Introduction to CS 250, Fall 2014

• I will expect you to read the assigned parts of Rosen, 7th Edition, carefully, before each relevant lecture. Read with pen and paper in hand. Write down important concepts and anything you have a question about. If you haven’t been able to answer your questions by the time you are done reading, then ask me in class or by email the night before class.

• I will help you learn how to read and learn from such a book.

• I will add additional material and provide exercises and problems to help you learn

• **Check at least twice a week:** Syllabus Page: readings, problem sets, handouts and Moodle Page: online quizzes, how you’re doing.

peopel.cs.umass.edu/~immerman/cs250
What You Will Learn in 250 [cf. Rosen, p. vii]

1. **Mathematical Reasoning:** how to understand and construct valid mathematical proofs and to appreciate the certainty that ensues. ***a truly new way to think***

2. **Combinatorial Analysis:** → this will be covered instead in CS 240: Reasoning under Uncertainty

3. **Discrete Structures:** sets, permutations, relations, functions, integers, graphs, trees, finite automata

4. **Algorithmic Thinking:** main topic of CS 311, we will give an introduction here.

5. **Applications and Modeling:** how to model the real problem you need to solve as a purely mathematical problem, and then how to interpret your solution of the mathematical problem for the real problem.
Rosen Chapter 1: Mathematical Logic and Proofs

Logic was already well articulated by Aristotle (384 - 322 BC). It codifies correct reasoning.

Mathematical Logic was well established much later, e.g., Tarski (1901 - 1983). It formalized mathematical reasoning. It has become crucial in computer science for understanding the correctness and security of programs, especially given enormous complexity:

- huge programs written by many people
- huge memory size and number of instructions per second executed by each processor
- many processors running simultaneously
- many players with very different motives and strategies
- constant change of languages, requirements, hardware
1.1 Propositional Logic

Propositional logic is sometimes called Propositional Calculus or Boolean Logic.

A proposition is a declarative statement that states a completely precise fact which is thus either true or false.

<table>
<thead>
<tr>
<th>Possible Propositions</th>
<th>Prop?</th>
<th>T/F/?</th>
<th>Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obama was elected president in 2008.</td>
<td>Y</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Bush was elected president in 2000.</td>
<td>Y</td>
<td>T</td>
<td>which Bush?</td>
</tr>
<tr>
<td>Gore would have won the election in 2000 had the Florida ballots been counted accurately.</td>
<td>Y</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>Clinton was elected president in 1992.</td>
<td>Y</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Bush was elected president in 1988.</td>
<td>Y</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>[x + y = z]</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[x + y = z], where [x = 1], [y = 2], [z = 3], and “+” refers to addition over the natural numbers.</td>
<td>Y</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Clinton will be elected president in 2016.</td>
<td>Y</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>This sentence is true.</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This sentence is false.</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This sentence is a proposition.</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This sentence is not a proposition.</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Twas brillig, and the slithy toves did gyre and gimble in the wabe.</td>
<td>N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It is subtle exactly what constitutes a proposition. I won’t test you on this at least until we have studied the semantics of predicate logic. In 250 we will only care about propositions as mathematical statements.

In English, much is ambiguous, depending on context and shared knowledge.

When we call an English sentence a proposition, we mean that it could be transformed into a proposition by rigorously defining all of the ambiguous words and phrases, in the “natural way”, e.g., “Obama” means Barack Obama, current president of the US; president means president of the US; etc.
Propositional Variables and Truth Values

**Propositional Variables:**  \( \text{PVar} = \{p, q, r, s, p_1, q_1, r_1, s_1, p_2, q_2, r_2, s_2, \ldots \} \)

**Truth Values:**  true:  \( T, 1, \top \)  \hspace{2em} false:  \( F, 0, \bot \)

Propositional variables are **atomic representations of propositions** – we can’t look inside.

The **semantics** (meaning) assigned to propositional variables is \( T \) or \( F \), period.
Syntax of Prop Logic: PVar plus boolean connectives

1. nullary: $T, F$

2. unary: $\neg$: not

3. binary: $\land$ (and); $\lor$ (or); $\rightarrow$ (implies); $\leftrightarrow$ (iff); $\oplus$ (exclusive or)

Examples of Prop Logic formulas:

- $T$
- $p$
- $\neg p$
- $p \rightarrow F$
- $p \land r$
- $(p \land r) \rightarrow p$
- $\neg ((p \land r) \rightarrow p)$
- $(p \oplus q) \leftrightarrow (p \leftrightarrow \neg q)$
Precedence of Operators in PropCalc

- \( \neg \)
- \( \land, \lor, \oplus \)
- \( \rightarrow, \leftrightarrow \)

