Basic propositional and predicate logic

The fundamentals of deduction
Logic and its application

• Logic is the study of the patterns of deduction

• Logic plays two main roles in computation:
  – **Modeling**: logical sentences are the building blocks of computation models
  – **Analyzing**: logic proofs are fundamental for reasoning about, and inferring properties of computational systems’ models
Basic elements in logic

• **Declarative sentences**
  also called **Propositions**
  Is any statement which can be **unambiguously** determined to be either **True** or **False**

• **Logical connectives**
  Connect logical statements. There are four logical connectives

<table>
<thead>
<tr>
<th>NAME</th>
<th>Semantic</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negation</td>
<td>NO</td>
<td>(\neg)</td>
</tr>
<tr>
<td>Conjunction</td>
<td>AND</td>
<td>(\land)</td>
</tr>
<tr>
<td>Disjunction</td>
<td>OR</td>
<td>(\lor)</td>
</tr>
<tr>
<td>Implication</td>
<td>IF – THEN</td>
<td>(\implies)</td>
</tr>
</tbody>
</table>
Atomic and compound sentences

• An **atomic** sentence is a proposition that does not contain any logical connectives
  – Example: A:“3 > 4”
  – This is clearly a **false** atomic sentence

• A **compound** sentence is a proposition composed by two or more sentences linked with logical connectives
  – Example: A:“3 > 4 ⇒ 2 > 3”
  – This is a compound proposition since it **has an implication**, but
  – Is this proposition true or false?
Proof system

• The previous question cannot be properly answer without a well-established **Proof System**

• A *proof system* is a set of rules and operations for determining the truth value of a compound sentence out of the truth value of its atomic constituents

• Examples of proof systems are:
  – **Boolean algebra** (to be reviewed soon)
  – **Natural deduction** (modus ponens, modus tollens, etc. *Not in the scope of this course*)
Variables

• In science, **variables** are **place holders of values**. As such, variables have:
  • **Meaning** – what the **symbol** represents
  • **Value** – any value in the **domain** of the variable
• **Example**: Let’s represent the physical concept of **time with a variable**. Let
  – *t* be the **symbol** denoting the time
  – The natural **domain** of *t* is the set of all nonnegative real numbers.

  **We write:**
  – *t* : time when **defining** the **meaning of symbol** *t*
  – *t* = 1.25 seconds when **evaluating** (i.e. choosing a **value in the domain of**) *t*
Boolean Variables

• A **Boolean variable** is any variable whose domain is the set \{0, 1\} (i.e. can take only the values 0 or 1)

• Propositions are normally represented by Boolean variables. The values that such variable take are conventionally interpreted as
  – 0 if the proposition is **false** and
  – 1 if the proposition is **true**

• Example of a Boolean variable
  – A : “3 is less than 3.2” (meaning of symbol A)
  – A = 1 (value of A)
Propositional vs. predicate logic

• **Propositional logic** deals solely with propositions and logical connectives
  – Example: A: “3 < 4”
  – Clearly A=1
• **Predicate logic** adds **predicates** and **quantifiers**
• A predicate is a **logical statement that depends on one or more variables** (not necessarily Boolean variables)
  – Example: \( B(x, y) : \text{“x is less than y”} \)
  The definition and value of \( B(x, y) \) depend on \( x \) and \( y \)
  • If \( x = 3 \) and \( y = 4 \), then \( B(x, y) = B(3, 4) = 1 \)
  • If \( x = 4 \) and \( y = 3 \), then \( B(x, y) = B(4, 3) = 0 \)
  • If \( x = + \) and \( y = / \), the \( B(x, y) \) is (in principle) undefined
Quantifiers

Quantifiers are used to declare ranges of a predicate’s variables in which the predicate assumes either 0 or 1, but no both.

A predicate with no quantifiers is said to be a free predicate.

There are two quantifiers:

<table>
<thead>
<tr>
<th>NAME</th>
<th>SEMANTIC</th>
<th>SYMBOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal</td>
<td>FOR ALL</td>
<td>( \forall )</td>
</tr>
<tr>
<td>Existential</td>
<td>THERE EXISTS</td>
<td>( \exists )</td>
</tr>
</tbody>
</table>
Consider the free predicate $A(x, y): "x < y \implies x < y^2"$

where $x$ and $y$ are real numbers.

