

SE212

Andy Zhang

Fall 2014

# Contents

<b>1</b>	<b>Module 2 - Propositional Logic</b>	<b>2</b>
1.1	Definitions . . . . .	2
<b>2</b>	<b>Module 3 - Predicate Logic</b>	<b>5</b>
2.1	Definitions . . . . .	5
<b>3</b>	<b>Module 4 - Theories</b>	<b>7</b>
3.1	Definitions . . . . .	7
<b>4</b>	<b>Module 5 - Sets</b>	<b>9</b>
4.1	Definitions . . . . .	9

# Chapter 1

## Module 2 - Propositional Logic

### 1.1 Definitions

**Proof Theory** are methods that manipulate strings of symbols base on pattern matching. There may be multiple ways to prove a formula.

**Sound** If  $p_1, p_2, \dots, p_n \models q$  (proof), then  $p_1, p_2, \dots, p_n \vdash q$  (valid)

**Complete** If  $p_1, p_2, \dots, p_n \vdash q$  (valid) then  $p_1, p_2, \dots, p_n \models q$  (proof)

**Boolean Valuation** is a function  $v$  from the set of formulas in propositional logic to the set  $T_r$ . Boolean valuation is also called a model or an interpretation.

- $v(false) = F, v(true) = T$
- $v(\neg p) = NOT(v(p))$
- For the connectives:
  - $v(p \wedge q) = v(p) AND v(q)$
  - $v(p \vee q) = v(p) OR v(q)$
  - $v(p \Rightarrow q) = v(p) IMP v(q)$
  - $v(p \Leftrightarrow q) = v(p) IFF v(q)$

**Satisfiability** If a formula is **satisfiable**, then there exists a Boolean valuation  $v$  such that  $v(p) = T$

**Tautologies** A propositional formula  $p$  is a **tautology** or **valid** if  $v(p) = T$  for all Boolean valuations  $v$ . When a formula  $q$  is a tautology, we write

$$\vdash q$$

**Logical Implication** A formula  $p$  **logically implies** a formula  $q$  iff for all Boolean valuations  $v$ , if for all premises  $v(p_i) = T$ , then  $v(q) = T$ , meaning

$$p \vdash q \text{ which is equivalent to } \vdash p \Rightarrow q$$

**Contradiction** A propositional formula  $a$  is a **contradiction** if  $v(a) = F$  for all Boolean valuations  $v$

**Contingent** A **contingent** is one that is neither a **tautology** nor a **contradiction**

**Logical Equivalence** Two formulas are **logically equivalent** iff their equivalence is a tautology i.e.  $v(p) = v(q)$  for all  $v$ .

$p \leftrightarrow q$  which also means  $\vdash p \leftrightarrow q$

$\leftrightarrow$  Logical equivalence

$\Leftrightarrow$  Material equivalence

**Consistency** A collection of formulas is **consistent** if there exists a boolean valuation where all the formulas can be true simultaneously. If a set of formulas in the antecedent of an implication, they can be used to prove a contradiction.

### Transformational Proof Rules

Comm  $p \wedge q \Leftrightarrow q \wedge p$

Lem  $p \vee \neg \Leftrightarrow \text{true}$

Contr  $p \wedge \neg p \Leftrightarrow \text{false}$

Impl  $p \Rightarrow q \Leftrightarrow \neg p \vee q$

Idemp  $p \wedge p \Leftrightarrow p$

Neg  $\neg(\neg p) \Leftrightarrow p$

Simp1  $p \wedge \text{true} \Leftrightarrow p$

Assoc  $p \wedge (q \wedge r) \Leftrightarrow (p \wedge q) \wedge r$

Dm  $\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$

Distr  $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$

Contrapos  $p \Rightarrow q \Leftrightarrow \neg q \Rightarrow \neg p$

Equiv  $p \leftrightarrow q \Leftrightarrow (p \Rightarrow q) \wedge (q \Rightarrow p)$

Simp2  $p \vee (p \wedge q) \Leftrightarrow p$

### Rules

1. **Rule of substitution** substituting an equivalent for a subformula
2. **Rule of transitivity** If  $p \Leftrightarrow q$  and  $q \Leftrightarrow r$ , then  $p \Leftrightarrow r$

**Literal** is a propositional letter or the negation of a proposition letter

**Conjunctive Normal Form** is a conjunction (AND) of clauses, where a clause is a disjunction (OR) of literal or a single literal

**Disjunctive Normal Form** is a disjunction (OR) of clauses, where a clause is a conjunction (AND) of literals or a single literal

**Argument** is a collection of formulas, one of which referred to as the conclusion, is justified by the others, referred to as the premises.

**Deductive** means the conclusion of an argument is wholly justified by the premises

**Inductive** arguments conclude more general new knowledge from a small number of particular facts or observations.

**Valid** means all Boolean valuations where the premises have the value T, the conclusion has the truth value T

**Invalid** means there is one Boolean valuation where premises is true but the conclusion is false

**Natural Deduction** is a collection of rules called inference rules which allows us to infer new formulas from given formulas.

