Propositional Logic, day 1

Philosophy 180
May 20, 2014

\[ \sim (P \cdot Q) \]
\[ \sim P \quad Q \]
\[ \quad P \]

\[ \sim (P \cdot Q) \]
\[ (P \cdot Q) \]
\[ P \quad Q \]
Outlines

The Language $PL$
- Syntax for $PL$
- Semantics for $PL$

Translation from $PL$ to English
- A Translation Guide
- Canonical Logical Form

Translations from English to $PL$
- Translating into Canonical Logical Form
OUTLINE

THE LANGUAGE PL

Syntax for PL
Semantics for PL

TRANSLATION FROM PL TO ENGLISH

A Translation Guide
Canonical Logical Form

TRANSLATIONS FROM ENGLISH TO PL

Translating into Canonical Logical Form
The Plan

- Construct an artificial language, ‘PL’, within which we can be incredibly precise about which arguments are deductively valid and which are deductively invalid.
Construct an artificial language, ‘PL’, within which we can be incredibly precise about which arguments are deductively valid and which are deductively invalid.

Provide a method for translating statements of English into PL and statements of PL into English.
Construct an artificial language, ‘PL’, within which we can be incredibly precise about which arguments are deductively valid and which are deductively invalid.

Provide a method for translating statements of English into PL and statements of PL into English.

One advantage: we can theorize about relations of deductive validity without having to worry about the ambiguity of English (e.g., equivocation and amphiboly).
Languages in General

- To specify a language, we need to provide:
LANGUAGES IN GENERAL

▶ To specify a language, we need to provide:
  ▶ a vocabulary for the language
Languages in General

- To specify a language, we need to provide:
  - a vocabulary for the language
    - e.g., a list of the words of English and their parts of speech
Languages in General

- To specify a language, we need to provide:
  - a vocabulary for the language
    - e.g., a list of the words of English and their parts of speech
  - a grammar for the language
Languages in General

▶ To specify a language, we need to provide:
  ▶ a vocabulary for the language
    ▶ *e.g.*, a list of the words of English and their parts of speech
  ▶ a grammar for the language
    ▶ *e.g.*, rules for saying which strings of words are grammatical and which aren’t
**Languages in General**

- To specify a language, we need to provide:
  - a vocabulary for the language
    - *e.g.*, a list of the words of English and their parts of speech
  - a grammar for the language
    - *e.g.*, rules for saying which strings of words are grammatical and which aren’t
      - ‘Bubbie makes pickles’ ✓
LANGUAGES IN GENERAL

- To specify a language, we need to provide:
  - a vocabulary for the language
    - *e.g.*, a list of the words of English and their parts of speech
  - a grammar for the language
    - *e.g.*, rules for saying which strings of words are grammatical and which aren’t
      - ‘Bubbie makes pickles’ ✓
      - ‘Up bouncy ball door John variously catapult’ ×
Languages in General

▶ To specify a language, we need to provide:
  ▶ a vocabulary for the language
    ▶ e.g., a list of the words of English and their parts of speech
  ▶ a grammar for the language
    ▶ e.g., rules for saying which strings of words are grammatical and which aren’t
    ‘Bubbie makes pickles’ ✓
    ‘Up bouncy ball door John variously catapult’ ✗
  ▶ a way to interpret the meaning of every grammatical expression of the language
Languages in General

- To specify a language, we need to provide:
  - a vocabulary for the language
    - *e.g.*, a list of the words of English and their parts of speech
  - a grammar for the language
    - *e.g.*, rules for saying which strings of words are grammatical and which aren’t
      - ‘Bubbie makes pickles’ ✓
      - ‘Up bouncy ball door John variously catapult’ ×
  - a way to interpret the meaning of every grammatical expression of the language
    - *e.g.*, a dictionary entry for every word of English and rules for constructing the meaning of sentences out of the meanings of words
LANGUAGES IN GENERAL

SYNTAX —— { 1. Vocabulary
2. Grammar

SEMANTICS —— 3. Meaning
OUTLINE

THE LANGUAGE PL
Syntax for PL
Semantics for PL

TRANSLATION FROM PL TO ENGLISH
A Translation Guide
Canonical Logical Form

TRANSLATIONS FROM ENGLISH TO PL
Translating into Canonical Logical Form
The Vocabulary of PL