**Some values** for this predicate are:
- $A(.5, .9) = 1$
- $A(.5, .6) = 0$
- $A(-.6, -.5) = 1$
- $A(.9, .93) = 0$
- $A(.9, .96) = 1$

**By imposing quantifiers** we may produce the following **true** predicates:

- $B(x, y) : "(\forall x)(x \geq 1) A(x, y)"
- $C(x, y) : "(\forall x, y)(0 \leq x \land y \leq \frac{2}{\sqrt{x}} < 1) \neg A(x, y)"
- $D(x, y) : "(\forall x)(x < 0) A(x, y)"
- $E(x, y) : "(\forall x, y)(0 \leq x \land \frac{2}{\sqrt{x}} < y < 1) A(x, y)"

**Remark:** A predicate that is “**always true or always false**” can be treated as a **proposition** that is true or false, respectively. This is, it can be evaluated and analyzed with the tools of **propositional logic**.
Propositional logic

• A logical theory that involves only propositions (i.e. it has no predicates) is called Propositional Logic.

• There are two main versions of Propositional Logic:
  – Semantic propositional logic
  – Propositional calculus

• These two versions differ only in their proof rules
  – Proofs in semantic propositional logic are based on Boolean algebra
  – Proofs in Propositional calculus are based on the so-called rules of natural deduction
Truth tables and valuations

• A **valuation** of a sentence is an assignment (any assignment) of truth values to the statements that compose the sentence.

• If we combine two statements with a logical connective, the value of the resulting compound statement is determined by two things:
  – The chosen valuation
  – The semantic (meaning) of the logical connective

• The meaning of each connective under all possible valuations is declared in a **truth table**.
The truth table of negation

- The simplest case of truth value is negation. In this case, there is only one sentence, and therefore, only two possible valuations.
- The semantic is also very natural: if the sentence is true, its negation is false, and vice versa.
- Thus, the truth table of \textit{negation} is simply

<table>
<thead>
<tr>
<th>A</th>
<th>(\neg A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Disjunction and conjunction truth tables

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A \land B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A \lor B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

There are four possible valuations in each of these cases.
Implication’s truth table: Hypothesis->Thesis

In implications, the variable A represents a hypothesis statement while the variable B, a thesis statement.

The first two rows in this table indicate that when the hypothesis is false, the implication will not discriminate between a true or a false thesis. It will always return true. These are called vacuously true implications.

The second two rows tell a different story. In this case (as we always hope in practice) the hypothesis is true. Then, the truth value of the thesis is faithfully reflected in the value of the implication. If the thesis is false, so is the implication, and vice versa.
Computing with truth tables

We are now in position to answer the question
Is A : ”3 < 4 ⇒ 2 < 3” true or false?

**Answer:** Identify the atomic sentences that compose A. Let B : “3 < 4”, and C : “2 < 3”
Now, we see that A has the form B ⇒ C
Since B = 0 and C = 0, from the very first row of the table of implication we conclude that A is True (although vacuously!)

Is there a valuation making the statement true?

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>¬P</th>
<th>P ∧ (¬P ∧ Q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Yes: P=0, Q=1
Equivalence between propositions

- Two propositions $A$ and $B$ are said to be **equivalent** if their truth values are the same in all possible valuations. Equivalence is denoted $A \iff B$

- Equivalence is not the same as equality: two propositions may be equivalent even if they differ in their meaning

- Equivalence is a shorthand for

\[(A \Rightarrow B) \land (B \Rightarrow A)\]
Important equivalences

- **Conjunctive form of implication**
  \[(P \Rightarrow Q) \iff (\neg P \lor Q)\]

- **Contrapositive form of implication**
  \[(P \Rightarrow Q) \iff (\neg Q \Rightarrow \neg P)\]

- **Idempotency rules**
  \[-(\neg P) \iff P\]
  \[P \land P \iff P\]
  \[P \lor P \iff P\]

- **De Morgan’s rules**
  \[-(P \lor Q) \iff (\neg P \land \neg Q)\]
  \[-(P \land Q) \iff (\neg P \lor \neg Q)\]

- **Distributive rules**
  \[P \land (Q \lor R) \iff (P \land Q) \lor (P \land R)\]
  \[P \lor (Q \land R) \iff (P \lor Q) \land (P \lor R)\]

- **Associative rules**
  \[P \land (Q \land R) \iff (P \land Q) \land R\]
  \[P \lor (Q \lor R) \iff (P \lor Q) \lor R\]

- **Commutative rules**
  \[P \lor Q \iff Q \lor P\]
  \[P \land Q \iff Q \land P\]
Some applications

