Propositional Logic Part1

Dr. Igor Ivkovic

iivkovic@uwaterloo.ca

[with material from “Mathematical Logic for Computer Science”, by Zhongwan, published by World Scientific]

Objectives

- Propositions and Connectives
- Propositional Language
- Propositional Formulas

Introduction

- **Proposition:**
  - A statement that is either **true** or **false**
  - The values of any proposition however are **truth** and **falsehood**
- For any proposition A:
  - The proposition "A or not A" is true
- In propositional logic, **simple propositions** are the basic building blocks used to create **compound propositions** using **connectives**
- **Propositional logic analyzes the compound statements and their composition**
  - It does not analyze the simple propositions, which are taken as either true or false

Propositions and Connectives /1

- Connectives:
  - Used to form compound propositions
  - Commonly used connectives are **”not”**, **”and”**, **”or”**, **”if then”**, and **”iff”**
  - All are binary except for "not" which is unary (i.e., operates on one proposition)

- Some examples of propositions:
  1. 3 is not even (**not** "3 is even")
  2. 4 is even **and** not prime
  3. **If** "x is greater than 2" **and** "x is prime" **then** "x is not 4“
  4. Paul is taller than Mike **iff** Mike is shorter than Paul

Propositions and Connectives /2

- For two propositions A and B, the following are formed using common connectives:
  - **Not A**
  - **A and B**
  - **A or B**
  - **If A then B**
  - **A iff B**
- **For “Not A”:**

A	Not A
0	1
1	0

Propositions and Connectives /3

- **For “A and B”:**

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

- **For “A or B”:**

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

Propositions and Connectives /4

- **For “if A then B”:**

A	B	If A then B
0	0	1
0	1	1
1	0	0
1	1	1

- **For “A iff B”:**

A	B	A iff B
0	0	1
0	1	0
1	0	0
1	1	1

Propositional Language /1

- **The Propositional Language L^{p} :**
  - The formal language of the propositional logic consists of **the proposition symbols**, **five connectives**, and **two punctuation symbols**
  - The proposition symbols are denoted with small Latin letters such as p, q, and r (no default ordering)
  - The five connectives are $\neg$ **(not / negation),** $\wedge$ **(and / conjunction),** $\vee$ **(or / disjunction),** $\Rightarrow$ **(if-then / implication), and** $\Leftrightarrow$ **(iff / equivalence)**
  - The two punctuation symbols are “(“ and “)”; that is, the left and right parentheses
  - **Expressions** are finites strings of symbols and **the length of an expression** is the number of symbols in it

Propositional Language /2

- **Properties of L^{p} :**
  - Empty expression (of length 0) denoted with $\varnothing$
  - Two expressions U and V are equal, written as U = V, iff they are of the same length with same symbols in order
  - UV is the concatenation of two expressions U and V
  - If U = W1VW2 then V is a **segment** of U; if U ≠ V then V is a proper segment of U
  - If U = VW where V is an **initial segment** of U and W is a **terminal segment** of U
  - If V is non-empty then W is a **proper terminal segment** and if W is non-empty then V is a **proper initial segment**
  - **Atoms** (or atomic formulas) and **Formulas** are defined from expressions

Propositional Language /3

- **Definition 1. Atom(L^{P}):** (Definition 2.2.1 from textbook)
  - The set of expressions of L^{P} that consists of propositions symbols only
  - p, q, r… $\in$ Atom(L^{P}); but (p) $\notin$ Atom(L^{P})

Referred to as Well-Formed Formula (WFF)

- **Definition 2. Form(L^{P}):** (Definition 2.2.2)
  - An expression of L^{P} is a formula of L^{P} iff it can be generated using the following (formation) rules:
    [1] Atom(L^{P}) $\subseteq$ Form(L^{P}),
    [2] If A $\in$ Form(L^{P}) then ($\neg$A) $\in$ Form(L^{P})
    [3] If A, B $\in$ Form(L^{P}) then (A * B) $\in$ Form(L^{P}), where * stands for any of the five connectives in L^{P}
  - [1], [2], and [3] are the formation rules of formulas of L^{P}