- The vocabulary of PL includes the following symbols:
The Vocabulary of PL

- The vocabulary of PL includes the following symbols:
  1. An infinite number of statement letters:
     \[ A, B, C, \ldots, Y, Z, A_1, B_1, C_1, \ldots, Y_1, Z_1, A_2, B_2, C_2, \ldots \]
The Vocabulary of PL

- The vocabulary of PL includes the following symbols:
  1. An infinite number of statement letters:

\[ A, B, C, ..., Y, Z, A_1, B_1, C_1, ..., Y_1, Z_1, A_2, B_2, C_2, ... \]

  2. Logical operators:

\[ \sim, \cdot, \lor, \exists, \equiv \]
The Vocabulary of PL

- The vocabulary of PL includes the following symbols:
  1. An infinite number of statement letters:
     \( A, B, C, \ldots, Y, Z, A_1, B_1, C_1, \ldots, Y_1, Z_1, A_2, B_2, C_2, \ldots \)
  2. Logical operators:
     \( \neg, \cdot, \lor, \sqsupset, \equiv \)
  3. Parentheses
     \( (, ) \)
The Vocabulary of PL

- The vocabulary of PL includes the following symbols:
  1. An infinite number of statement letters:
     \[ A, B, C, \ldots, Y, Z, A_1, B_1, C_1, \ldots, Y_1, Z_1, A_2, B_2, C_2, \ldots \]
  2. Logical operators:
     \[ \sim, \bullet, \lor, \exists, \equiv \]
  3. Parentheses
     \[ (, ) \]

- Nothing else is included in the vocabulary of PL.
THE GRAMMAR OF PL

- Any sequence of the symbols in the vocabulary of PL is a formula of PL.
The Grammar of PL

- Any sequence of the symbols in the vocabulary of PL is a formula of PL.

\[((())A_{23} \quad \cdot \quad \supset \supset Z \\]
\[P \supset (Q) \supset \cdot (()) \]
\[(P \supset (Q \supset (R \supset (S \supset T)))) \]
\[A \cdot B \cdot (C \sim D)) \]
Any sequence of the symbols in the vocabulary of PL is a formula of PL.

\[((()A_{23} \cdot \cdot \supset \supset Z \supset (Q) \supset \cdot (\cdot (P \supset (Q \supset (R \supset (S \supset T))))))\]
\[A \cdot B \cdot (C \sim D)))\]
The Grammar of PL

- We define a well-formed formula (‘wff’) of PL with the following rules.
THE GRAMMAR OF PL

- We define a well-formed formula (‘wff’) of PL with the following rules.
  
  SL) Any statement letter, by itself, is a wff.
THE GRAMMAR OF PL

- We define a well-formed formula (‘wff’) of PL with the following rules.
  1. Any statement letter, by itself, is a wff.
  2. If ‘p’ is a wff, then ‘¬p’ is a wff.
The Grammar of PL

We define a well-formed formula (‘wff’) of PL with the following rules.

1. Any statement letter, by itself, is a wff.
2. If ‘p’ is a wff, then ‘¬p’ is a wff.
3. If ‘p’ and ‘q’ are wffs, then ‘(p • q)’ is a wff.
The Grammar of PL

- We define a well-formed formula (‘wff’) of PL with the following rules.

  SL) Any statement letter, by itself, is a wff.
  ~) If ‘p’ is a wff, then ‘~p’ is a wff.
  •) If ‘p’ and ‘q’ are wffs, then ‘(p • q)’ is a wff.
  ∨) If ‘p’ and ‘q’ are wffs, then ‘(p ∨ q)’ is a wff.

▶ ‘p’ and ‘q’ do not appear in the vocabulary of PL.

They are formulae variables—variables whose potential values are formulae of PL.
THE GRAMMAR OF PL

- We define a well-formed formula (‘wff’) of PL with the following rules.

SL) Any statement letter, by itself, is a wff.
~) If ‘p’ is a wff, then ‘~p’ is a wff.
•) If ‘p’ and ‘q’ are wffs, then ‘(p • q)’ is a wff.
∨) If ‘p’ and ‘q’ are wffs, then ‘(p ∨ q)’ is a wff.
⊃) ‘p’ and ‘q’ are wffs, then ‘(p ⊃ q)’ is a wff.
The Grammar of PL

We define a well-formed formula ('wff') of PL with the following rules.

