

STEP 2 Identify compound propositions …

STEP 3 Enumerate all combinations of truth values for atomic propositions …

STEP 4 Fill in values of compound propositions for each row …

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

Example 3: $(p \vee q \vee \neg r) \wedge r$

- STEP 1 Identify the constituent atomic propositions … p, q, and r
- STEP 2 Identify compound propositions …
- STEP 3 Enumerate all combinations of truth values for atomic propositions …
- STEP 4 Fill in values of compound propositions for each row …

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

Outline

Introduction to Propositional Logic — Summary

1. Introduction 3
   - Propositional logic is concerned with analysing propositions (true or false statements).
   - A proposition may be atomic or compound (build up using logical connectives).
   - Constructing compound propositions using *And*, *Or* and *Not*.

2. Truth tables 13
   - Evaluating an expression for all possible input combinations.

3. Tautologies and Contradictions 21
   - Statements that are always true or always false.

Satisfiable, Tautologies and Contradictions

Satisfiable

A proposition is satisfiable if it is **True** for at least one set of inputs (case).

Tautology

A tautology is an expression involving logical variables that is **True** in all cases.

- Examples
  - $p \vee \neg p$ "*Tomorrow, I will be dead or I will be alive*"
  - $(p \wedge q) \vee (p \wedge \neg q) \vee \neg p$

Contradiction

A contradiction is an expression involving logical variables that is **False** in all cases.

- Examples
  - $p \wedge \neg p$ "*On Friday, I will win the lottery and not win the lottery.*"