5. Propositional and Predicate Logic

CS 3030 Lecture Notes
Yan Shi
UW-Platteville

Read: Textbook Chapter 7
What is Logic?

- **Reasoning** about the **validity** of arguments.

- An argument is valid if its conclusions follow logically from its premises – even if the argument doesn’t actually reflect the real world:
  
  — All lemons are blue
  — Mary is a lemon
  
  \[\text{Premises}\]

  — Therefore, Mary is blue.

  \[\text{Conclusion}\]

- **Truth values**: true/false.
  
  — fundamental units of logic.
Logical Operators

- And \( \land \) (conjunction)
- Or \( \lor \) (disjunction)
- Not \( \neg \)
- Implies \( \rightarrow \) (if... then... / implies)
- Iff \( \leftrightarrow \) (if and only if)

The order of precedence:

\([(), \neg, \land, \lor, \rightarrow, \leftrightarrow]\)
Translating between English and Logic

- Facts and rules need to be translated into logical notation.
  - It is Raining and it is Thursday:
    - \( R \land T \)
    - R means “It is Raining”, T means “it is Thursday”.

- More complex sentences need predicates of property(object)
  - It is raining in New York:
    - \( R(N) \)
    - Could also be written \( N(R) \), or even just R.

- **granularity**: It is important to select the correct level of detail for the concepts you want to reason about.
Guideline to Translation

- Simple declarative sentence as atomic propositions
  - Alice is happy: H(A) or simply P
- Identify connecting words such as and, but, or, if...then, iff, just in case, unless, only if, when, etc. Translate them into logical operators.
- Determine the order. Use parenthesis if necessary.
Exercise

- Translate the following sentence to logic:
  - Bob stayed up late last night.
  - It is not the case that Alice isn’t sick.
  - Chris is singing but David isn’t listening.
  - It is not true that Alice is sick and Chris is singing.
  - Alice is sick and Bob stayed up late last night or Chris is singing.
  - If Alice isn’t sick and Chris is singing, David is listening.
  - If Alice isn’t sick, Chris is singing, and vice versa.
  - I only eat a pizza and a burger when I am very hungry!
  - Whenever Chris sings, David listens.
  - Whenever he eats burger that have pickles, he ends up either asleep at his desk or singing loud songs.
Truth Table

- Tables that show truth values for all possible inputs to a logical operator.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>~A</td>
<td>A&amp;B</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>

- A truth table shows the semantics of a logical operator.
Complex Truth Tables

- We can produce truth tables for complex logical expressions, which show the overall value of the expression for all possible combinations of variables:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>A ∧ (B ∨ C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

- A truth table for n variables will have $2^n$ lines.
Tautology

- **Tautology**: an expression that is true under any interpretation.
  - A is a tautology: this is written as $\vdash A$
  - $A \lor \neg A$
  - $A \rightarrow A$

- **Contradictory**: an expression which is false under any interpretation. ($\bot$)

- An expression is **satisfiable** if they are true under some interpretation.
Equivalence

- Two expressions are equivalent if they always have the same logical value under any interpretation:
  - $A \land B \equiv B \land A$

- Equivalences can be proven by examining truth tables.
Some Useful Equivalences (1)

- $A \lor A \equiv A$
- $A \land A \equiv A$
- $A \land (B \land C) \equiv (A \land B) \land C$  \hspace{1cm} (associative)
- $A \lor (B \lor C) \equiv (A \lor B) \lor C$  \hspace{1cm} (associative)
- $A \land (B \lor C) \equiv (A \land B) \lor (A \land C)$  \hspace{1cm} (distributive)
- $A \land (A \lor B) \equiv A$
- $A \lor (A \land B) \equiv A$

- $A \land \text{true} \equiv A$
- $A \land \text{false} \equiv \text{false}$
- $A \lor \text{true} \equiv \text{true}$
- $A \lor \text{false} \equiv A$
Some Useful Equivalences (2)

- \( A \lor B \equiv \neg(\neg A \land \neg B) \)
- \( A \land B \equiv \neg(\neg A \lor \neg B) \)
  
  DeMorgan’s Laws!

- \( A \rightarrow B \equiv \neg A \lor B \)
- \( A \leftrightarrow B \equiv \neg(\neg(\neg A \lor B) \lor \neg(\neg B \lor A)) \)
  
  This means we don’t need \( \rightarrow \) and \( \leftrightarrow \) symbols at all!