SL) Any statement letter, by itself, is a wff.
    ∼) If ‘p’ is a wff, then ‘∼p’ is a wff.
     ▷) If ‘p’ and ‘q’ are wffs, then ‘(p • q)’ is a wff.
     ⊃) ‘p’ and ‘q’ are wffs, then ‘(p ⊃ q)’ is a wff.
     ≡) ‘p’ and ‘q’ are wffs, then ‘(p ≡ q)’ is a wff.
The Grammar of PL

- We define a well-formed formula (‘wff’) of PL with the following rules.

  SL) Any statement letter, by itself, is a wff.

  ~) If ‘p’ is a wff, then ‘~p’ is a wff.

  •) If ‘p’ and ‘q’ are wffs, then ‘(p • q)’ is a wff.

  ∨) If ‘p’ and ‘q’ are wffs, then ‘(p ∨ q)’ is a wff.

  ⊃) ‘p’ and ‘q’ are wffs, then ‘(p ⊃ q)’ is a wff.

  ≡) ‘p’ and ‘q’ are wffs, then ‘(p ≡ q)’ is a wff.

  −) Nothing else is a wff.
We define a well-formed formula (‘wff’) of PL with the following rules.

SL) Any statement letter, by itself, is a wff.
   ~) If ‘p’ is a wff, then ‘∼p’ is a wff.
   •) If ‘p’ and ‘q’ are wffs, then ‘(p • q)’ is a wff.
   ∨) If ‘p’ and ‘q’ are wffs, then ‘(p ∨ q)’ is a wff.
   ⊃) ‘p’ and ‘q’ are wffs, then ‘(p ⊃ q)’ is a wff.
   ≡) ‘p’ and ‘q’ are wffs, then ‘(p ≡ q)’ is a wff.
   −) Nothing else is a wff.

‘p’ and ‘q’ do not appear in the vocabulary of PL.
The Grammar of PL

» We define a well-formed formula (‘wff’) of PL with the following rules.

SL) Any statement letter, by itself, is a wff.
   ~) If ‘p’ is a wff, then ‘~p’ is a wff.

• ) If ‘p’ and ‘q’ are wffs, then ‘(p • q)’ is a wff.

∨) If ‘p’ and ‘q’ are wffs, then ‘(p ∨ q)’ is a wff.

⊃) ‘p’ and ‘q’ are wffs, then ‘(p ⊃ q)’ is a wff.

≡) ‘p’ and ‘q’ are wffs, then ‘(p ≡ q)’ is a wff.

¬) Nothing else is a wff.

» ‘p’ and ‘q’ do not appear in the vocabulary of PL.
   » They are formulae variables—variables whose potential values are formulae of PL.
THE GRAMMAR OF PL

- To show that ‘(\sim (P \lor Q) \supset R)’ is a wff:
THE LANGUAGE PL

To show that ‘(¬(P ∨ Q) ⊃ R)’ is a wff:

a) ‘P’ is a wff [from (SL)]
To show that ‘(\(\neg (P \lor Q) \supset R\))’ is a wff:

a) ‘\(P\)’ is a wff [from (SL)]

b) ‘\(Q\)’ is a wff [from (SL)]
The Grammar of PL

- To show that ‘(¬ (P ∨ Q) ⊃ R)’ is a wff:
  a) ‘P’ is a wff [from (SL)]
  b) ‘Q’ is a wff [from (SL)]
  c) So, ‘(P ∨ Q)’ is a wff [from (a) and (b) and (∨)]
To show that ‘(\(\sim (P \lor Q) \supset R\))’ is a wff:

a) ‘P’ is a wff [from (SL)]
b) ‘Q’ is a wff [from (SL)]
c) So, ‘(P \lor Q)’ is a wff [from (a) and (b) and (\(\lor\))]  
d) So, ‘\(\sim (P \lor Q)\)’ is a wff [from (c) and (\(\sim\))]
To show that ‘(\(\sim (P \lor Q) \supset R\))’ is a wff:

a) ‘P’ is a wff [from (SL)]
b) ‘Q’ is a wff [from (SL)]
c) So, ‘(P \lor Q)’ is a wff [from (a) and (b) and (\(\lor\))]d) So, ‘\(\sim (P \lor Q)\)’ is a wff [from (c) and (\(\sim\))]e) ‘R’ is a wff [from (SL)]
The Grammar of PL