\( \land, \lor, \text{ and } \oplus \) are associative; \( \rightarrow \) associates from right to left.

\[
\neg a \land b \rightarrow c \equiv ((\neg a) \land b) \rightarrow c
\]

\[
a \rightarrow b \rightarrow c \equiv a \rightarrow (b \rightarrow c)
\]


**Semantics of Prop Logic: Truth Tables**

**Definition 1.1 (Prop Logic World)** A **Prop Logic World** \( w : A \rightarrow \{0, 1\} \) is a function from \( \text{dom}(w) \subseteq \text{PVar} \) to \( \{0, 1\} \), i.e., it gives a truth value to every element in its domain.

A Prop Logic World is an environment that gives meaning (semantics) to every appropriate Prop Logic formula, \( f \).

**Definition 1.2** Prop Logic World \( w \) is **appropriate** for Prop Logic formula \( f \) iff \( \text{var}(f) \subseteq \text{dom}(w) \).

**Proposition 1.3** Prop Logic World \( w \) determines a truth value for any appropriate Prop Logic formula \( f \).

<table>
<thead>
<tr>
<th>world ( w )</th>
<th>( p )</th>
<th>( q )</th>
<th>( T )</th>
<th>( F )</th>
<th>( \neg p )</th>
<th>( p \land q )</th>
<th>( p \lor q )</th>
<th>( p \rightarrow q )</th>
<th>( p \leftrightarrow q )</th>
<th>( p \oplus q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_0 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( w_1 )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( w_2 )</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( w_3 )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

The following recursive algorithm extends the domain of \( w \) to the set of all Prop Formulas for which \( w \) is appropriate:

\[
\begin{align*}
    w(T) & \quad \text{def} = 1 \\
    w(F) & \quad \text{def} = 0 \\
    w(\neg a) & \quad \text{def} = 1 - w(a) \\
    w(a \land b) & \quad \text{def} = \min(w(a), w(b)) \\
    w(a \lor b) & \quad \text{def} = \max(w(a), w(b)) \\
    w(a \rightarrow b) & \quad \text{def} = w(\neg a \lor b) = \max(1 - w(a), w(b)) \\
    w(a \leftrightarrow b) & \quad \text{def} = w(a) == w(b) \\
    w(a \oplus b) & \quad \text{def} = w(a) + w(b) \pmod{2}
\end{align*}
\]

We will give a rigorous proof of Prop. 1.3 using mathematical induction in Lecture 21.

The meaning of \( \neg, \land, \lor, \rightarrow, \leftrightarrow, \oplus \) is defined by the above equations. We pronounce these symbols using familiar English words: “not”, “and”, “or”, “implies”, “iff”, “exclusive or”, not to confuse you, but to help you remember their definitions.
More About Appropriate Prop Logic Worlds

Prop Logic World $w$ is appropriate for Prop Logic formula $f$ iff $\text{var}(f) \subseteq \text{dom}(w)$.

A Prop Logic World corresponds to a line of a truth table.

<table>
<thead>
<tr>
<th>world</th>
<th>$p$</th>
<th>$q$</th>
<th>T</th>
<th>F</th>
<th>$\neg p$</th>
<th>$p \land q$</th>
<th>$p \lor q$</th>
<th>$p \rightarrow q$</th>
<th>$p \leftrightarrow q$</th>
<th>$p \oplus q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_0$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$w_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$w_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$w_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$w_3 = \{(p, 1), (q, 1)\}$; $w_2 = \{(p, 1), (q, 0)\}$; $w_1 = \{(p, 0), (q, 1)\}$; $w_0 = \{(p, 0), (q, 0)\}$

$\text{dom}(w_3) = \text{dom}(w_2) = \text{dom}(w_1) = \text{dom}(w_0) = \{p, q\}$,

So $w_0, \ldots w_3$ are appropriate for the following Prop Logic formulas:

<table>
<thead>
<tr>
<th>$f$</th>
<th>T</th>
<th>$p$</th>
<th>$\neg p$</th>
<th>$\neg q$</th>
<th>$p \rightarrow q$</th>
<th>$(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{var}(f)$</td>
<td>$\emptyset$</td>
<td>${p}$</td>
<td>${p}$</td>
<td>${q}$</td>
<td>${p, q}$</td>
<td>${p, q}$</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>

But they are not appropriate for any of the following Prop Logic formulas:

<table>
<thead>
<tr>
<th>$f$</th>
<th>$r$</th>
<th>$p \lor q \lor r$</th>
<th>$p \land (r \lor \neg r)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{var}(f)$</td>
<td>${r}$</td>
<td>${p, q, r}$</td>
<td>${p, r}$</td>
</tr>
</tbody>
</table>

Notice that $p \equiv p \land (r \lor \neg r)$, i.e., they look different but they mean the same thing, so appropriateness is a syntactic property of formulas, not a semantic property.
$p \rightarrow q$  not concerned with causality

Only the truth values of $p$ and $q$ matter

$p$ implies $q \equiv$ if $p$ then $q \equiv \neg p \lor q$

$p \rightarrow q \equiv \neg q \rightarrow \neg p$  contrapositive
is equivalent.

