Special Topics on Applied Mathematical Logic

Spring 2012

Lecture 02

Jie-Hong Roland Jiang

National Taiwan University

March 2, 2012

Outline

Sentential Logic
- Building Elements
- Well-Formed Formulas
- Truth Assignments
- Formulas and Boolean Functions
- Compactness
- Effectiveness and Computability

Sentential Logic

- *Sentential logic* is also known as *propositional logic*
- Sentential logic deals with "sentences" in the viewpoint of first-order logic
  - A sentence in first-order logic is abstracted as a sentence symbol in propositional logic
- Sentential logic is used to model propositional statements in natural languages

Use of Sentential Logic in Natural Languages

Consider the double-slit experiment of quantum mechanics with the following events

A1: There is no detector behind both slits

A2: Electron detected at Slit 1

A3: Electron pass Slit 1

A4: Electron pass Slit 2

Example formulas:

$$A_1 \Rightarrow \neg A_2 \tag{1}$$

$$A_2 \Rightarrow \neg A_1 \tag{2}$$

$$A_2 \Rightarrow A_3 \tag{3}$$

$$A_1 \Rightarrow A_3 \tag{4}$$

$$A_2 \wedge A_3 \tag{5}$$

$$A_1 \Rightarrow (A_3 \wedge A_4) \tag{6}$$

Building Elements of Sentential Logic

symbol	meaning
(	left parenthesis for punctuation
)	right parenthesis for punctuation
$\neg$	negation
$\wedge$, $\cdot$	conjunction
$\vee$, $+$	disjunction
$\Rightarrow$	implies
$\Leftrightarrow$, $\equiv$, $\overline{\oplus}$	iff
A_1, A	sentence/propositional symbols (Boolean variables)
A_2, A'	sentence/propositional symbols (Boolean variables)
$\vdots$	$\vdots$

- Logical symbols: (,), $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$
  - Sentential connectives: $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$
- Nonlogical symbols (parameters): $A_1, A_2, \ldots$

Well-Formed Formulas

- A **well-formed formula** (wff) φ is a "grammatically correct" expression
- An operational (recursive) definition of a wff φ is as follows

  $$\varphi := A_i \mid (\neg\varphi_1) \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid (\varphi_1 \Rightarrow \varphi_2) \mid (\varphi_1 \Leftrightarrow \varphi_2)$$

  where ":=" is read as "can be", "|" is read as "or", A_i is some sentence symbol, φ_1 and φ_2 are wffs.
- A wff is an expression that can be built up from the sentence symbols by applying some *finite* number of times the formula-building operations

  $$\begin{aligned} \mathcal{E}_\neg(\alpha) &= (\neg\alpha), \text{ and} \\ \mathcal{E}_\Box(\alpha, \beta) &= (\alpha\Box\beta) \end{aligned}$$

  for $\Box = \wedge, \vee, \Rightarrow, \Leftrightarrow$
  Mind these parentheses!

Ancestral Trees

- Formula construction can be shown with an ancestral tree
  E.g., $(((A_1 \vee A_2) \Rightarrow A_3) \Leftrightarrow (\neg(A_4 \wedge (\neg A_3))))$

$((A_1 \vee A_2) \Rightarrow A_3) \qquad (\neg(A_4 \wedge (\neg A_3)))$

$(A_1 \vee A_2) \qquad A_3 \qquad (A_4 \wedge (\neg A_3))$

$A_1 \qquad A_2 \qquad A_4 \qquad (\neg A_3)$

A_3

Properties of Wffs

The following properties can be shown by induction

- The construction tree of any wff is unique
- If S is a set of wffs containing all sentence symbols and closed under the formula-building operations, then S is the set of all wffs
- Any expression with more left parentheses than right ones is not a wff

Formula Simplification and Polish Notation

To save on parentheses, we may use Polish notation (wffs $\to$ P-wffs)

