Section 6.1 Symbols and Translation

Terms to learn:

- operators/connectives
- simple and compound statements
- main operator

The Operator Chart

<table>
<thead>
<tr>
<th>Operator</th>
<th>Name</th>
<th>Logical Function</th>
<th>Used to Translate</th>
</tr>
</thead>
<tbody>
<tr>
<td>~</td>
<td>tilde</td>
<td>negation</td>
<td>not, it is false that, it is not the case that</td>
</tr>
<tr>
<td></td>
<td>dot</td>
<td>conjunction</td>
<td>and, also, but, moreover, however, nevertheless, still, both, additionally, furthermore</td>
</tr>
<tr>
<td>∨</td>
<td>wedge</td>
<td>disjunction</td>
<td>or, unless</td>
</tr>
<tr>
<td>⊃</td>
<td>horseshoe</td>
<td>implication</td>
<td>if...then..., only if, given that, provided that, in case, on condition, that, sufficient condition for, necessary condition for</td>
</tr>
<tr>
<td>≡</td>
<td>triple bar</td>
<td>equivalence</td>
<td>if and only if, is a necessary and sufficient condition for</td>
</tr>
</tbody>
</table>

Tips for translation:

- Remember that translation from ordinary English to logical expressions often results in a "distortion of meaning" because not all ordinary statements can be adequately expressed in logical terms.

- Be sure you can locate the main operator in a statement. Parenthesis can point us to the main operator. If you are not sure which operator functions as the main operator, please reread this section.
• There are five types of operators:
  1. **negation**
  2. **conjunction** or conjunctive statements
  3. **disjunction** or disjunctive statements
  4. **conditionals** that express *material implication* (antecedent and consequent)
  5. **biconditionals** that express *material equivalence*

• **Conditional statements do not always translate in an intuitive manner.** Sometimes it will be necessary to switch the order of the subject and predicate terms to get an accurate translation.
  - **Necessary and sufficient conditions:** be sure you know the differences highlighted
  - *When translating a conditional proposition, place the sufficient condition in the antecedent position and the necessary condition in the consequent position.*

• **Biconditionals** are used to translate the phrase "is a necessary and sufficient condition for" and should not be confused with conditional statements. In order to be translated as a biconditional, the proposition in question must serve as both a necessary and sufficient condition.

• When there are more than two simple propositions (i.e., two letters), it is necessary to group phrases together for an accurate translation. Look for clues given by commas, semicolons, and grouping phrases like "either," "both," etc.

• The object of this exercise is to create **well-formed formulas** (WWFs) that are syntactically accurate translations.
6.2 Truth Functions

- **Terms you should know:**
  - statement variables
  - statement forms
  - truth table

- **Hints for understanding truth tables for the operators:**
  - **Conjunction:** the only time this operator evaluates "true" is when both conjuncts are true.
  - **Disjunction:** the only time this operator evaluates "false" is when both disjuncts are false.
    - INCLUSIVE vs. EXCLUSIVE uses of "or": the general approach is to
  - **Conditional:** the only time this operator evaluates "false" is when the antecedent is true and consequent is false. Please read the "intuitive" approach for understanding why this operator only evaluates false in one case.
  - **Biconditional:** the only time this operator evaluates "true" is "when its two components have the same truth value."
    - The biconditional formula is a shorthand way of expressing these two conditionals: \((p \supset q) \land (q \supset p)\)

- **Process for solving longer propositions**
  - Please pay close attention to the order for evaluating longer propositions.
  - These problems are solved from the inside out just like equations with parenthesis from Algebra 1. Work with the simplest expressions inside parenthesis (letter truth values) and precede outward towards the main operator.

- **Real World Applications**
  - Meaning can sometimes be distorted when we translate expressions into logical terms. Hurley reminds us that we should not use logical operators to
translate statements that depend on "subtleties of expression."

- Note the case of inclusive and exclusive disjunction using the word "unless."

- There is a sharp distinction between the truth functional interpretation of a conditional and any inferential connection expressed in a conditional statement. In ordinary language we usually interpret the conditional statement as expressing an inferential connection between antecedent and consequent. If the truth functional interpretation of a particular statement conflicts with the ordinary interpretation, it is unwise to translate the statement using this operator.

- Subjunctive/counterfactual conditionals are evaluated based on the inferential connection between the antecedent and consequent. Thus, they are not good candidates for logical translation using the horseshoe operator; this also holds for subjunctive biconditionals.
What are truth tables and why do we use them in logic?

- Truth tables "give the truth value of a compound proposition for every possible truth value of its simple components. Each line in the truth table represents one such possible arrangement of truth values."

