1 Preliminaries.

Goals: Motivate propositional logic syntax and inferencing. Feel comfortable manipulating logical symbols so that, when we add complexity to the logical systems with quantifiers, etc., you’ll still have these basic natural deduction rules and strategies in your arsenal.

A language contains an infinite number of sentences; we’re able to understand sentences that we haven’t heard before and their meanings corresponding to thoughts that may be entirely new to us. Moreover, we’re able to draw inferences from the thoughts to new thoughts. In order to model this kind of understanding and reasoning, we need a way to build sentence meanings from their component parts.

Recall Frege’s Compositionality Principle:

**The Principle of Compositionality:** The meaning of a complex expression is determined by the meaning of its parts and the way those parts are combined.

This motivates our “to do list”: we begin, in this handout, to consider how thoughts corresponding to complex sentences (actually, propositions) may be composed from thoughts corresponding to their simpler propositional parts.

- A sentence can be true or false in a given situation or circumstance.
  
  (1) The pope talked to Prince Williams between 3 and 4 pm on Feb. 5, 2005.
- Although we may not know what the facts are, we know what they ought to be in order to judge the sentence true (i.e., truth conditions).
- Conversely, even if we know what the facts are, we cannot use these facts to evaluate whether the sentence is true, if we do not understand what the sentence means.

Thus, the truth condition is a necessary component of sentence meaning, although it may not be a sufficient component.

We begin by considering a very simple interpreted formal language - P(ropositional) L(ogic), and a reasoning system called Natural Deduction in this language. This language has a truth-conditional semantics, and can serve as the first attempt at natural language semantics. In describing PL, we’ll also introduce the items corresponding to PL connectives in natural language. We will examine:

i. Their meaning / denotation.

ii. The way they combine with the rest of the sentence.
2 Syntax of Propositional Logic (PL).

• What is a proposition? Anything that can be true or false.

(2) Sophia sometimes dances Argentine tango.

This whole thing is one proposition. Clearly, that hides a lot of the internal structure of the sentence that we will want to get at.

That’s where Predicate Logic (First Order Logic) comes in, but, for now, were content to hide all of the internal stuff and just call this entire sentence $p$.

• What are logical connectives? They hold atomic sentences like $p$ and $q$ together to make complex sentences.

We basically know what each connective means, but that doesn’t matter at all for natural deduction. So, while I will introduce a truth-conditional semantics for PL, this will be only to motivate the rules of inferencing in N(atural) D(eduction). We will ignore it when doing proofs in ND.

Natural deduction is just matching symbols and pushing symbols around! It doesn’t matter in the slightest what the propositions or connectives stand for.

Our connectives:
- Conditional: $\rightarrow$
- Negation: $\neg$
- And: $\land$ (also &)
- Or: $\lor$

• What is the main connective? (a.k.a. What kind of sentence is it?) The connective that holds the whole sentence together.
  - Conditional: $\phi \rightarrow \psi$
  - Negation: $\neg \phi$
  - Conjunction: $\phi \land \psi$ (or else $\phi \& \psi$)
  - Disjunction: $\phi \lor \psi$
  - Conditional: $(\phi \land \psi) \rightarrow \neg (\rho \rightarrow (\psi \lor \alpha))$

Finding the main connective is like asking, which syntactic rule was applied last in building up the formula of PL? This is important both in calculating the truth-value of the whole formula, and also in drawing inferences from the formula using ND.

• The set of legitimate statements (formulas) in PL can be defined recursively like this:

(3) **Lexical entries**: the letters $p, q, r, s, \ldots$ representing atomic statements.
(4) The set of all formulas in PL.
   a. Any atomic statement—represented with the letters $p, q, r, s, \ldots$—is a formula in PL.
   b. If $\phi$ is a formula in PL, then $\neg\phi$ is a formula in PL too. It reads "It is not the case
      that $\phi$”
   c. If $\phi$ and $\psi$ are formulae in PL, then
      $(\phi \land \psi)$ (reads "$\phi$ and $\psi$")
      $(\phi \lor \psi)$ (reads "$\phi$ or $\psi$")
      $(\phi \rightarrow \psi)$ (reads "if $\phi$ then $\psi$")
      are also formulae in PL
   d. Nothing else is a formula in PL.