- $(\alpha \wedge \beta)$ becomes $\wedge\alpha\beta$
- $\mathcal{E}_\neg(\alpha) = (\neg\alpha)$ becomes $\mathcal{D}_\neg = \neg\alpha$
- $\mathcal{E}_\Box(\alpha, \beta) = (\alpha\Box\beta)$ becomes $\mathcal{D}_\Box(\alpha, \beta) = \Box\alpha\beta$ for $\Box \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$
  E.g., $\Leftrightarrow\Rightarrow \wedge AB\neg C \vee \neg DE$

Besides Polish notation, an alternative simplification is to apply the following rules *in order*:

1. omit outermost parentheses
2. $\neg$ applies to as little as possible
3. $\wedge$ applies to as little as possible
4. $\vee$ applies to as little as possible
5. for a repeated connective symbol, grouping is to the right, e.g., $A \Rightarrow B \Rightarrow C \Rightarrow D$ is read as $A \Rightarrow (B \Rightarrow (C \Rightarrow D))$

Syntax vs. Semantics

Back to our example of double-slit experiment

- $(A_1 \Rightarrow (\neg A_2))$:
  "grammatically" or "syntactically" correct (i.e., a wff);
  "physically" or "semantically" correct
- $(A_2 \wedge A_1)$:
  "grammatically" correct; "physically" incorrect

$\left\{\begin{array}{l}\text{syntax — depends only on } \textit{expressions} \\ \text{semantics — depends on } \textit{interpretations} \text{ or } \textit{truth assignments}\end{array}\right.$

Truth Assignments

- Let $\{F, T\}$ be the set of **truth values** with F being the **falsity** and T being the **truth**
- A **truth assignment** is a function $v : S \to \{F, T\}$ assigning either F or T to each sentence symbol in S
- To study the truth or falsity of a wff under some truth assignment, we extend v to $\overline{v} : \overline{S} \to \{F, T\}$, where $\overline{S}$ is the set of wffs that can be built from S by formula-building operations

Truth Assignments

Define $\overline{v}$ as follows

case 0 For $A \in S$, $\overline{v}(A) = v(A)$

case 1 For $\overline{v}((\neg\alpha)) = \begin{cases} T & \text{if } \overline{v}(\alpha) = F \\ F & \text{otherwise} \end{cases}$

case 2 For $\overline{v}((\alpha \wedge \beta)) = \begin{cases} T & \text{if } \overline{v}(\alpha) = T \text{ and } \overline{v}(\beta) = T \\ F & \text{otherwise} \end{cases}$

case 3 For $\overline{v}((\alpha \vee \beta)) = \begin{cases} T & \text{if } \overline{v}(\alpha) = T \text{ or } \overline{v}(\beta) = T \\ F & \text{otherwise} \end{cases}$

case 4 For $\overline{v}((\alpha \Rightarrow \beta)) = \begin{cases} T & \text{if } \overline{v}(\alpha) = F \text{ or } \overline{v}(\beta) = T \\ F & \text{otherwise} \end{cases}$

case 5 For $\overline{v}((\alpha \Leftrightarrow \beta)) = \begin{cases} T & \text{if } \overline{v}(\alpha) = \overline{v}(\beta) \\ F & \text{otherwise} \end{cases}$

where $\alpha, \beta \in \overline{S}$

Truth Assignments

E.g., $(((A_1 \vee A_2) \Rightarrow A_3) \Leftrightarrow (\neg(A_4 \wedge (\neg A_3))))$
Applying $\overline{v}$ with $v(A_1) \mapsto T, v(A_2) \mapsto F, v(A_3) \mapsto F, v(A_4) \mapsto T$ yields

$(((A_1 \vee A_2) \Rightarrow A_3) \Leftrightarrow (\neg(A_4 \wedge (\neg A_3))))$: T

$((A_1 \vee A_2) \Rightarrow A_3)$: F $\qquad (\neg(A_4 \wedge (\neg A_3)))$: F

$(A_1 \vee A_2)$: T $\qquad A_3$: F $\qquad (A_4 \wedge (\neg A_3))$: T

A_1: T $\qquad A_2$: F $\qquad A_4$: T $\qquad (\neg A_3)$: T

A_3: F

Truth Assignments

E.g., $(((A_1 \vee A_2) \Rightarrow A_3) \Leftrightarrow (\neg(A_4 \wedge (\neg A_3))))$
Applying $\overline{v}$ with $v(A_1) \mapsto T, v(A_2) \mapsto T, v(A_3) \mapsto F, v(A_4) \mapsto F$ yields

