Logic 1/28/15

Propositional (sentential) logic

Formal language consists of proposition variables + logical connectives +
punctuation (parentheses, etc.)

Truth tables define the meaning of connectives (punctuation?) -"Truth-

Factual Semantics" - meanings defined by

conditions (in terms of truth of

props.) under which expressions

are true.

Truth tables look like grids of functions

(b/c maps values of prop variables to

values of expression?)

Material implication

<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>

Has to be defined this

Way in order to be

Unique.

Why \( \land, \lor, \neg, \rightarrow, \leftrightarrow \) (aren't \( \leftrightarrow \rightarrow \)

reducible to \( \land \lor \neg \) any way?)

Are there other possible connectives?
a new column = a unique column of truth values. So for 2 variables there are 16 possible connectives.

? There are more w/ more variables?

"hund" = T = "not born"  
+ "nor" = \( \land \) = "not or"

"we can think of \( \neg \) as a function of two arguments, we just don't use one of them."

\[ 
\begin{array}{c|c|c}
P & Q & P \lor Q \lor P \\
\hline
T & T & T \\
T & F & T \\
F & T & T \\
F & F & T \\
\end{array}
\]

\[ 
\begin{array}{c|c|c|c|c|c}
P & (P \rightarrow Q) \lor (Q \rightarrow P) \\
\hline
T & T & T & T \\
T & F & T & T \\
F & T & T & T \\
F & T & T & T \\
\end{array}
\]

All T's = a tautology!

\( \neg \) is \( P, \neg \psi \) are legit expressions, so are \((P \lor \psi), (P \land \psi), (P \iff \psi)\) etc.
A well-formed formula is any expression you can get combining propositional variables with connectives and right syntax (e.g., syntax vs. semantics (truth-tables))

(put parentheses around all \( \land + \lor \) expressions)

We don't really need (.) and \( \land \) expression\( \land \) is associative, but this is semantics, addressed from a syntax perspective we don't know that?

For every \( \wedge \) there is a "construction history" (tree):

\[
((P \wedge Q) \land R) \lor ((P \wedge Q) \rightarrow R) \\
\wedge \\
((P \wedge Q) \land R) \\
\wedge \\
(P \wedge Q) \rightarrow R \\
\wedge \\
P \wedge Q \\
P \wedge Q
\]

(greek letters for complex (maybe) expressions)
This construction enables (\( \wedge \)).

The argument method: Induction on formulas. If all prep variables have (smallest): a common property and this property is preserved by the four building operations, then all formulas have that property.

\[
( (A \wedge B) \wedge (C \Rightarrow D) ) \quad \text{Infix}
\]

\[
\Rightarrow \wedge \wedge \wedge \wedge \quad \text{Prefix}
\]

You can just add parentheses and have it still be well formed, even though it doesn't make sense from a semantic perspective.

Tautology testing

Truth tables not always feasible - e.g.: is \( ((A \wedge B) \Rightarrow C) \Rightarrow ((A \wedge \neg A) \Rightarrow (C \vee (B \wedge \neg A)) \Rightarrow C) \) a tautology?

1) whenever \( C \) is true, tautology

is true, all consequent will be true.
2) So what if C is false? Then in order for the consequent to turn out true, and the antecedent false, we'd need (A \lor B) to be true \land A \lor B \land A to be true, A \lor A \lor A \lor B \land A \land A \lor B \land A \implies impossible. So it's a tautology.

Logical equivalence:
P is logically eq. to \psi iff \
same truth tables. \underline{Same as:} P is 
logically equiv. to \psi iff \Leftrightarrow \psi is 
a tautology.

If we have n variables, how 
many diff. possible truth tables 
concerning (binary?) (2^n)

\text{Disjunctive Normal Form:}

Every truth function with any 
number of variables can be expressed using 
just \neg, \land, \lor. \neg \text{ can be expressed } 
as a disjunction of \lor \text{ of literals (either a prop. variable } 
or the negation of a prop. variable). 
\emph{s.e. can be expressed in Disjunctive } 
\emph{Normal form.}
Proof: Every row in a truth-table is expressible as a conjunction clause exactly detailing the truth or falsity of each \( P_i \):

\[
P_1, P_2, P_3, \ldots, P_n
\]

"conjunctive normal form" - conjunction of disjunctions of literals

A complete set of connectives:

A set of connectives is complete if we can express any truth-functional expressing them.

\[
P \rightarrow Q = \neg (P \land \neg Q) = \neg P \lor Q
\]

\[
P \rightarrow Q = (P \land Q) \lor (\neg P \land \neg Q)
\]

\[
\therefore \forall V, \Lambda, \neg \exists \text{ is complete}
\]

\[
P \land Q = \neg (\neg P \lor \neg Q) \quad \text{(De Morgan's law)}
\]

\[
\therefore \forall V, \Lambda \exists \text{ is complete}
\]

\[
P \lor Q = \neg (\neg P \land \neg Q) \quad \text{ (De Morgan's law)}
\]

\[
\therefore \exists V, \Lambda \exists \text{ is complete}
\]
Is there a single binary truth function that's complete? Yes! Nand!
\[ p \rightarrow q = \neg (p \land q) \]
\[ p \uparrow p = \neg p \]
\[ \neg (p \rightarrow q) = p \land q = (A \lor B) \uparrow (A \land B) \]
Also, nor!
\[ \neg A = A \uparrow A \]
\[ A \uparrow B = \neg (A \land B) = (A \lor B) \uparrow (A \land B) \]

A formula \( \phi \) is satisfiable if

some row of its truth table is \( t \)
(thus a possible world in which its true)

\[
\begin{array}{c|c|c|c|c|c}
 p & q & r & p \lor q & r \rightarrow q & \text{etc...} \\
 t & t & t & t & t & T \\
 t & t & f & f & t & T \\
 f & f & t & f & t & T \\
 f & f & f & t & f & T \\
\end{array}
\]

a world

Satisfiability problem: How can we determine if \( \phi \) is satisfiable?

(w/out making + checking a truth table) \( \Rightarrow \) no feasible algorithm known!

\( ? \)

P vs. NP problem; its easy to tell whether \( \phi \) is satisfiable if you run with \( \phi \) to look at... ?