**Inference rule** is a primitive valid argument form. Each inference rule enables the elimination or the introduction of a logical connective

**Syllogism** is a kind of logical argument that applies deductive reasoning to arrive at a conclusion based on two or more propositions that are asserted or assumed to be true.

### Summary of Natural Deduction Rules

- $p, q$  (and i)  $p \wedge q$
- $pq$  (and e)  $p$
- $p$  (or i)  $p \vee q$
- $p \vee r$  (cases)  $\text{case } p \{ \dots q \} \text{ case } r \{ \dots q \} q$
- assume  $r$  { ...  $q$  } (imp i)  $r \Rightarrow q$
- $p \Rightarrow q, p$  (imp e)  $q$
- disprove  $r$  { ... false } (raa)  $\neg r$
- $p, \neg p$  (not e)  $q$
- $p$  (not not i)  $\neg \neg p$
- $\neg \neg p$  (not not e)  $p$
- $p \Rightarrow q, q \Rightarrow p$  (iff i)  $p \Leftrightarrow q$
- $p \Leftrightarrow q$  (iff e)  $p \Rightarrow q$
- $p \vee q, \neg p$  (or e)  $q$
- (lem)  $p \vee \neg p$

### Summary of Semantic Tableaux Rules

- $p \wedge q$  (and nb)  $p, q$
- $\neg(p \wedge q)$  (not and br) 1.  $\neg p$  2.  $\neg q$
- $p \vee q$  (or br) 1.  $p$  2.  $q$
- $\neg(p \vee q)$  (not or nb)  $\neg p, \neg q$
- $p \Rightarrow$  (imp br) 1.  $\neg p$  2.  $q$
- $\neg(p \Rightarrow q)$  (not imp nb)  $p, \neg q$
- $\neg \neg p$  (not not nb)  $p$
- $p \Leftrightarrow q$  (iff br) 1.  $p \wedge q$  2.  $\neg p \wedge \neg q$
- $\neg(p \Leftrightarrow q)$  (not iff br) 1.  $p \wedge \neg q$  2.  $\neg p \wedge q$

# Chapter 2

## Module 3 - Predicate Logic

### 2.1 Definitions

**Predicate** is a symbol denoting the meaning of an attribute of an object or the meaning of a relationship between two or more objects

**Unary, binary, N ary predicate** takes 1, 2 or n number of objects as arguments

**Constant** is a symbol denoting a particular object

#### Quantifiers

forall  $\forall$ , predicate holds for all elements (every, all, for all). The formula must be true for all substitutions of an object in the domain for the quantified variable

exists  $\exists$ , predicate holds for some element (at least one, some, there exists). For an existentially quantified variable, the formula must be true for some substitution of an object in the domain for the quantified variable

#### Terms

- Every constant is a term
- Every variable is a term
- If  $t_1, t_2, t_3, \dots, t_n$  are terms and  $f$  is an  $n$  ary function symbol, then  $f(t_1, t_2, \dots, t_n)$
- Nothing is a term
- Terms describe objects

**Scope of a quantifier** is a subformula over which the quantifier applies in the given formula. Without brackets, we assume the scope of the quantifier extends to the right end of the formula

**Bound Variable** is an occurrence of a variable that is within the scope of a quantifier over that variable. A variable is bound to the closest quantifier to the left of its name whose scope it is within. All variable occurrences that are bound to the same quantifier represent the same object

**Free** , a variable is free if it does not fall within the scope of a quantifier for that variable

**Closed** A wff is closed if it contains no free variables (we work with closed wff)

**Type** is a set of objects describing the possible values of a variable. These values are constant of that type. We assume types are non empty, i.e. at least one element of any type exists. We won't need types unless explicitly instructed.

**Satisfiability** A predicate logic is satisfiable iff there exists an interpretation I that satisfies the formula. An interpretation I satisfies a predicate logic formula A iff  $M[[A]] = T$  when evaluated I

**Tautology** A predicate logic formula is a tautology iff every interpretation satisfies the formula,  $\vdash P$

**Variable capture** means a free variable becomes bound after the substitution

**Genuine variable** is a free variable such that the universal quantification of it yields a formula that is true. It represents any value

**Unknown variable** is a free variable such that the existential quantification of it yields a formula that is true. It represents a specific (but unknown) value

### Summary of Natural Deduction for Predicate Logic

- for every  $x_g \{ \dots P[x_g/x] \}$  (forall i)  $\forall x \bullet P$
- $\forall x \bullet P$  (forall e)  $P[t/x]$
- $P[t/x]$  (exists i)  $\exists x \bullet P$
- $\exists x \bullet P$ , for some  $x_u$   $P[x_u/x] \{ \dots Q \}$  (exists e) Q,  $x_u$  does not appear in Q

