Lecture #2

Propositional Logic
"proposition": a declarative sentence that is either True (T/1) or False (F/0/⊥)

"bottom"/"falsum"/

"absurdum"
propositional variables (e.g., "p", "q").) stand in for propositions, and help us focus on the logic (rather than the propositions themselves).

We often prefer to use variables to refer to atomic propositions, which cannot be expressed in terms of simpler propositions.
we can qualify or combine propositions w/ logical operators:

- negation: \( \neg p \) — “not \( p \)”
- conjunction: \( p \land q \) — “\( p \) and \( q \)”
- disjunction: \( p \lor q \) — “\( p \) or \( q \)” (inclusive-or)
- exclusive or: \( p \oplus q \) — “\( p \) xor \( q \)”

in order of decreasing precedence: \( \neg p \land q \)
Disjunction is the typical programming language "or"

e.g. `def foo(x, y):
    if x < 0 or y < 0:
        raise Exception("no negative inputs")`

Exclusive or is often implied by the English "or"

e.g. "you can have cake or ice cream for dessert."
we use truth tables to show the value of a proposition for all combinations of values taken by its variables.

\[
\begin{array}{c|c}
P & \neg P \\ \hline
T & F \\
F & T 
\end{array}
\quad
\begin{array}{c|c|c}
P & q & P \lor q \\ \hline
T & T & T \\
T & F & T \\
F & T & T \\
F & F & F 
\end{array}
\quad
\begin{array}{c|c|c}
P & q & P \lor q \\ \hline
T & T & F \\
T & F & T \\
F & T & T \\
F & F & F 
\end{array}
\]
How many rows in a truth table for a proposition of $N$ variables?

- $N=2$: $2^2 = 4$
- $N=3$: $2^3 = 8$
- $N=4$: $2^4 = 16$

For example, a proposition with 2 variables would have 4 rows, a proposition with 3 variables would have 8 rows, and a proposition with 4 variables would have 16 rows.

$\text{e.g. } N=2 \quad \text{e.g. } N=3 \quad \text{e.g. } N=4$

$2^N$
Applying logic to English propositions:

\[ p = \text{"I love cats"} \]
\[ q = \text{"computer science is a science"} \]
\[ r = \text{"4 < 100"} \]

read: \[ p \land q \]
\[ \neg p \lor \neg q \]
\[ p \land (q \lor \neg r) \]
\[ p \land \neg p = F \lor T \]
\[ q \lor \neg q = T \]
A tautology is a proposition that is always true.

e.g. \( T \)
\[
\begin{align*}
& p \lor \neg p \\
& (p \land q) \lor \neg p \lor \neg q
\end{align*}
\]

A contradiction is a proposition that is always false.

e.g. \( F \)
\[
\begin{align*}
& \neg T \\
& p \land \neg p \\
& \neg (p \lor q) \lor \neg p \lor \neg q
\end{align*}
\]
How to prove \((p \land q) \lor \neg p \lor \neg q\) is a tautology?

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p \land q</th>
<th>\neg p</th>
<th>\neg q</th>
<th>\neg p \lor \neg q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

\((p \land q) \lor \neg p \lor \neg q\) is a tautology.
\( p \rightarrow q \) is a proposition known as a conditional statement (aka. implication)

read "if \( p \), then \( q \)"

hypothesis / antecedent

conclusion / consequent
caution: the logic conditional statement is NOT equivalent to the "if" statement in imperative programming!

e.g. if cond { not a proposition!
e.g. "if the AC is on, then I'll be cold."

$p = \text{the AC is on}$
$q = \text{I'll be cold}$

```
<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$P \rightarrow Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
</tbody>
</table>
```

"If I'm strong enough, then I can lift this weight!"

we could be cold for some other reason
If I'm strong enough then I can lift this weight.

I can lift this weight only if I'm strong enough.
Many other ways to express conditional in English:

Some tricky ones for $p \rightarrow q$:

"$p$ is sufficient for $q$"

"$p$ only if $q$"

"$q$ is necessary for $p$"

"$q$ unless $\neg p$"
Can we express ‘→’ using just ¬, ∧, ∨?

\[
\begin{array}{c|c|c|c|c|c|c|c}
 p & q & p \rightarrow q & T & T & F & F & T \\
 T & T & T & T & F & F & T & T \\
 T & F & F & T & F & F & T & T \\
 F & T & T & T & T & T & T & T \\
 F & F & T & T & F & F & T & T \\
\end{array}
\]

if \( T_p \) then \( p \rightarrow q \) is T

otherwise \( p \rightarrow q \) is equivalent \( q \)

so \( p \rightarrow q \equiv T_p \lor q \)