$(((A_1 \vee A_2) \Rightarrow A_3) \Leftrightarrow (\neg(A_4 \wedge (\neg A_3))))$: F

$((A_1 \vee A_2) \Rightarrow A_3)$: F $\qquad (\neg(A_4 \wedge (\neg A_3)))$: T

$(A_1 \vee A_2)$: T $\qquad A_3$: F $\qquad (A_4 \wedge (\neg A_3))$: F

A_1: T $\qquad A_2$: T $\qquad A_4$: F $\qquad (\neg A_3)$: T

A_3: F

Truth Assignments

The truth or falsity of a wff depends on the interpretations/truth assignments.

- Applying $\overline{v}$ with $v(A_1) \mapsto T, v(A_2) \mapsto F, v(A_3) \mapsto F, v(A_4) \mapsto T$ yields

$$(((A_1 \vee A_2) \Rightarrow A_3) \Leftrightarrow (\neg(A_4 \wedge (\neg A_3))))$$

$$T \quad T \quad F \quad\quad F \quad F \quad\quad T \quad F \quad T \quad\quad T \quad T \quad F$$

- Applying $\overline{v}$ with $v(A_1) \mapsto T, v(A_2) \mapsto T, v(A_3) \mapsto F, v(A_4) \mapsto F$ yields

$$(((A_1 \vee A_2) \Rightarrow A_3) \Leftrightarrow (\neg(A_4 \wedge (\neg A_3))))$$

$$T \quad T \quad T \quad\quad F \quad F \quad\quad F \quad T \quad F \quad\quad F \quad T \quad F$$

Satisfiability and Tautology

- We say a truth assignment v **satisfies** a formula (wff) φ iff $\overline{v}(\varphi) = T$
- A set Σ of wffs **tautologically implies** τ, written $\Sigma \models \tau$, iff every truth assigment for the sentence symbols in $\Sigma; \tau$ that satisfies every member of Σ also satisfies τ
  - $\models$ is about *semantics*, rather than *syntax*
  - For $\Sigma = \emptyset$, we have $\emptyset \models \tau$, simply written $\models \tau$. It says every truth assignment satisfies τ. In this case, τ is a **tautology**.
    - $\models \tau$ should be distinguished from $F \models \tau$ and $\{A, \neg A\} \models \tau$
  - For Σ is a singleton $\{\sigma\}$, we write $\{\sigma\} \models \tau$ as $\sigma \models \tau$
- If $\sigma \models \tau$ and $\tau \models \sigma$, then σ and τ are tautologically equivalent, written as $\sigma \models\dashv \tau$

Compactness Theorem

Theorem (Compactness Theorem)

Let Σ be an infinite set of wffs s.t., for any finite subset $\Sigma_0 \subseteq \Sigma$, there is a truth assignment that satisfies every member of Σ_0. Then there is a truth assignment that satisfies every member of Σ.

Truth Tables

- Consider $(\neg(A \wedge B)) \models ((\neg A) \vee (\neg B))$ (De Morgan's Law)

A	B	$(\neg(A \wedge B))$	$((\neg A) \vee (\neg B))$
F	F	$T\ F\ F\ F$	$TF\ T\ TF$
F	T	$T\ F\ F\ T$	$TF\ T\ FT$
T	F	$T\ T\ F\ F$	$FT\ T\ TF$
T	T	$F\ T\ T\ T$	$FT\ F\ FT$

- More effective enumeration (enumerate product terms rather than minterms)

  E.g., $((A \vee (B \wedge C)) \Leftrightarrow ((A \vee B) \wedge (A \vee C)))$

  $\underline{T}\ T \qquad\qquad\quad T \quad \underline{T}\ T \quad\ T\ \underline{T}\ T$

  $F\ T\ \ \underline{T}\ T\ \underline{T} \quad T \quad F\ T\ \underline{T}\ \ T\ \ F\ T\ \underline{T}$

  $F\ F\ \ \underline{T}\ F\ F \quad T \quad F\ T\ \underline{T}\ \ F\ \ F\ F\ F$

  $F\ F\ \ F\ F\ F \quad T \quad F\ F\ F\ \ F\ \ F\ F\ F$

Selection of Sentential Connectives

Why $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$?