- To show that ‘(~ (P ∨ Q) ⊃ R)’ is a wff:
  a) ‘P’ is a wff [from (SL)]
  b) ‘Q’ is a wff [from (SL)]
  c) So, ‘(P ∨ Q)’ is a wff [from (a) and (b) and (∨)]
  d) So, ‘~ (P ∨ Q)’ is a wff [from (c) and (~)]
  e) ‘R’ is a wff [from (SL)]
  f) So, ‘(~ (P ∨ Q) ⊃ R)’ is a wff [from (d), (e), and (⊃)]
To show that ‘(¬(P ∨ Q) ⊃ R)’ is a wff:

- a) ‘P’ is a wff  [from (SL)]
- b) ‘Q’ is a wff  [from (SL)]
- c) So, ‘(P ∨ Q)’ is a wff  [from (a) and (b) and (∨)]
- d) So, ‘¬(P ∨ Q)’ is a wff  [from (c) and (∼)]
- e) ‘R’ is a wff  [from (SL)]
- f) So, ‘(¬(P ∨ Q) ⊃ R)’ is a wff  [from (d), (e), and (⊃)]

A convention: we will allow ourselves to omit the outermost parentheses, writing ‘¬(P ∨ Q) ⊃ R’ instead of ‘(¬(P ∨ Q) ⊃ R)’
Main Operators

- A wff's *main operator* is just the operator associated with the last rule which would have to be applied if we were building the formula up by applying the rules for wffs.
Main Operators

▶ A wff’s main operator is just the operator associated with the last rule which would have to be applied if we were building the formula up by applying the rules for wffs.
▶ E.g., what is the main operator of ‘∼P • Q’?
A wff’s main operator is just the operator associated with the last rule which would have to be applied if we were building the formula up by applying the rules for wffs.

E.g., what is the main operator of ‘\( \sim P \cdot Q \)’?

a) ‘\( P \)’ is a wff [from (SL)]
Main Operators

- A wff's main operator is just the operator associated with the last rule which would have to be applied if we were building the formula up by applying the rules for wffs.

- E.g., what is the main operator of ‘∼P • Q’?
  - a) ‘P’ is a wff
  - b) So, ‘∼P’ is a wff

  [from (SL)]

  [from (a) and (∼)]
**Main Operators**

- A wff’s *main operator* is just the operator associated with the last rule which would have to be applied if we were building the formula up by applying the rules for wffs.

- E.g., what is the main operator of ‘\(\sim P \cdot Q\)’?
  
  a) ‘\(P\)’ is a wff [from (SL)]
  
  b) So, ‘\(\sim P\)’ is a wff [from (a) and (\(~\))] 
  
  c) ‘\(Q\)’ is a wff [from (SL)]
A wff’s main operator is just the operator associated with the last rule which would have to be applied if we were building the formula up by applying the rules for wffs.

E.g., what is the main operator of ‘∼ P • Q’?

- ‘P’ is a wff
- So, ‘∼ P’ is a wff
- ‘Q’ is a wff
- So, ‘(∼ P • Q)’ is a wff

[from (SL)]
[from (a) and (∼)]
[from (SL)]
[from (b), (c), and ( • )]
**Main Operators**

- A wff’s *main operator* is just the operator associated with the last rule which would have to be applied if we were building the formula up by applying the rules for wffs.
- E.g., what is the main operator of ‘∼P • Q’?
  
  a) ‘P’ is a wff [from (SL)]  
  b) So, ‘∼P’ is a wff [from (a) and (∼)]  
  c) ‘Q’ is a wff [from (SL)]  
  d) So, ‘(∼P • Q)’ is a wff [from (b), (c), and (•)]

- So, the main operator is ‘•’
Main Operators

- What if we had applied the rule (•) first, and then applied the rule (∼)?
What if we had applied the rule ( • ) first, and then applied the rule ( ~ )?