- We use truth tables to analyze compound propositions and arguments. For compound propositions we classify arguments based on their truth table results as either tautologous, self-contradictory or contingent. When we compare statements (analyze full arguments) we classify them as either logically equivalent, contradictory, consistent or inconsistent.

What is the process for setting up a truth table?

1. To calculate the number of lines in a truth table, use this simple formula: \( L = 2^n \). \( L \) is the number of lines and the small "\( n \)" is a variable representing the number of simple propositions contained in the statement. Simple propositions are expressions represented by a single letter.

2. Symbolize the statement you are evaluating.

3. Determine the number of lines necessary using the formula presented above.

4. Divide the total number of lines in half and assign the truth value "T" to the lines beneath the first simple proposition and assign the value "F" to the remaining lines.

5. For each successive simple proposition, divide the number of "T" values in half again and assign the value "T" to the first half and "F" to the second half. This process sounds far more complicated than it actually is. To see the process in action follow through the sample arguments that follow.

6. After truth values have been assigned to each simple proposition, evaluate the proposition working from inside the parenthesis to the outer layers.

7. When you evaluate the column under the main operator, you will be asked to classify your argument in accordance with the categories below.

<table>
<thead>
<tr>
<th>Statement Classification</th>
<th>Column Under Main Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>tautologous (logically true)</td>
<td>all true</td>
</tr>
<tr>
<td>self-contradictory (logically false)</td>
<td>all false</td>
</tr>
<tr>
<td>contingent</td>
<td>at least one true, at least one false</td>
</tr>
</tbody>
</table>
See Hurley’s explanation covering the classification of propositions. In short, we first try to classify a proposition as either logically equivalent or contradictory and then, if it does not meet that criteria, the categories of consistent and inconsistent apply.

**Real World Applications:**

Arguments that are logically consistent can be said to “make sense” because “there will be at least one line in the group of truth tables where all of the person’s statements are true.” Arguments that are contradictory or inconsistent indicate that there is not an instance where both the premises and conclusion are true.

### 6.4 Truth Tables for Arguments

The process for testing the validity of arguments with truth table is detailed above. The process is the same as that for testing individual propositions, except we are now testing multiple propositions strung together. **Thus, we will be looking for lines on which all premises are true and the conclusion false signaling the argument is invalid. If no such line exists, then the argument is valid.**
### 6.5 Indirect Truth Tables

Indirect truth tables provide a shortcut method for testing argument validity. This exercise is a review of the skills we learned in Section 1.5 where the method of counterexample was introduced. In order to set up this shortcut method, we have to consider all possible situations in which all premises could be true and the conclusion false.

**Testing arguments for validity:**

The process for testing arguments for validity is described in the section. The method is:

1. Look at the conclusion and determine the ways in which the main operator could evaluate "false."

2. Assign truth values of false to the conclusion and true for the premises for each situation in which the main operator in the conclusion could evaluate false.

3. Working backwards from the truth values generated by evaluating the conclusion false, deduce the truth values for the remaining simple propositions and premises.

4. **If you do not encounter a contradiction** while assigning truth values to the premises, then it is possible for all premises to be true and the conclusion false. Thus, the argument is invalid.

5. **If a contradiction is produced** in the attempts to assign truth values to the premises, circle the contradiction and declare the argument valid.

6. For **arguments with multiple ways of evaluating false:**
   - you must generate a contradiction on every line for the argument to be valid.
   - You can stop evaluating on a single line when you produce a contradiction.
   - If you fail to produce a contradiction on any line, (i.e., your deductions produce true premises and a false conclusion), you can stop evaluating the argument. It is invalid.

Depending on the order of assignment of truth values in the premises, contradictions can be produced in multiple ways if an argument is valid. So it is possible that the solution presented in the text will be different from the one that you deduce. As long as you have evaluated the simple and complex propositions correctly, your solution is equally valid.

**Testing statements for consistency:**

The process for testing statements for consistency is similar to testing arguments for validity. The process differs from the above in the following ways:

1. Assign the value of true to the main operator of each statement.

2. Pick one of the statements and create a line for each situation in which the main
operator could evaluate true.

3. Deduce the remaining truth values on the line.

4. If no contradiction is produced, the truth values are consistent.

5. If a contradiction is produced, the statements are inconsistent.
6.6 Argument Forms and Fallacies

This section introduces common valid and invalid argument forms. Six valid and two invalid forms are introduced. It is possible to affirm validity/invalidity for any of these forms and I invite you examine each one of them via the truth table method. Without further delay, here are the forms:

Disjunctive syllogism (DS):