Propositional Language /4

- **Definition 3. Closure of Form($\mathcal{L}^p$):** (Definition 2.2.3)
  - Form($\mathcal{L}^p$) is the smallest class of expression of $\mathcal{L}^p$ closed under the formation rules of $\mathcal{L}^p$
- Applying the formulas:
  - Let us generate several expression using the formation rules to prove that these are indeed formulas of $\mathcal{L}^p$
  - $(q \vee p)$
  - $(\neg q)$
  - $(p \wedge r)$
  - $((\neg q) \Leftrightarrow (p \wedge r))$
  - $((q \vee p) \Rightarrow ((\neg q) \Leftrightarrow (p \wedge r)))$
- We use roman capital letters to indicate formulas, such as A, B, C, …

Propositional Formulas /1

- **Lemma 1:** (Lemma 2.3.1)
  - Every formula L^p has the same number of left and right parentheses
- **Lemma 2:** (Lemma 2.3.2)
  - Any non-empty proper initial segment of a formula of L^p has more left than right parentheses, and any non-empty proper terminal segment of a formula of L^p has less left than right parentheses
- **Theorem 1. Formula Uniqueness:** (Theorem 2.3.3)
  - Every formula of L^p is of exactly one of the six forms: **an atom, $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \Rightarrow B)$, and $(A \Leftrightarrow B)$;** and in each case it is of that form in exactly one way

Propositional Formulas /2

- Based on the above theorem:
  - The generation of a formula is unique given that the ordering of certain steps is not considered
- **Definition 4. Formula Types:** (Definition 2.3.4)
  - $(\neg A)$ is called a negation (formula)
  - $(A \wedge B)$ is called a conjunction (formula)
  - $(A \vee B)$ is called a disjunction (formula)
  - $(A \Rightarrow B)$ is called an implication (formula)
  - $(A \Leftrightarrow B)$ is called an equivalence (formula)

Propositional Formulas /3

- **Definition 5. Formula Scope:** (Definition 2.3.5)
  - If ($\neg$A) is a segment of C then A is called the scope in C of the $\neg$ on the left of A
  - If (A * B) is a segment of C then A and B are called the left and right scopes in C of the * between A and B
- **Theorem 2. Scope Uniqueness:** (Theorem 2.3.6)
  - Any $\neg$ in any A has a unique scope, and any * in any A has unique left and right scopes
- **Theorem 3. Segment Scope:** (Theorem 2.3.7)

  [1] If A is a segment of ($\neg$B) then A is a segment of B or A = ($\neg$B)

  [2] If A is a segment of (B * C) then A is a segment of B, or A is a segment of C, or A = (B * C)

Propositional Formulas /4

- **Algorithm 1. Verify Expression as a Formula:**
  - **Input:** U is an expression of L^{P}
  - **Output:** true if U is a formula of L^{P}; false otherwise
  - **Steps:**

(1) If U is empty, empty expression is not a formula so return false

(2) If U is a single propositional symbol then U is a formula so return true; otherwise if U is any other single symbol, return false

(3) If U contains more than one symbol, it must start with the left parenthesis; otherwise return false

(4) If the second symbol is $\neg$, U must be $(\neg V)$ where V is an expression; otherwise return false. Now, recursively apply the same algorithm to V, which is of smaller size

Propositional Formulas /5

- **Algorithm 1. Continued…**
  (4) …
  (5) If U begins with a left parenthesis but the second symbol is not $\neg$, scan from left to right until (V segment is found where V is a proper expression; if no such V is found, return false. U must be (V * W) where W is also an expression; otherwise return false.
  (6) Now apply the same algorithm recursively to V and W
- Termination:
  - Since every expression is finite in length by definition, and since in each iteration the analyzed expressions are getting smaller, **the algorithm terminates in a finite number of steps**

Propositional Formulas /6

- **Discussion:**
  - Parentheses, even though included in the L^p definition, can be omitted
  - There is an ordering of propositional connectives, similar to the order of algebraic symbols +, -, *, \
  - That is, the following is the order of precedence (from highest to lowest) of the propositional connectives:

    (1) $\neg$

    (2) $\wedge$

    (3) $\vee$

    (4) $\Rightarrow$

    (5) $\Leftrightarrow$

Food for Thought

- **Read:**
  - Chapter 2, Sections 2.1, 2.2, and 2.3 from Zhongwan
    - Read proofs presented in class in more detail
    - Cursory reading of proofs omitted but mentioned in class
- Answer the following exercises: (short answers)
  - Exercises 2.2.1 and 2.2.2
  - Exercises 2.3.1 and 2.3.2
- (Optional) Read:
  - Chapters 2 and 3, Sections 3.1 and 3.2 from Nissanke
    - Complete at least a few exercises from each section