a) ‘P’ is a wff [from (SL)]
Main Operators

- What if we had applied the rule (•) first, and then applied the rule (~)?
  a) ‘P’ is a wff
  b) ‘Q’ is a wff

[from (SL)]

[from (SL)]
Main Operators

▶ What if we had applied the rule (•) first, and then applied the rule (~)?
   a) ‘P’ is a wff [from (SL)]
   b) ‘Q’ is a wff [from (SL)]
   c) So, ‘(P • Q)’ is a wff [from (a), (b), and (•)]
Main Operators

What if we had applied the rule \(( \cdot )\) first, and then applied the rule \((\sim)\)?

a) ‘\(P\)’ is a wff \[\text{[from (SL)]}\]
b) ‘\(Q\)’ is a wff \[\text{[from (SL)]}\]
c) So, ‘\((P \cdot Q)\)’ is a wff \[\text{[from (a), (b), and (\(\cdot\))]}\]
d) So, ‘\(\sim(P \cdot Q)\)’ is a wff \[\text{[from (c) and (\(\sim\))]}\]
What if we had applied the rule (\(\cdot\)) first, and then applied the rule (\(\sim\))? 

a) ‘\(P\)’ is a wff  
   b) ‘\(Q\)’ is a wff  
   c) So, ‘(\(P \cdot Q\))’ is a wff  
   d) So, ‘\(\sim (P \cdot Q)\)’ is a wff  

‘\(\sim (P \cdot Q)\)’ is not the same as ‘(\(\sim P \cdot Q\))’
**Subformulae**

- $p$ is a *subformula* of $q$ if and only if, in the course of building up $q$ by applying the rules for wffs, $p$ appears on a line before $q$. 
SUBFORMULAE

- $p$ is a subformula of $q$ if and only if, in the course of building up $q$ by applying the rules for wffs, $p$ appears on a line before $q$.
  - ‘~ $P$’ is a subformula of ‘~ $P \cdot Q$’
SUBFORMULAE

- $p$ is a subformula of $q$ if and only if, in the course of building up $q$ by applying the rules for wffs, $p$ appears on a line before $q$.
  - ‘$\neg P$’ is a subformula of ‘$\neg (P \cdot Q)$’
  - ‘$\neg P$’ is not a subformula of ‘$\neg (P \cdot Q)$’
SUBFORMULAE

- *p* is a subformula of *q* if and only if, in the course of building up *q* by applying the rules for wffs, *p* appears on a line before *q*.
  - ‘*∼ P*’ is a subformula of ‘*∼ P • Q*’
  - ‘*∼ P*’ is not a subformula of ‘*∼ (P • Q)*’

- A formula’s immediate subformulae are those wffs whose lines were appealed to in the final step of building to formula up.
SUBFORMULAE

- \( p \) is a subformula of \( q \) if and only if, in the course of building up \( q \) by applying the rules for wffs, \( p \) appears on a line before \( q \).
  - ‘\( \sim P \)’ is a subformula of ‘\( \sim P \cdot Q \)’
  - ‘\( \sim P \)’ is not a subformula of ‘\( \sim (P \cdot Q) \)’

- A formula’s immediate subformulae are those wffs whose lines were appealed to in the final step of building to formula up.
  - the immediate subformulae of ‘\( \sim P \cdot Q \)’ are ‘\( \sim P \)’ and ‘\( Q \)’
SUBFORMULAE

- $p$ is a subformula of $q$ if and only if, in the course of building up $q$ by applying the rules for wffs, $p$ appears on a line before $q$.
  - ‘$\sim P$’ is a subformula of ‘$\sim (P \cdot Q)$’
  - ‘$\sim P$’ is not a subformula of ‘$\sim (P \cdot Q)$’

- A formula’s immediate subformulae are those wffs whose lines were appealed to in the final step of building to formula up.
  - the immediate subformulae of ‘$\sim (P \cdot Q)$’ are ‘$\sim P$’ and ‘$Q$’
  - the immediate subformula of ‘$\sim (P \cdot Q)$’ is ‘$(P \cdot Q)$’
SUBFORMULAE