This argument form relies on the "method of elimination" for its validity. Note that one of the two disjuncts must be eliminated for this argument form to be valid. This is the case because "inclusive disjunction includes the possibility of both disjuncts being true."

\[ p \lor q \]
\[ \sim p \]
\[ \therefore q \]

Pure Hypothetical Syllogism (HS):

A hypothetical syllogism is a chain of conditional statements such that the consequent of one premise is the antecedent of the remaining premise. Not every argument with three conditional statements is a hypothetical syllogism.

\[ p \supset q \]
\[ q \supset r \]
\[ \therefore p \supset r \]

Modus Ponens (MP):

Modus ponens is dubbed "asserting mode" because it affirms the antecedent in the second premise and the consequent in the conclusion.

\[ p \supset q \]
\[ p \]
\[ \therefore q \]
Modus Tollens (MT):

Modus tollens is dubbed "denying mode" because it denies the consequent in the second premise and the antecedent in the conclusion. See Hurley's explanation for the logic behind modus tollens.

\[ p \supset q \]

\[ \neg q \]

\[ \therefore \neg p \]

Be sure not to confuse the invalid forms of affirming the consequent and denying the antecedent with MP and MT. Although they look like their valid cousins, these latter two forms are invalid and can be proven so by using truth tables or counterexamples.

Constructive Dilemma (CD):

Be sure to study this form closely as many of the exercises in the text and on our exams will mimic this form, but not match it. Hence it is wise to note that the second premise contains the antecedents of each conditional statement and the conclusion contains each consequent.

\[ (p \supset q) \land (r \supset s) \]

\[ p \lor r \]

\[ \therefore q \lor s \]

Destructive Dilemma (DD):

Again, a caution to study this form closely as many of the exercises in the text and on our exams will mimic this form, but not match it. Hence it is wise to note that the second premise denies the consequents of each conditional statement and the conclusion denies the antecedents.

\[ (p \supset q) \land (r \supset s) \]

\[ \neg q \lor \neg s \]

\[ \therefore \neg p \lor \neg r \]
Refuting Constructive and Destructive Dilemmas:

1. There are two direct methods for refuting CD and DD arguments. The first is labeled "grasping by the horns" and it requires us to prove that one of the two conditional statements in the first conjunctive premise is false. Remember conditional statements evaluate false if the antecedent is true and the consequent is false. The second technique is labeled "escaping between the horns" and in order to use it to prove invalidity, we must prove the disjunctive premise false.

2. An indirect method of refuting CD an DD arguments is to show that the arguer has not considered all possible scenarios in the conditional statements that comprise the conjunctive premise; depending on which of the two arguments you are attempting to refute, the disjunctive premise should remain unchanged and the remaining set of antecedents or consequents should be changed.

3. A method for recognizing argument forms:

4. These tips are taken from the last few pages of section 6.6.

5. Symbolize the ordinary language argument.

6. Compare the valid argument forms to the argument you have symbolized and look for patterns that would signal you have a valid form.

7. Remember, the statement $p \lor q$ is logically equivalent to $q \lor p$ and you may have to change the order of the disjunctive propositions to see the pattern.

8. Negated letters may also be substituted for $p$, $q$, $r$, and $s$.

9. The statement $p$ is logically equivalent to the statement $\sim\sim p$.

10. The order of premises does not affect the form of an argument. You may have to rearrange the premises to recognize the form.

Smartboard Notes from Chapter 6 Lectures:

Section 6.1:

Part 1 and 2 problems done in class:
Part 1: Homework
30. \((I \cdot B) \vee M\)
31. \((M \cdot T) \vee (\neg C \cdot \neg A)\)  \(\Rightarrow \neg (B \cdot V)\)
32. \((P \cdot T) \cdot N\)
33. \((W \cdot M) \cdot \neg (W \cdot M)\)
34. \(\neg M \Rightarrow I \Rightarrow (D \cdot V)\)

39. \([B \cdot (M \cdot V) \cdot H]\)
36. \(\neg [D \vee (G \cdot H)]\)
37. \(S \Rightarrow [D \Rightarrow (P \cdot M)]\)
38. \(H \Rightarrow [S \Rightarrow (B \Rightarrow \epsilon)]\)
39. \(\neg [(B \cdot L) \vee (L \cdot H)]\)

40. \(\neg [(M \cdot V) \cdot (T \cdot V)\)]\)

P2
1. \(R \cdot V \cdot F\)  5. \(\neg (O \cdot Y)\)
2. \(P \Rightarrow S\)  (\(\neg O \cdot V \cdot Y\))
3. \(H \cdot \epsilon\)  (\(N \cdot A) \equiv (R \cdot C)\)
4. \(\neg (H \cdot V \cdot S)\)  \(\equiv \neg H \cdot \neg S\)
Section 6.2:

Text Example:

6.2 Part 1 Problems:
Homework from Section 6.2:
Smartboard Notes for Sections 6.3 & 6.4:

**Truth Tables (6.3 & 6.4):**

- \( \text{# of lines} \rightarrow 2^n \)
- \( n = \# \text{ of different variables/letters} \)
- 2 letters: \( 2^2 = 4 \) lines
- 3 letters: \( 2^3 = 8 \) lines
- 4 letters: \( 2^4 = 16 \) lines

---

<table>
<thead>
<tr>
<th>pg.</th>
<th>#</th>
<th>((C \lor D) \land E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>TFFT</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>TFFT</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>TTTF</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>TTTT</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>pg.</th>
<th>Line #</th>
<th>((A \lor B) \land B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>TFFT</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>TTTF</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>TTTF</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>FTTF</td>
</tr>
</tbody>
</table>

---

**Section 6.3 text example:**

**Section 6.3 text example: comparing two statements.**
Section 6.4 text example

Section 6.4 text example

Section 6.4 Part II Exercises #18 p. 323
Homework Review Week 12 Sections 6.3 & 6.4

Section 6.3 Homework Part I #6

\[ \neg A \equiv X \wedge (X \wedge A) \wedge (A \land \neg X) \]

Line #
1  F F T T F F T T
2  F F T T F F T T
3  T T T T T T T T
4  F F T T F F T T

Section 6.3 Homework Part II #5

\[ H \equiv \neg \neg H \wedge (G \vee H) \wedge (\neg G \land \neg H) \]

Line #
1  T F F F T F F F
2  T F F F T F F F
3  F F T T F F T T
4  F F T T F F T T

Section 6.3 Homework Part II #6

\[ (Q \wedge P) \vee (Q \wedge R) \wedge (P \wedge R) \]

Line #
1  T T T F T F T T
2  T T T F T F T F
3  T T T F T F T T
4  T T T F T F T F
5  T T T F T F T T
6  T T T F T F T F
7  T T T F T F T T
8  F F F F F F F F
### Section 6.4 Homework #5

<table>
<thead>
<tr>
<th>Line</th>
<th>Expression</th>
<th>Truth Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$k = n$</td>
<td>T</td>
</tr>
<tr>
<td>2</td>
<td>$\neg (L \cdot n)$</td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>$k &lt; L$</td>
<td>T</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>invalid</td>
</tr>
</tbody>
</table>

---

### 6.4 Homework #9

<table>
<thead>
<tr>
<th>Line</th>
<th>Expression</th>
<th>Truth Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$A = (B \lor C) \land C \lor B$</td>
<td>T</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>T</td>
</tr>
</tbody>
</table>

**Note:** The table contains various entries, including `T` (true), `F` (false), and `invalid`.
Smartboard Notes from Sections 6.5 & 6.6

Section 6.5 Text example

Section 6.5 Text example #2

6.5 Text example

6.6 Forms

Disjunctive Syllogism

\[ \frac{p \lor q}{\sim p} \quad \frac{\sim q}{\sim M} \]

\[ \frac{q}{\sim T} \]
Hypothetical Syllogism
\[ \frac{W = C}{C = P} \]
\[ \therefore W = P \]

\[ \frac{\neg W = C}{\neg W = C} \]
\[ \therefore \neg C = W \]

Modus Ponens (MP)
\[ p = q \]
\[ \therefore \neg p, \neg q \]

\[ \therefore \neg q \]

\[ \frac{T = S}{T = S} \]
\[ \therefore I \]
\[ \therefore S \]
\[ \therefore \neg S \]
\[ \therefore \neg C \]

INVALID variations on MP + MT
1. Affirming the consequent
\[ p = q \]
\[ \therefore q \]
\[ \therefore \neg p \]

2. Denying the antecedent
\[ p = q \]
\[ \therefore \neg q \]
\[ \therefore \neg p \]
**Constructive Dilemma**

\[(p \rightarrow q) \cdot (s \rightarrow t)
\]
\[p \lor r
\]
\[q \lor s
\]
\[\therefore q \lor r
\]

\[(N = I) \cdot (\neg D A)
\]
\[\neg N \lor C
\]
\[\therefore \neg I \lor A
\]

**Destructive Dilemma**

\[(p \rightarrow q) \cdot (s \rightarrow t)
\]
\[\neg q \lor r \lor s
\]
\[\therefore \neg q \lor r \lor s
\]

\[(R > N) \cdot (L = C)
\]
\[\neg N \lor r \lor c
\]
\[\therefore \neg r \lor r \lor l
\]