1 Introduction

- Idea: A Grammar is a system of rules that defines (i) what is acceptable, and (ii) what is not acceptable, in a language, L
  → When humans acquire language, they acquire this system of rules

- The phonological component of the grammar determines what sounds and combinations of sounds acceptable in L
  - inventories (phonetic, phonological, syllable, ...)
  - phonological rewrite rules

- The lexical and morphological component of the grammar determines the morphemes and combinations of morphemes (i.e., words) acceptable in L
  - the lexicon, lexical categories, thematic grids
  - morphological rules (i.e., morphological c-selectional restrictions)

- The syntactic component of the grammar determines the phrases and combinations of phrases (i.e., sentences) acceptable in L
  - lexical phrases, functional phrases
  - phrase structure rules (PSRs), transformational rules
  - the theta-criterion (i.e., syntactic c-selectional restrictions)

- The rules are restrictive
  (only forms that follow these rules are grammatical)...
  ...but also productive (all forms that follow these rules are grammatical)

- This conceptualization of language can model the design feature:

  Hockett's Design Feature: PRODUCTIVITY
  Language-users can create and understand novel/original utterances
  e.g., you can produce and understand utterances that you have never heard before, like “The jaguar with the heart-shaped pattern was very vain of her shiny coat."

- The generative rules allow us to create novel (grammatical) utterances
- ...but why can we understand these novel utterances?

  The semantic component of the grammar determines what meanings and combinations of meanings can be produced/conveyed in L

  1. lexical semantics: The meaning of words
  2. formal semantics: The meaning of sentences

- Idea: Just as there are phonological rules, morphological rules, and syntactic rules, there are semantic rules that determine the meaning of sentences

- Semantic Questions:
  What is the range of meaning for a sentence in L?
  What semantic rules determine this?

- Syntax Questions:
  What is the range of structures for a sentence in L?
  What syntactic rules determine this?

- Syntax-Semantics Interface Questions:
  How is the meaning of a sentence related to its structure?
2 Formalizing the Meaning of Sentences

• Natural languages are really complex! We’ll start by looking at the syntax-semantics interface of simpler systems
  → **PROPOSITIONAL LOGIC** and **PREDICATE LOGIC**

• A **LOGIC** is a language created to investigate methods of reasoning

2.1 PROPOSITIONAL LOGIC (PL)/STATEMENT LOGIC

• Propositional Logic is extremely simple, because abstracts away from the meaning of atomic/simple sentences - i.e., it treats simple sentences as its basic units

• **PROPOSITIONAL LOGIC** provides two set or rules
  1. **Syntactic Rules**: How to form grammatical statements in PL
  2. **Semantic/Interpretive Rules**: How to interpret non-atomic statements in PL

• **PL Ontology**: PL Statements consist of

  1. Atomic Propositions/Statements \{p, q, r, s, \ldots\}
  2. Truth-Values \{1, 0\}
  3. Propositional Operators \{\neg\}
  4. Propositional Connectives \{\land, \lor, \rightarrow, \leftrightarrow\}

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name</th>
<th>“Meaning”</th>
</tr>
</thead>
<tbody>
<tr>
<td>\neg</td>
<td>negation</td>
<td>‘It is not the case that...’</td>
</tr>
<tr>
<td>\land</td>
<td>conjunction</td>
<td>‘(Both)...and...’</td>
</tr>
<tr>
<td>\lor</td>
<td>disjunction</td>
<td>‘(Either) ...or ...’</td>
</tr>
<tr>
<td>\rightarrow</td>
<td>conditional</td>
<td>‘If...then...’</td>
</tr>
<tr>
<td>\leftrightarrow</td>
<td>biconditional</td>
<td>‘...if and only if...’</td>
</tr>
</tbody>
</table>

• **Syntactic Rules of PL**
  1. Every atomic proposition \(p, q, r,...\) is a statement
  2. If \(\phi\) is a statement, then \(\neg\phi\) is a statement
  3. If \(\phi\) and \(\psi\) are statements, then \((\phi \land \psi)\) is a statement
  4. If \(\phi\) and \(\psi\) are statements, then \((\phi \lor \psi)\) is a statement
  5. If \(\phi\) and \(\psi\) are statements, then \((\phi \rightarrow \psi)\) is a statement
  6. If \(\phi\) and \(\psi\) are statements, then \((\phi \leftrightarrow \psi)\) is a statement