Demonstrate that
\[ \neg(p \vee \neg(r \land \neg s)) \]
is equivalent to
\[ \neg(p \vee s) \land r \]

\textbf{Proof:}
\[
\begin{align*}
\neg(p \vee \neg(r \land \neg s)) & \\
\iff \neg p \land (r \land \neg s) & \\
\iff \neg p \land (\neg s \land r) & \\
\iff (\neg p \land \neg s) \land r & \\
\iff \neg(p \vee s) \land r & \\
\end{align*}
\]

Demonstrate that
\[ p \Rightarrow (q \Rightarrow (r \Rightarrow s)) \]
is equivalent to
\[ (p \land q \land r) \Rightarrow s \]

\textbf{Proof:}
\[
\begin{align*}
p \Rightarrow (q \Rightarrow (r \Rightarrow s)) & \\
\iff \neg p \lor (q \Rightarrow (r \Rightarrow s)) & \\
\iff \neg p \lor (\neg q \lor (r \Rightarrow s)) & \\
\iff \neg p \lor (\neg q \lor (\neg r \lor s)) & \\
\iff (\neg p \lor \neg q \lor \neg r) \lor s & \\
\iff \neg(p \land q \land r) \lor s & \\
\iff (p \land q \land r) \Rightarrow s & \\
\end{align*}
\]
The negation of a predicate

• Negating a predicate that contains quantifiers demands, in addition, the use of the semantic table.

• These negation rules must be applied together with the previously discussed equivalences, when treating predicates as propositions.

<table>
<thead>
<tr>
<th>QUANTIFIER</th>
<th>NEGATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal</td>
<td>Existential</td>
</tr>
<tr>
<td>Existential</td>
<td>Universal</td>
</tr>
</tbody>
</table>
Yet another important equivalence

• The next equivalences describe the actual negation operation for elemental predicates with quantifiers

• Let \( P(x, y, \ldots, z) \) be a predicate. Then,

\[ \neg(\forall x, y, \ldots, z)P(x, y, \ldots, z) \iff (\exists x \vee \exists y \vee \ldots \vee \exists z)\neg P(x, y, \ldots, z) \]
\[ \neg(\exists x, y, \ldots, z)P(x, y, \ldots, z) \iff (\forall x \vee \forall y \vee \ldots \vee \forall z)\neg P(x, y, \ldots, z) \]
Let’s consider the negation of

\[ B : (\forall x)(1 \leq x)A(x, y) \]

where

\[ A(x, y) : x < y \implies x^2 < y^2 \]

**Answer:**

\[ \neg B \iff (\exists x)(1 \leq x)\neg A(x, y) \]

• **Remark:** this is, as expected, a false statement

The informal expression

\[ (\forall x, y) \]

is a short hand for

\[ (\forall x) \land (\forall y) \]

In general, the symbol “, ” is a to be taken as a short hand for “and”

Thus,

\[ \neg (\forall x, y) \iff (\exists x)\lor (\exists y) \]

and

\[ \neg((\forall x, y)(0 \leq x \land y \leq 1)\neg A(x, y)) \iff (\exists x \lor \exists y)(0 \leq x \land y \leq 1)A(x, y) \]
Composed quantifiers

Also common are predicates whose quantifier is a composition of the universal and the existential quantifiers

- For each $x$ there is a $y$ such that $P(x, y)$ is true
  \[(\forall x)(\exists y) P(x, y)\]
- There is an $x$ such that for all $y$ $P(x, y)$ is true
  \[(\exists x)(\forall y) P(x, y)\]
- Their negations are
  \[(\exists x)(\forall y) \neg P(x, y)\text{, and}\]
  \[(\forall x)(\exists y) \neg P(x, y)\]

Respectively
Boolean formulas

• A **Boolean formula** is a “well-formed” combination of Boolean variables and logical connectives
  – The combination \( p \Rightarrow (\neg q \land (s \land \neg r)) \) is a Boolean formula
  – The combination \( \Rightarrow p \neg \)qsr (\lor \land \lor \) is not a Boolean formula

• Any representation of a proposition in terms of Boolean variables and logical connectives yields a Boolean formula.
Two important problems

• Given a combination of symbols and logical connectives: Is it a Boolean formula?
  – Domain: A set of symbols and logical connectives
  – Instance: A string of symbols and logical connectives
  – Question: Is the string a Boolean formula?

• Given a Boolean formula: can we find a valuation that makes it true?
  – Domain: All Boolean formulas
  – Instance: A Boolean formula
  – Question: Is there a valuation that makes the formula true?