### Summary of Semantic Tableau for Predicate Logic

- $\forall x \bullet P(x)$  (forall nb)  $P(t)$ , where  $t$  is a term that is a legal substitution
- $\neg(\forall x \bullet P(x))$  (not forall nb)  $\exists x \bullet \neg P(x)$
- $\exists x \bullet P(x)$  (exists nb)  $P(y)$ , where  $y$  is a variable that has not been used in the tableau so far
- $\neg(\exists x \bullet P(x))$  (not exists nb)  $\forall x \bullet \neg P(x)$

# Chapter 3

## Module 4 - Theories

### 3.1 Definitions

**Enthymeme** is an argument that contains a hidden premise, that is, an argument that contains unstated premises that are obviously true

#### Theory

- A set of statements or principles devised to explain a group of facts or phenomena, especially one that has been repeatedly tested or is widely accepted and can be used to make predictions about natural phenomena
- A set of theorems that constitute a systematic view of a branch of mathematics

**Leibniz Law** if  $t_1 == t_2$  is a theorem, then so is  $P[t_1/x] \Leftrightarrow P[t_2/x]$

Leibniz Law is generally referred to as the ability to substitute equals for equals

**Normal interpretations** are interpretations in which the symbol  $==$  is interpreted as equality on the objects of the domain.

#### Induction

- $P(0)$  is called the base case
- $P(k)$  is called the induction hypothesis
- $\forall k \wedge P(k) \Rightarrow P(suc(k))$  is called the induction proof or inductive step

**Deduction** is showing a conclusion follows from the stated premises using rules of inference

**Philosophical induction** is the process of deriving general principles from particular observations

**Recursive function** is one that is defined in terms of itself and certain terminating clauses

**Satisfiable Modulo Theories** is a decision problem for logical formulas with respect to combinations of background theories expressed in classical first order logic with equality.



**Rule of Thumb** when formalizing a set of sentences.

1. Identify all possible logical connectives. Recall the heuristic from propositional logic: pick the smallest phrases without and or if then etc.
2. Identify what might be quantified, implicit quantifiers, constants
3. Identify the possible predicates and functions. The quantified variables must be arguments to the predicates
4. Look for similar phrases to determine how much detail you need to provide
5. Think of an outline of how your proof will proceed.

**Rule of Thumb** for types

1. If a unary predicate is used in all premises and conclusions as part of the antecedent of a universally quantified formula, it is a good candidate to make a type
2. If a unary predicate is used in all premises as one conjunct of an existentially quantified formula, it is a good candidate to make a type

# Chapter 4

## Module 5 - Sets

### 4.1 Definitions

**Set** is a collection of element or members

**Set enumeration** List the item in the set

**Set comprehension** Define a set by using a predicate

**Z notation**  $\{\langle term \rangle \bullet \langle signature \rangle | \langle predicate \rangle\}$

**Term** is a term in predicate logic i.e an expression using variables and functions that returns an object. Term can be omitted if it is just a variable and we have a signature

**Signature** lists the variables used in term and their types. If we are not using types, the signature can be omitted, but we need one of the term or the signature

**Predicate** is any wff formula in predicate logic with the variables used in term as free variables in the formula. If no predicate is needed, i.e the formula true would be used, then it can be omitted

**Empty set** is written  $\emptyset$

**Universal Set** consists of all the objects of concern in any discussion. Often, the universal set is an appropriate choice for a type. It is denoted as  $U$

**Single set** is a set consisting of only one element

**Power set** of a set is the set of all of its subsets. P is the function that returns the power set of a set. Usually we leave off the brackets around the argument of this function.

## Set Theory Summary

- Types as sets:  $\text{forall } x : B \bullet P(x) \Leftrightarrow \text{forall } x \bullet x \in B \Rightarrow P(x)$
- Set comprehension:  $x \in \{y \bullet y : S \mid P(y)\} \Leftrightarrow x \in S \wedge P(x)$
- Empty set:  $\forall x \bullet \neg(x \in \emptyset)$
- Set equality:  $D == B \Leftrightarrow (\forall x \bullet x \in D \Leftrightarrow x \in B)$
- Subset:  $D \subseteq B \Leftrightarrow (\forall x \bullet x \in D \Rightarrow x \in B)$
- Proper Subset:  $D \subset B \Leftrightarrow D \subseteq B \wedge \neg(D == B)$
- Power Set:  $PD == \{B \mid B \subseteq D\}$
- Set Union:  $D \cup B == \{x \mid x \in D \vee x \in B\}$
- Set intersection:  $D \cap B == \{x \mid x \in D \wedge x \in B\}$
- Absolute Complement:  $D' == \{x \mid x \in U \wedge \neg(x \in D)\}$
- Set difference:  $D - B == \{x \mid x \in D \wedge \neg(x \in B)\}$

**Disjoint** Two sets D and B are disjoint if their intersection is empty