$p \rightarrow q \not\equiv q \rightarrow p$  converse
not equivalent.

<table>
<thead>
<tr>
<th>world</th>
<th>$p$</th>
<th>$q$</th>
<th>$\neg p$</th>
<th>$\neg q$</th>
<th>$p \rightarrow q$</th>
<th>$q \rightarrow p$</th>
<th>$\neg p \lor q$</th>
<th>$\neg q \rightarrow \neg p$</th>
<th>$\neg p \rightarrow \neg q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_0$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$w_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$w_2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$w_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
## English to Prop Logic Translation Table

<table>
<thead>
<tr>
<th></th>
<th>$p$ implies $q$</th>
<th>$q$ implies $p$</th>
<th>$p$ unless $q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>if $p$ then $q$</td>
<td>$p$ if $q$</td>
<td>$q$ if $p$</td>
<td></td>
</tr>
<tr>
<td>$p$ only if $q$</td>
<td>$p$ iff $q$</td>
<td>$p$ iff $q$</td>
<td></td>
</tr>
<tr>
<td>$p$ is sufficient for $q$</td>
<td>$p$ is necessary for $q$</td>
<td>$p$ is necessary and sufficient for $q$</td>
<td></td>
</tr>
</tbody>
</table>

**world** | $p$ | $q$ | $p \to q$ | $q \to p$ | $p \leftrightarrow q$ | $\neg q \to p$ | $\neg p \to q$ | $p \lor q$ |
--- | --- | --- | --- | --- | --- | --- | --- | --- |
$w_0$ | 0 | 0 | 1 | 1 | 1 | 0 |
$w_1$ | 0 | 1 | 1 | 0 | 0 | 1 |
$w_2$ | 1 | 0 | 0 | 1 | 0 | 1 |
$w_3$ | 1 | 1 | 1 | 1 | 1 | 1 |

English is ambiguous; Prop Logic is precise.

Translating between them can thus be subtle.
Translating from English to Prop Logic.

“You may not ride the roller coaster if you are under 4 feet tall, unless you are older than 16.”

- Use intuitive variables: \( r = \text{may ride}; \)
  \( s = \text{under 4 ft., i.e., short}; \)
  \( o = \text{older than 16} \)

- Substitute the variables into the English: \( \neg r \text{ if } s, \text{ unless } o \)

- Translate, using the English to Prop Logic Translation Table.

\[
\begin{align*}
\neg r \text{ if } s & \equiv s \rightarrow \neg r \\
(s \rightarrow \neg r), \text{ unless } o & \equiv \neg o \rightarrow (s \rightarrow \neg r)
\end{align*}
\]

- Simplify, using truth tables or other methods we will learn: \( (s \land \neg o) \rightarrow \neg r \)
  “If you are short and not old then you may not ride.”

- Note: the English doesn’t say, but might be assumed to imply, “Otherwise you may ride”,
  i.e., \( (s \land \neg o) \leftrightarrow \neg r \)
Logical Puzzles. [Smullyan]

“Knights always tell the truth; Knaves always lie.”

Each of $A$ and $B$ is either a knight or a knave.

$A$ says, “$B$ is a Knight.”; $B$ says, “$A$ and $B$ are opposite types.”

Let $a = A$ is a Knight.; $b = B$ is a Knight.

Let $S_A = A$ says, “$B$ is a Knight.”
Let $S_B = B$ says, “$A$ and $B$ are opposite types.”

<table>
<thead>
<tr>
<th>$a$</th>
<th>$b$</th>
<th>$S_A$</th>
<th>$S_B$</th>
<th>$S_A \land S_B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Thus $A$ and $B$ are both knaves.
Welcome to CS250

Please make sure to do the following:

1. Buy text and iClicker2 from textbook store.

2. Find the moodle course page and register your iClicker.

3. Do the first reading assignment: R1 and take its associated quiz, by midnight 9/4/14.

4. Do the second reading assignment: R2, before L2.

5. Bring your registered iClicker to all lectures, from now on.

6. Take the R2 quiz by midnight 9/7/14