- $p$ is a subformula of $q$ if and only if, in the course of building up $q$ by applying the rules for wffs, $p$ appears on a line before $q$.
  - ‘$\sim P$’ is a subformula of ‘$\sim P \cdot Q$’
  - ‘$\sim P$’ is not a subformula of ‘$\sim (P \cdot Q)$’

- A formula’s immediate subformulae are those wffs whose lines were appealed to in the final step of building to formula up.
  - the immediate subformulae of ‘$\sim P \cdot Q$’ are ‘$\sim P$’ and ‘$Q$’
  - the immediate subformula of ‘$\sim (P \cdot Q)$’ is ‘$(P \cdot Q)$’

- Another way of notating proofs that a formulae of PL is well-formed: syntax trees
SYNTACTIC STRUCTURE

\[(\sim (P \lor Q) \supset R)\]

\[\supset\]

\[\sim (P \lor Q)\]

\[R\]

\[\sim\]

\[(SL)\]

\[(P \lor Q)\]

\[\lor\]

\[P\]

\[Q\]

\[(SL)\]

\[(SL)\]
Syntactic Structure

\[(\sim (P \lor Q) \supset R)\]

\[
\begin{array}{c}
\sim (P \lor Q) \\
\mid \\
(P \lor Q) \\
\end{array}
\]

\[
\begin{array}{c}
P \\
Q
\end{array}
\]
Syntactic Structure

\[
(\sim P \cdot Q)
\]

\[
\sim P \quad Q
\]

\[
(\sim (P \cdot Q))
\]

\[
(P \cdot Q)
\]

\[
P \quad Q
\]
OUTLINE

THE LANGUAGE PL
  Syntax for PL
  Semantics for PL

TRANSLATION FROM PL TO ENGLISH
  A Translation Guide
  Canonical Logical Form

TRANSLATIONS FROM ENGLISH TO PL
  Translating into Canonical Logical Form
**Meaning**

- Our guiding assumption: what it is to understand the meaning of an expression is just to understand the circumstances in which it is true
Our guiding assumption: what it is to understand the meaning of an expression is just to understand the circumstances in which it is true.

Parentheses just make syntactic structure explicit. They do not make any contribution to meaning beyond that.
Our guiding assumption: what it is to understand the meaning of an expression is just to understand the circumstances in which it is true.

Parentheses just make syntactic structure explicit. They do not make any contribution to meaning beyond that.

So: we must say what the meanings of the statements letters are and what the meanings of the logical operators are.
The Meaning of the Statement Letters

▶ Each statement letter represents a statement in English.
Each statement letter represents a statement in English.
The statement letter is true if and only if the statement in English is true.
THE MEANING OF ‘~’

▶ ‘~’ is called the tilde.
**The Meaning of ‘∼’**

- ‘∼’ is called the *tilde*.
- A wff whose main operator is the tilde is called a *negation*.
The Meaning of ‘∼’

- ‘∼’ is called the *tilde*.
- A wff whose main operator is the tilde is called a *negation*.
- Its immediate subformula is called the *negand*.
The Meaning of ‘∼’

- ‘∼’ is called the *tilde*.
- A wff whose main operator is the tilde is called a *negation*.
- Its immediate subformula is called the *negand*.

<table>
<thead>
<tr>
<th>p</th>
<th>∼p</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
THE MEANING OF ‘∼’

- ‘∼’ is called the tilde.
- A wff whose main operator is the tilde is called a negation.
- Its immediate subformula is called the negand.

\[
\begin{array}{c|c}
    p & ∼p \\
    \hline
    T & F \\
    F & T \\
\end{array}
\]

- Note: ‘p’ is not a wff of PL
The Meaning of ‘∼’

- ‘∼’ is called the tilde.
- A wff whose main operator is the tilde is called a negation.
- Its immediate subformula is called the negand.

<table>
<thead>
<tr>
<th>p</th>
<th>∼p</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

- Note: ‘p’ is not a wff of PL
  - we are using ‘p’ and ‘q’ as variables ranging over the wffs of PL
THE MEANING OF ‘•’

- ‘•’ is known as the *dot*. 
The Meaning of ‘•’

- ‘•’ is known as the dot.
- A wff whose main operator is the dot is known as a conjunction.
THE MEANING OF ‘●’

- ‘●’ is known as the *dot*.
- A wff whose main operator is the dot is known as a *conjunction*.
- Its immediate subformulae are called *conjuncts*.
THE MEANING OF ‘●’

- ‘●’ is known as the *dot*.
- A wff whose main operator is the dot is known as a *conjunction*.
- Its immediate subformulae are called *conjuncts*.