QUESTION 1: Which of the following expressions are formulae in PL?

i. $\neg\neg p$
ii. $(\neg p)$
iii. $\neg(p)$
iv. $p \land q \lor r$
vi. $pq$
v. $p \land q \land r$
vii. $(p \lor q)$
viii. $(p \land q) \rightarrow \neg r$
ix. $\neg(p \land q)$

(5) Syntactic structure of $\neg(p \land q)$:

```
   ┌──┐
   │   │
   │   └──┐
   │      │
   │      └──┐
   │         │
   │         └───┐
   │                 │
   │                 └───┐
   │                     │
   │                     └───┐
   │                           │
   │                           └───┐
   │                                   │
   │                                   └───┐
   │                                           │
   │                                           └───┐
   │                                               │
   │                                               └───┐
   │                                                   │
   │                                                   └───┐
   │                                                       │
   │                                                       └───┐
   │                                                           │
   │                                                           └───┐
   │                                                               │
   │                                                               └───┐
   │                                                                   │
   │                                                                   └───┐
   │                                                                       │
   │                                                                       └───┐
   │                                                                                     │
   │                                                                                     └───┐
   │                                                                                                           │
   │                                                                                                           └───┐
   │                                                                                                               │
   │                                                                                                               └───┐
   │                                                                                                                   │
   │                                                                                                                   └───┐
   │                                                                                                                       │
   │                                                                                                                       └───┐
   │                                                                                                                               │
   │                                                                                                                               └───┐
   │                                                                                                                                   │
   │                                                                                                                                   └───┐
   │                                                                                                                                       │
   │                                                                                                                                       └───┐
   │                                                                                                                                            │
   │                                                                                                                                            └───┐
   │                                                                                                                                                │
   │                                                                                                                                                └───┐
   │                                                                                                                                                    │
   │                                                                                                                                                    └───┐
   │                                                                                                                                                                                  │
   │                                                                                                                                                                                  └───┐
   │                                                                                                                                                    ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ··
• ψ is a logical consequence of φ
  iff, for every w such that \([φ]^w = 1\), we have \([ψ]^w = 1\).
  iff φ → ψ is a tautology

• Two formulas φ and ψ are logically equivalent iff, for every w, \([φ]^w = [ψ]^w\)

If something holds in all circumstances, we can omit w and just write \([φ]\) which reads "the truth-value of φ".

Semantic value of the PL connectives: For any PL formulae φ and ψ,

(7) Negation: \([¬φ] = 1\) iff \([φ] = 0\). 

<table>
<thead>
<tr>
<th>φ</th>
<th>¬φ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

(8) Conjunction: \([φ ∧ ψ] = 1\) iff \([φ] = 1\) and \([ψ] = 1\).

<table>
<thead>
<tr>
<th>φ</th>
<th>ψ</th>
<th>φ ∧ ψ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

(9) Disjunction: \([φ ∨ ψ] = 1\) iff \([φ] = 1\) or \([ψ] = 1\) (or both).

<table>
<thead>
<tr>
<th>φ</th>
<th>ψ</th>
<th>φ ∨ ψ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

(10) Conditional: \([φ → ψ] = 0\) iff \([φ] = 1\) and \([ψ] = 0\).

<table>
<thead>
<tr>
<th>φ</th>
<th>ψ</th>
<th>φ → ψ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

• Translations from natural language (English) into PL:

QUESTION 3: Translate the following sentences into PL: (mostly from GAMUT I)

i. This engine is noisy and it uses a lot of energy.

ii. Joan or Mary came.

iii. If Peter and Susan come, I will be upset.

iv. It is not the case that I will be upset if you don’t come.

v. It is not the case that Cain is guilty and Abel is not.

vi. John is not only stupid but also nasty.

vii. Johnny saw Santa and the Easter Rabbit last year, but I saw neither.

viii. Charles goes to work by car, or by bike and train.
ix. John will come only if Peter comes.

x. Charles comes if Elsa comes, and the other way around.

xi. If father and mother both go, then I won’t, but if only father goes, then I will go.

Deriving the semantic value of a complex formula compositionally:
the truth value of each node depends on the truth value of its daughter nodes and on the semantic
rule corresponding to the syntactic rule through which we combined the daughters.

(11) If Peter and Susan come, I will be upset.
KEY: p=Peter comes; q=Susan comes; r=I will be upset.

\[
\begin{array}{c}
(\neg (p \land q) \rightarrow r) \\
(4c, \rightarrow)
\end{array}
\]

\[
\begin{array}{c}
(4c, \lor) \\
(p \land q) \\
p \\
q \\
r
\end{array}
\]

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
<th>(p\land q)</th>
<th>((p\land q)\rightarrow r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

QUESTION 4: Draw the syntactic trees and construct the truth tables for the compositional
semantic interpretation of (12) and (13).

(12) It is not the case that I will be upset if you don’t come.
KEY: p= I will be upset; q= you come

(13) If father and mother both go, then I won’t, but if only father goes, then I will go.
KEY: p=father comes; q=mother comes; r=I will go

References

- Jessica Moszkowicz’s lecture notes in Ling 130: Semantics, 1/22/2010, Brandeis University