• Technically, whenever you use a connective to combine two statements, the resulting statement should be enclosed in parentheses...
  ...in practice, parentheses are often left off the outermost statement eg., \(p \land q\) is written instead of \((p \land q)\), and \((p \land q) \land r\) is written instead of \(((p \land q) \land r)\) (but *\(p \land q \land r\))

• **Semantic Rules of PL**:
  All of the semantic rules of PL can be represented by these **TRUTH-TABLES**
  Each **TRUTH-TABLE** lists the **TRUTH-CONDITIONS** for a complex statement

\[
\begin{array}{c|c|c|c|c}
 p & \neg p & p & q & (p \land q) & p & q & (p \lor q) \\
\hline
 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c}
 p & q & (p \rightarrow q) & p & q & (p \leftrightarrow q) \\
\hline
 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 0 & 0 & 1 & 0 & 0 \\
 0 & 1 & 1 & 0 & 1 & 0 \\
 0 & 0 & 1 & 0 & 0 & 1 \\
\end{array}
\]
(1) a. Rapunzel has long hair
   b. The witch has a long nose
   c. Rapunzel doesn’t have long hair
   d. Rapunzel has long hair and the witch has a long nose
   e. Rapunzel has long hair or the witch has a long nose
   f. If Rapunzel has long hair, then the witch has a long nose
   g. Rapunzel has long hair if the witch has a long nose

   • Notice that the semantic rules of propositional logic are directly tied to the syntactic rules; for each syntactic rule (except 1), there is a correlating semantic rule about how to interpret the structure.

   Student Exercises: The Syntax of Propositional Logic

   1. Try to rewrite the syntactic rules of PL into the form of PSRs (don’t worry about parentheses, eg., \( \phi \to \psi \land \mu \))

   2. Based on the syntactic rules of PL, indicate whether the statements in (2) are grammatical statements of PL, or not (*)

   hint: If you can draw a tree for the statement, where each node follows a syntactic rule, then the statement is a grammatical PL statement.

   3. Correct the ungrammatical statements of PL in (2) by adding brackets, propositional statements, operators, connectives, etc.

(2) Assume atomic propositions \( \{p, q, r\} \)

   a. \( p \land q \)
   b. \( p \land q \lor r \)
   c. \( (p \land q) \lor r \)
   d. \( \neg r \land p \)
   e. \( (p \to (p \land q)) \)
   f. \( (p \land q) \leftrightarrow q \)
   g. \( (p \lor q) \leftrightarrow q \)
   h. \( \land q \to (p \lor q) \)
   i. \( (p \lor r) \leftrightarrow \neg q \)

   • You can use the basic truth-tables to calculate the truth-conditions of complex PL statements - to do this, construct a **COMPLEX TRUTH-TABLE**

   1. See how many distinct atomic propositions are in the statement, eg.,
   \[ (p \land \neg q) \] has two,
   \[ (p \land \neg q) \to p \] has two,
   \[ (p \land \neg q) \to r \] has three...

   2. Create columns for each of these atomic propositions; list all of the possible combinations of truth-values in each row, eg.,

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

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

   3. **Option 1**: In a final column, write out the complex statement

   **Option 2**: For each sub-statement inside the complex statement (minus the atomic statements), add a column, plus a final column for the complete complex statement.
<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>((p ∧ ¬q) → p)</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>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>¬q</th>
<th>(p ∧ q)</th>
<th>((p ∧ q) → p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

4. **Option 2:** Using the basic truth-tables, for each row, calculate the truth-conditions for each column, eg., under ¬q you list the opposite truth-value from the q column, under the (p ∧ ¬q), you list 1 in the rows where both p and ¬q are 1 (and 0 otherwise), etc., under the ((p ∧ ¬q) → p), you list 1 in the rows where both arguments are 1, and in the rows where the first argument (p ∧ ¬q) is 0

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>¬q</th>
<th>(p ∧ ¬q)</th>
<th>((p ∧ ¬q) → p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Option 1:** If you are using this option, you do the same thing, but list the relevant truth-tables under the connective - eg., under ¬ for the first column; under the ∧ for the second column, and under the → for the final column

**Syntax-Semantics Interface Questions:**
How is the meaning of a PL sentence related to its structure?
Is the meaning of a PL sentence restricted by its structure?