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p • 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>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
The Meaning of ‘∨’

- The operator ‘∨’ is known as the *wedge*. 
The Meaning of ‘∨’

- The operator ‘∨’ is known as the wedge.
- A wff whose main operator is the wedge is known as a disjunction.
The Meaning of ‘∨’

- The operator ‘∨’ is known as the wedge.
- A wff whose main operator is the wedge is known as a disjunction.
- Its immediate subformulae are called disjuncts.
The Meaning of ‘∨’

- The operator ‘∨’ is known as the wedge.
- A wff whose main operator is the wedge is known as a disjunction.
- Its immediate subformulae are called disjuncts.

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ∨ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
THE MEANING OF ‘◻’

▶ The operator ‘◻’ is known as the *horseshoe*. 
The Meaning of ‘⊃’

- The operator ‘⊃’ is known as the horseshoe.
- A wff whose main operator is the horseshoe is known as a material conditional.
THE MEANING OF ‘⇝’

- The operator ‘⇝’ is known as the *horseshoe*.
- A wff whose main operator is the horseshoe is known as a *material conditional*.
- The immediate subformulae which precedes the horseshoe is known as the *antecedent*. 
THE MEANING OF ‘⊃’

- The operator ‘⊃’ is known as the *horseshoe*.
- A wff whose main operator is the horseshoe is known as a *material conditional*.
- The immediate subformulae which precedes the horseshoe is known as the *antecedent*.
- The immediate subformulae which follows the horseshoe is known as the *consequent*.
## The Meaning of ‘⊃’

- The operator ‘⊃’ is known as the *horseshoe*.
- A wff whose main operator is the horseshoe is known as a *material conditional*.
- The immediate subformulae which precedes the horseshoe is known as the *antecedent*.
- The immediate subformulae which follows the horseshoe is known as the *consequent*.

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ⊃ 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>
The Meaning of ‘⊃’

- The operator ‘⊃’ is known as the horseshoe.
- A wff whose main operator is the horseshoe is known as a material conditional.
- The immediate subformulae which precedes the horseshoe is known as the antecedent.
- The immediate subformulae which follows the horseshoe is known as the consequent.

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ⊃ 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>

- Note: this is the only binary operator which is not symmetric.
The Meaning of ‘⊃’

- The operator ‘⊃’ is known as the *horseshoe*.
- A wff whose main operator is the horseshoe is known as a *material conditional*.
- The immediate subformulae which precedes the horseshoe is known as the *antecedent*.
- The immediate subformulae which follows the horseshoe is known as the *consequent*.

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ⊃ 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>

- Note: this is the only binary operator which is not symmetric.
  - ‘p ⊃ q’ does not have the same meaning as ‘q ⊃ p’
The Meaning of ‘≡’

- The operator ‘≡’ is known as the *triple bar*. 
The Meaning of ‘≡’

- The operator ‘≡’ is known as the *triple bar*.
- A wff whose main operator is the triple bar is known as a *material biconditional*. 
THE MEANING OF ‘≜’

- The operator ‘≜’ is known as the *triple bar*.
- A wff whose main operator is the triple bar is known as a *material biconditional*.
  - The immediate subformula which appears before the triple bar is known as the biconditional’s *left hand side*.
The Meaning of ‘≡’

- The operator ‘≡’ is known as the *triple bar*.
- A wff whose main operator is the triple bar is known as a *material biconditional*.
  - The immediate subformula which appears before the triple bar is known as the biconditional’s *left hand side*.
  - The immediate subformula which appears after the triple bar is known as the biconditional’s *right hand side*. 
The Meaning of ‘\( \equiv \)’

- The operator ‘\( \equiv \)’ is known as the *triple bar*.
- A wff whose main operator is the triple bar is known as a *material biconditional*.
  - The immediate subformula which appears before the triple bar is known as the biconditional’s *left hand side*.
  - The immediate subformula which appears after the triple bar is known as the biconditional’s *right hand side*.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>( p \equiv 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>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
Determining the Truth-value of a WFF of PL