- Any binary logical operator can be expressed using \( \neg \) and \( \lor \).

- We can use equivalences to simplify logical expressions:
  
  \( \neg (C \land D) \lor ((C \land D) \land E) \)
Propositional Logic

- Propositional logic is a logical system.
- It deals with propositions.
- Propositional Calculus is the language we use to reason about propositional logic.

Syntax:
- \( \Sigma = \{ \text{true, false, } \neg, \rightarrow, (, ), \land, \lor, \iff, p_1, p_2, \ldots, p_n, \ldots \} \) : set of legal symbols
- A sentence in propositional logic is called a well-formed formula (wff).

Semantics:
- Defined by truth tables: “what does a wff mean?”
- \( P \land Q \) means “true when \( P \) is true and \( Q \) is true”
Deduction

- The process of deriving a conclusion from a set of assumptions.

- If we deduce a conclusion C from a set of assumptions, we write:
  \[
  \{A_1, A_2, \ldots, A_n\} \vdash C
  \]

- If C can be concluded without any assumptions, we write:
  \[
  \vdash C
  \]

- Use a set of **inference rules** to perform deduction.
Inference Rules

- **∧-Introduction:**
  \[
  \begin{array}{c}
  A \\
  \hline
  B \\
  \end{array}
  \]
  \[
  A \land B
  \]
  Given A and B, we can deduce A∧B.

- **∧-Elimination:**
  \[
  \begin{array}{cc}
  A \land B & A \land B \\
  \hline
  A & B \\
  \end{array}
  \]
  Given A∧B, we can deduce A and we can deduce B.

- **∨-Introduction:**
  \[
  \begin{array}{c}
  A \\
  \hline
  \end{array}
  \]
  \[
  A \lor B
  \]
  Given A, we can deduce the disjunction of A with any expression.
Inference Rules

- **Introduction:**
  
  A
  
  ...

  \[ \underline{C} \]

  A \rightarrow C

  If in carrying out a proof we start from an assumption A and derive a conclusion C, then we can deduce A \rightarrow C.

- **Elimination** (Modus Ponens):

  A A \rightarrow B

  \[ \underline{B} \]

  B

  If A is true and A implies B, then we can deduce B is true.
Inference Rules

- **Reductio Ad Absurdum** (proof by contradiction):
  
  \[ \neg A \]
  
  \[ \vdash \bot \]  
  
  \[ A \]

If we assume A is false and this leads to a contradiction, then we can deduce that A is true.

- **\(\neg\neg\) Elimination**:

  \[ \neg\neg A \]
  
  \[ A \]

  If we have a sentence that is negative twice, we can conclude that the sentence itself.
Deduction Example

- \{A \land B\} \vdash A \lor B
- \{P, P \rightarrow Q\} \vdash P \land Q
- \vdash (\neg A \rightarrow B) \rightarrow (\neg B \rightarrow A)

\[
\begin{array}{c}
\neg A \\
\neg A \rightarrow B \\
\hline
B \\
\neg B \\
\hline
B \\
B \rightarrow \bot \\
\hline
\bot \\
\hline
A \\
\hline
\neg B \rightarrow A
\end{array}
\]

assumptions
modus ponens
rewriting \neg B
modus ponens
reductio ad absurdum
\rightarrow introduction
\rightarrow introduction
Deduction Theorem

- if \( A \cup \{B\} \vdash C \), then \( A \vdash (B \rightarrow C) \)
- if \( A \vdash (B \rightarrow C) \), then \( A \cup \{B\} \vdash C \)

- “If it is sunny and I am energetic, I will go hiking.” is the same as “When it is sunny, I will go hiking if I am energetic.”
- can make propositional logic proof easier:
  - Prove \( \{A \rightarrow B, B \rightarrow C\} \vdash A \rightarrow C \)
  - Prove \( \{A \rightarrow B\} \vdash A \rightarrow (C \rightarrow B) \)
Exercises

- \{P \land Q \rightarrow R, Q \rightarrow P, Q\} \vdash R
- \{P \rightarrow Q, Q \rightarrow R\} \vdash P \rightarrow Q \land R
- \{P \rightarrow Q, \neg Q\} \vdash \neg P
- \vdash (A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (C \rightarrow D) \rightarrow (A \rightarrow D))

Predicate Calculus

- Limitation of propositional calculus: no *properties* of objects and *relationships* between objects in propositions
  ➜ Use predicates!