Is the structure of a PL sentence restricted by its meaning? By what rules?

- **Some useful TERMINOLOGY**
  - **TAUTOLOGY:** A proposition is **tautologous** when it is necessarily true (i.e., 1 in all of the truth-conditional rows)
    - its truth-value does not depend on the circumstances
    eg., for PL, ((p ∧ ¬q) → p)
  - **CONTRADICTION:** A proposition is **contradictory** when it is necessarily false (i.e., 0 in all of the truth-conditional rows)
    - its truth-value does not depend on the circumstances
    eg., for PL, ((p ∧ ¬q) → q)
  - **CONTINGENCY:** A proposition is **contingent** when its truth-value depends on the circumstances - it can be either true or false
  - **EQUIVALENCE:** Two propositions are **logically equivalent** when they have the same truth-conditions (i.e., same pattern in the final column)

- Note that in PL, **tautologies** and **contradictions** depend solely on their form - the content of the atomic propositions (i.e., whether they are true or false) do not affect their truth-conditions at all

- This is a very strong connection between syntax and semantics

**Student Exercises:** The Semantics of Propositional Logic

1. Calculate the truth-conditions for the PL statements in (2) by creating a complex truth-table (including your corrected statements). If p, q = 1 and r = 0, what is the truth-value of each PL statement?
2. Identify whether any of the statements are tautologies, contradictions or contingent propositions.
### Logical entailments/valid arguments - i.e., rules of inference

#### Modus Ponens

<table>
<thead>
<tr>
<th>Hypothetical Syllogism</th>
<th>Simplification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ( P \rightarrow Q )</td>
<td>1. ( P \land Q )</td>
</tr>
<tr>
<td>2. ( P )</td>
<td>2. ( \therefore P )</td>
</tr>
<tr>
<td>3. ( \therefore Q )</td>
<td>3. ( \therefore P \rightarrow R )</td>
</tr>
</tbody>
</table>

#### Modus Tollens

<table>
<thead>
<tr>
<th>Disjunctive Syllogism</th>
<th>Addition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ( P \rightarrow Q )</td>
<td>1. ( P )</td>
</tr>
<tr>
<td>2. ( \neg Q )</td>
<td>2. ( \therefore P )</td>
</tr>
<tr>
<td>3. ( \therefore \neg P )</td>
<td>3. ( \therefore Q )</td>
</tr>
</tbody>
</table>

### Hypothetical Syllogism

1. \( P \rightarrow Q \)
2. \( P \)
3. \( \therefore Q \)

### Disjunctive Syllogism

1. \( P \rightarrow Q \)
2. \( Q \)
3. \( \therefore P \rightarrow R \)

### Simplification

1. \( P \land Q \)
2. \( \therefore P \)

### Conjunction

1. \( P \land Q \)
2. \( \therefore P \)
3. \( \therefore Q \)

### Addition

1. \( P \lor Q \)
2. \( \therefore P \lor Q \)

### More practice (...there may or may not be a pop quiz):

- Magnus 2014, pg.33 (chapter 2, sentential logic = propositional logic), Practice Exercises A, B, C, G (note: wff stands for “well-formed formula,” meaning a grammatical sentence in PL), pg.44, Practice Exercises A, B, C

### 2.2 PREDICATE LOGIC/QUANTIFIED LOGIC

#### 1st Order Predicate Logic:

**Statements consist of a combination of**

1. **Predicates:**
   - (i) one-place \( P(x) \)
   - (ii) two-place \( P(x, y) \)
   - (iii) ...
   - (iv) \( n \)-place \( P(x, y, ..., n) \)

2. **Terms:**
   - (i) **INDIVIDUAL CONSTANTS** \{a, b, c, d, e, f, ... \}
     - Denote particular individuals - yield statements when combined with predicate
   - (ii) **INDIVIDUAL VARIABLES** \{w, x, y, z\}
     - Place-holders - combine with predicates to yield open statements or propositional functions