- Can extend the language with other sentential connectives
  - E.g., 3-place majority symbol $\#$
    $\overline{v}(\#\alpha\beta\gamma)$ is agree with the majority of $\overline{v}(\alpha)$, $\overline{v}(\beta)$, $\overline{v}(\gamma)$
  - For any wff in the extended language, there is a tautologically equivalent wff in the original language. (The wff in the original language can be much longer however.)
    E.g., $\#\alpha\beta\gamma$ equals $(\alpha \wedge \beta) \vee (\alpha \wedge \gamma) \vee (\beta \wedge \gamma)$

Formulas and Boolean Functions

A Boolean function $B_\alpha^n : \{F, T\}^n \to \{F, T\}$ can be extracted from a wff α

- An n-place Boolean function B_α^n is defined by $B_\alpha^n(x_1, \ldots, x_n)$ $=$ the truth value given to α when $A_1, \ldots, A_n$ are given the values $x_1, \ldots, x_n$, where $A_1, \ldots, A_n$ are sentence symbols of α
  E.g., $\alpha = (A_1 \vee A_2)$

A_1	A_2	$A_1 \vee A_2$	
F	F	F	$B_\alpha^2(F, F) = F$
F	T	T	$B_\alpha^2(F, T) = T$
T	F	T	$B_\alpha^2(T, F) = T$
T	T	T	$B_\alpha^2(T, T) = T$

Formulas and Boolean Functions

Theorem

Let α and β be wffs whose sentence symbols are among $A_1, \ldots,$ A_n. Then

(a) $\alpha \models \beta$ *iff for all* $\vec{X} \in \{F, T\}^n$, $B_\alpha(\vec{X}) \leq B_\beta(\vec{X})$
  - *Here we impose the order:* $F < T$

(b) $\alpha \models\dashv \beta$ *iff* $B_\alpha = B_\beta$

(c) $\models \alpha$ *iff* B_α *is the constant function with value* T

Formulas and Boolean Functions

Theorem

Let G be an n-place Boolean function, $n \geq 1$. Then there exists a wff α such that $G = B_\alpha^n$ (i.e., α realizes G)

- Every Boolean function is realizable. The realization however is not unique.
- Tautologically equivalent wffs realize the same function

Formulas and Boolean Functions

- For any wff, there is a tautologically equivalent wff in disjunctive normal form (DNF), a.k.a. sum-of-products (SOP)
- Every n-place Boolean function with $n \geq 1$ can be realized by a wff using only the connective symbols $\{\wedge, \vee, \neg\}$
- $\{\wedge, \vee, \neg\}$ is functionally complete
  - $\{\neg, \wedge\}$ and $\{\neg, \vee\}$ are functionally complete
  - $\{\wedge, \Rightarrow\}$ is not functionally complete
- There are 2^{2^n} n-place Boolean functions
  - We can define 2^{2^n} n-ary connectives, each associate with an n-place Boolean function

Compactness

A set Σ of wffs is called **satisfiable** iff there is a truth assignment that satisfies every member of Σ

Theorem (Compactness)

A set Σ of wffs is satisfiable iff every finite subset is satisfiable. That is, Σ is satisfiable iff Σ is **finitely satisfiable**, *namely, every finite subset of Σ is satisfiable.*

Proof (sketch).