- **Predicates:**
  - *property*(object): New York is raining: R(N)
  - *relationship*(objects): I like cheese: L(me, cheese)

- **Functions:** f(x1,x2,...,xn)
  - My mom likes cheese: L(m(me), cheese)

- A *predicate* with variables can be made a *proposition* by:
  - assign a value to the variable, or
  - quantify the variable using a *quantifier*
Quantifiers

- **Universal quantifier:** ∀ (for all)
  - everybody likes cheese: ∀x P(x) → L(x,C)

- **Existential quantifier:** ∃ (there exists)
  - someone likes cheese: ∃x L(x, C)

- The quantifiers have higher precedence than all logical operators.

- We can combine quantifiers:
  - everyone likes something: ∀x ∃y L(x,y)

- Relationships between ∀ and ∃:
  - ∀x L(x,C) → ∃x L(x,C)
  - ∃x ≡ ¬ ∀x ¬. e.g., ∃x L(x, C) ≡ ¬ ∀x ¬ L(x, C)

- Bound and Free Variables:
  \[ (\forall x)(P(x, y, z) \to (\exists y)(Q(y, z))) \]
  - bound free bound free
First-Order Predicate Logic (FOPL)

- FOPL: The quantifiers can be applied to individuals.
- SOPL: Quantify over predicates, functions and sets of variables

- Term:
  - constant
  - variable
  - $f(x_1, \ldots, x_n)$, if $x_1, \ldots, x_n$ are all terms.

- atomic formula: wff in the form of $P(x_1, \ldots, x_n)$.
- well-formed formula (wff): a sentence
  - contains predicates, quantifiers and variables
FOPL Summary

- Syntax:
  - terms
  - quantifiers
  - wff

- Semantics:
  - world has infinite set of objects
  - properties and relations over objects
  - connectives defined by truth table

- Deduction:
  - start with same rules as for propositional logic
  - add natural deduction rules for ∀, ∃
Exercises

- Everybody likes somebody
- Nobody likes everybody.
- Someone likes everyone.
- Everyone has a mother.
- Only snakes and lizards live in the desert.
- Oranges and lemons are citrus fruits.

- \( \exists \) almost always goes with \( \land \), \( \forall \) with →
Properties of a Logic System

- **Axiom**: a fundamental truth \((A \rightarrow (B \rightarrow A))\)
- **Theorem**: can be proved by the rules of deductions from axioms \((\vdash A)\)
- **Sound**: every theorem is a tautology.
- **Complete**: every tautology is a theorem.
- **Decidable**: it is possible to produce an algorithm that will determine whether any wff is a theorem.
- **Monotonic**: a valid proof cannot be made invalid by adding additional premises or assumptions.
  - If we can prove \(\{A, B\} \vdash C\), then we can also prove \(\{A, B, X, Y\} \vdash C\)
## Properties of Propositional Logic and FOPL

<table>
<thead>
<tr>
<th></th>
<th>Propositional</th>
<th>FOPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sound</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Complete</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Decidable</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Monotonic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Abduction and Inductive Reasoning

- **Deductive reasoning:**
  - start from known facts, derive conclusions using sound rules

- **Inductive reasoning:**
  - reasoning from what we've seen before: “I've never seen a crow that isn't black and I've seen a lot of crows, so I assume all crows are black.”
  - even though it's not always correct, it's certainly a powerful tool

- **Abduction:**
  - \( \{B, A \rightarrow B\} \vdash A \)
  - similar to modus pones but is not logically sound.
  - however, provide a model that works reasonably well in the real world ➔ guess the cause of an observation.
Summary

- Logic:
  - validity vs. truth value
  - reasoning
  - translate English into logic
  - syntax, semantics, proof system
- Propositional logic:
  - truth table
  - natural deduction
  - simple, but inadequate for many problems
- Predicate logic:
  - function, quantifiers
  - much more expressive, harder to work with
- Soundness, completeness, decidability, monotonicity
- deduction, induction, abduction