**Syntactic Rules of 1st-Order Predicate Logic**

1. If \( a_1 \) and \( a_2 \) are individual terms, then \( a_1 = a_2 \) is a statement
2. If \( P \) is a \( n \)-place predicate, and \( t_1, t_2, ..., t_n \) are terms, then \( P(t_1, t_2, ..., t_n) \) is a statement
   - If \( t_1, t_2, ..., t_n \) are all **INDIVIDUAL CONSTANTS**, this statement is also a **PROPOSITION**
   - If one or more of \( t_1, t_2, ..., t_n \) are **INDIVIDUAL VARIABLES**, this is an open statement (not a proposition)
3. If \( \phi \) is a statement, and \( x \) is a variable, \( \forall x[\phi] \) and \( \exists x[\phi] \) are statements
4. If \( \phi \) is a statement, \( \neg \phi \) is a statement
5. If \( \phi \) and \( \psi \) are statements, \( \phi \land \psi \), \( \phi \lor \psi \), \( \phi \rightarrow \psi \), \( \phi \leftrightarrow \psi \) are statements

**Variables need to be BOUND by QUANTIFIERS to yield a proposition - i.e., something that can be interpreted by the semantics and given a truth-value**

- There are two quantifiers in predicate logic:
  1. **The Universal Quantifier**
     \( \forall x[P(x)] \) For all \( x \), \( P(x) \)
     (“All \( x \)-es are \( P \)”)
  2. **The Existential Quantifier**
     \( \exists x[P(x)] \) There exists some \( x \), such that \( P(x) \)
     (“An \( x \) is \( P \)”)

5
In Predicate Logic, a quantifier attaches to an open proposition (i.e., as an operator like \( \neg \)) - this proposition is its (semantic) **scope**.

A quantifier can only bind variables in its scope.

(3) **Definition:** **scope**

An occurrence of a variable is **bound** if it occurs in the scope of \( \forall x \) or \( \exists x \).

A variable is **free** if it is not bound.

- \( \exists x [P(x)] \)
- \( \exists x [P(x) \land Q(x,y)] \)
- \( \exists x [P(x)] \land Q(x,y) \)
- \( \forall x [P(x) \rightarrow Q(x,y)] \)

The propositional operators and connectives from propositional logic combine with predicate logic statements to create complex statements:

(4) **Complex Quantificationally Statements**

- \( \forall x [P(x) \rightarrow Q(x)] \) Forall x, if P(x), then Q(x)
- \( \exists x [P(x) \land Q(x)] \) There exists an x such that P(x) and Q(x)

**Note:** The particular syntactic rules provided above allow for statements with *unbound* variables and *vacuously quantified* variables, eg.,

\( \forall x P(a) \)

Vacuous Quantification

\( P(y) \)

Unbound Variable

Here we will rule these out as semantically uninterpretable, but other versions of predicate logic rule these out syntactically.

**Semantic Rules of 1st-Order Predicate Logic**

1. \( P(x) = 1 \) iff \( x \in P \)
   (True, if and only if x is a member of the set P)
2. \( P(x,y, ..., n) = 1 \) iff \( \langle x, y, ..., n \rangle \in P \)
   (True, if and only if the tuple \( \langle x, y, ..., n \rangle \) is a member of the set P)
3. \( \forall x [P(x)] = 1 \) iff \( x \in P \) for all individual values of x
   (True, if and only if, all individuals, x, are members of P)
4. \( \exists x [P(x)] = 1 \) iff \( x \in P \) for at least one individual value of x
   (True, if and only if, at least one individual, x, is a member of P)
5. \( \forall x[P(x) \rightarrow Q(x)] = 1 \) iff \( P \subseteq Q \)
   (True, if and only if P is a subset of Q)
6. \( \exists x [P(x) \land Q(x)] = 1 \) iff \( P \cap Q \neq \emptyset \)
   (True, if and only if the intersection of P and Q is not empty)
7. The semantics of the negative operator and binary connectives are (the same) as defined in propositional logic (i.e., the truth-tables)