($\Longrightarrow$) trivial
($\Longleftarrow$) ideas:

1. Extend Σ to a maximal set Δ that remains finitely satisfiable
2. Utilize Δ to make a truth assignment that satisfies Σ

$\square$

Proof of Compactness Theorem (cont'd)

1. We enumerate the wffs as $\alpha_1, \alpha_2, \ldots$ (countable)
   Define recursively

$$\begin{aligned} \Delta_0 &= \Sigma \\ \Delta_{n+1} &= \begin{cases} \Delta_n; \alpha_{n+1} & \text{if this is finitely satisfiable} \\ \Delta_n; \neg\alpha_{n+1} & \text{otherwise} \end{cases} \end{aligned}$$

   Let $\Delta = \bigcup_{n=1,\ldots} \Delta_n$ (the limit of Δ_n's)
   We know
   - i $\Sigma \subseteq \Delta$
   - ii for every wff α, either $\alpha \in \Delta$ or $\neg\alpha \in \Delta$, and
   - iii Δ is finitely satisfiable
2. Define truth assignment v such that

$$v(A) = T \text{ iff } A \in \Delta$$

   for any sentence symbol A
   Then by induction we can show that v satisfies φ iff $\varphi \in \Delta$
   Since $\Sigma \subseteq \Delta$, v must satisfy every member of Σ

Q.E.D.

Compactness

Corollary

If $\Sigma \models \tau$, then there is a finite $\Sigma_0 \subseteq \Sigma$ such that $\Sigma_0 \models \tau$

Proof.

$\Sigma \models \tau \Leftrightarrow \Sigma; \neg\tau$ is unsatisfiable
For contradiction, assume $\Sigma_0 \not\models \tau$ for every finite $\Sigma_0 \subseteq \Sigma$
$\Longrightarrow \Sigma_0; \neg\tau$ is satisfiable for every finite $\Sigma_0 \subseteq \Sigma$
$\Longrightarrow \Sigma; \neg\tau$ is finitely satisfiable
$\Longrightarrow \Sigma; \neg\tau$ is satisfiable
$\Longrightarrow \Sigma \not\models \tau$ $\square$

Effectiveness and Computability

- Given a set $\Sigma; \alpha$ of wffs, we are concerned about if there is an *effective* procedure that will decide whether or not $\Sigma \models \alpha$
  By *effectiveness*, the computation has to be of
  1. finite exact instructions (programs)
  2. mechanical reasoning
  3. finite run time
- There are uncountably many ($2^{\aleph_0}$) sets of expressions, but only countably many effective procedures (finite instructions)

Decidability vs. Semidecidability

- A set Σ of expressions is **decidable** iff there exists an *effective procedure* (algorithm) that, given an expression α, decides whether or not $\alpha \in \Sigma$
- A set Σ of expressions is **semidecidable** iff there exists an *effective procedure* (semialgorithm) that, given an expression α, produces the answer "yes" iff $\alpha \in \Sigma$
  - For $\alpha \notin \Sigma$, the procedure may or may not produce the answer "no"

Decidability vs. Semidecidability

- There is an effective procedure that, given an expression α, will decide whether or not it is a wff
- There is an effective procedure that, given a finite set $\Sigma; \alpha$ of wffs, will decide whether or not $\Sigma \models \alpha$
- For a finite set Σ of wffs, the set of tautological consequences of Σ is decidable. In particular, the set of tautologies is decidable.
- If Σ is an infinite set (even decidable) of wffs, its set of tautological consequences may be undecidable (Chapter 3)

Effective Enumerability

- A set Σ of expressions is **effectively enumerable** (or called **recursively enumerable**, **computably enumerable**, **Turing recognizable**) iff there exists an effective procedure that lists, in some order, the members of Σ
  - If Σ is infinite, then the procedure can never finish
- A set is effectively enumerable iff it is semidecidable
  - Any decidable set is semidecidable, and thus effectively enumerable
  - A set of expressions is decidable iff both it and its complement are effectively enumerable

Effective Enumerability

- If sets A and B are effectively enumerable, so are $A \cup B$ and $A \cap B$
- If sets A and B are decidable, so are $A \cup B$, $A \cap B$, and $\overline{A}$
- If Σ is a decidable set of wffs, then the set of tautological consequences of Σ is effectively enumerable
- There exists an enumeration for a set iff the set is countable
- Consider enumeration as a surjective (onto) mapping from $\mathbb{N}$ to some set S. S is recursively enumerable if the mapping (function) is computable
  - A function is (**effectively**) **computable** iff there exists an effective procedure that, given an input x, will eventually produce the correct output $f(x)$