**Rules of inference, for predicate logic**

- **Modus Ponens**
  1. \( \forall x[P(x) \rightarrow Q(x)] \) \( P \rightarrow Q \)
  2. \( P(a) \) \( P \)
  3. \( \therefore Q(a) \) \( \therefore Q \)

- **Modus Tollens**
  1. \( \forall x[P(x) \rightarrow Q(x)] \) \( P \rightarrow Q \)
  2. \( \neg Q(a) \) \( \neg Q \)
  3. \( \therefore \neg P(a) \) \( \therefore \neg P \)

- ...
Student Exercise: Predicate Logic

(5) Given the following symbolization key, translate the truth-conditions of the English-language sentences into predicate logic.

H(x): x attends Hogwarts  a: Harry
G(x): x is a Gryffindor  b: Ron
B(x): x is brave  c: Hermione
S(x): x is a Slytherin.  d: Neville

a. Harry, Hermione and Ron all attend Hogwarts.
b. Some Gryffindors are not brave.
c. Neville is a Gryffindor, but he’s not brave.
d. If Hermione loves Ron, then Ron is brave.
e. If both Ron and Hermione are Gryffindors, then Ron and Hermione are both brave.
f. There are brave Gryffindors.
g. If Harry is not a Slytherin, then he’s a Gryffindor.

Student Exercise: Predicate Logic

(6) Determine whether the last statement is entailed by the truth of the preceding statement(s).

1. \( \forall x [P(x) \rightarrow Q(x)] \)
2. \( \{x: M(x)\} \subseteq \{x: P(x)\} \)
3. \( \forall x [M(x) \rightarrow Q(x)] \)

1. \( Q(a) \)
2. \( \forall x [Q(x) \rightarrow P(x)] \)
3. \( P(a) \)

1. \( \forall x [P(x) \rightarrow Q(x)] \)
2. \( \exists x [Q(x)] \)
3. \( \exists x [P(x)] \)

1. \( \forall x [\neg P(x)] \)
2. \( \neg \exists x [P(x)] \)

1. \( \exists x [P(x) \land Q(x)] \)
2. \( P(b) \)
3. \( Q(b) \)

• More practice (...there may or may not be a pop quiz):
  Magnus 2014, pg.76 (chapter4), Practice Exercises A, C, E, H,
2.3 SUMMARY: The Syntax-Semantics of Logic

- **Summary**: We looked at the syntax-semantics of **propositional logic** and **predicate logic**.
- ...but what we really want, is a semantic system for **natural language**, and preferably, a system that relates to the **syntax** that we learned last term.
- Both propositional logic and predicate logic are commonly used as **metalanguages** by formal semantics - you will have an easier time if you become familiar with both!

3 PSRs and Compositionality

- **Review**: The Syntax-Semantics of PSRs
- PSRs/Constituents have **systematic, compositional** interpretations

  (i) **Nouns and AdjPs** consistently refer to **sets of individuals**

  (ii) **N’** consistently refers to the **set of individuals** that belong to both **AdjP** and **N** (i.e., \([\text{AdjP}] \cap [\text{N'}]\))

  \[ [X'] = [YP] \cap [X'] \text{ is the **compositional rule** (modification)} \]

  This describes how we interpret the PSR: **XP → YP \text{mod} x’**

  (iii) **NPs** consistently refer to **individuals**...

  ...an individual contained within the set \([\text{N’}]\)

  \[ [\text{NP}] = x, x \in [\text{N’}] \text{ is a **compositional rule** (selection)} \]

  This describes how we interpret the PSR: **NP → D \text{spec} N’**

- (iv) **VPs/V’s** also consistently refer to **sets of individuals**

- (v) The truth conditions of sentences \((S)\) are also consistent:

  \[ S \equiv \text{iff } h \in \{e, h\} \]

  \[ S \equiv \text{iff } b \in \{b, f\} \]

- This works for a very basic X’-Theoretic syntax...but what about **extended X’-Theory**? Can this simple semantic system be extended to account for **functional projections**?

- **Next Week**: Syntax Review (+ Transformational Rules)

References

Magnus, PD. 2014. forall x: An introduction to formal logic.