- Suppose ‘P’ is true and ‘Q’ is false
Determining the Truth-value of a Wff of PL

- Suppose ‘P’ is true and ‘Q’ is false
- What are the truth-values of ‘~P • Q’ and ‘~(P • Q)’?
Determining the Truth-value of a wff of PL

- Suppose ‘P’ is true and ‘Q’ is false
- What are the truth-values of ‘¬ P • Q’ and ‘¬(P • Q)’?

\[
\begin{align*}
(\sim P \cdot Q) \\
\sim P \quad Q \\
P
\end{align*}
\]

\[
\begin{align*}
\sim (P \cdot Q) \\
\sim (P \cdot Q) \\
(P \cdot Q) \\
P \quad Q
\end{align*}
\]
Determining the Truth-value of a Wff of PL

- Suppose ‘P’ is true and ‘Q’ is false
- What are the truth-values of ‘∼ P • Q’ and ‘∼(P • Q)’?

\[
\begin{align*}
\sim P & \quad Q[F] \\
\sim (P • Q) & \\
P & \quad Q & \quad \sim (P • Q) \\
\sim (P • Q) & \\
P & \quad Q
\end{align*}
\]
Determining the Truth-value of a Wff of PL

- Suppose ‘P’ is true and ‘Q’ is false
- What are the truth-values of ‘\(\sim P \cdot Q\)’ and ‘\(\sim (P \cdot Q)\)’?

\[
\begin{align*}
(\sim P \cdot Q) & \\
\sim P & F \\
\sim P[F] & Q \ F \\
P[T] & \\
\end{align*}
\]

\[
\begin{align*}
\sim (P \cdot Q) & \\
\sim P & \\
(P \cdot Q) & \\
P & Q \\
\end{align*}
\]
DETERMINING THE TRUEH-VALUE OF A WFF OF PL

▶ Suppose ‘P’ is true and ‘Q’ is false
▶ What are the truth-values of ‘∼ P • Q’ and ‘∼(P • Q)’?

\[
\begin{align*}
& (∼P • Q)[F] \\
& ∼P[F] \quad Q[F] \\
& P[T] \\
\end{align*}
\]

\[
\begin{align*}
& ∼(P • Q) \\
& (P • Q) \\
& P \quad Q
\end{align*}
\]
Determining the Truth-value of a Wff of PL

- Suppose ‘P’ is true and ‘Q’ is false
- What are the truth-values of ‘\sim P \cdot Q’ and ‘\sim (P \cdot Q)’?

\[
\begin{align*}
(\sim P \cdot Q)[F] \\
\sim P[F] & \quad \sim Q[F] \\
P[T] & \\
\end{align*}
\]

\[
\begin{align*}
\sim (P \cdot Q) \\
\sim (P \cdot Q)[T] & \\
 & \\
(\sim P \cdot Q) & \\
\end{align*}
\]
Determining the Truth-value of a WFF of PL

- Suppose ‘P’ is true and ‘Q’ is false
- What are the truth-values of ‘\(\sim P \cdot Q\)’ and ‘\(\sim (P \cdot Q)\)’?

\[
\begin{align*}
(\sim P \cdot Q)[F] & \quad \sim (P \cdot Q) \\
\sim P[F] & \quad (P \cdot Q)[F] \\
P[T] & \quad P[T] \\
Q[F] & \quad Q[F]
\end{align*}
\]
Determining the Truth-value of a WFF of PL

- Suppose ‘P’ is true and ‘Q’ is false
- What are the truth-values of ‘¬ P \cdot Q’ and ‘¬(P \cdot Q)’?

\[
\begin{align*}
(\sim P \cdot Q)[F] & \quad \sim(P \cdot Q)[T] \\
\sim P[F] & \quad \sim(P \cdot Q)[F] \\
Q[F] & \\
P[T] & \quad (P \cdot Q)[F] \\
\end{align*}
\]
Truth Tables

- Truth-table for ‘\( \sim P \cdot Q \)’:
Truth Tables

- Truth-table for ‘\( \sim P \cdot Q \)’:

<table>
<thead>
<tr>
<th>( P )</th>
<th>( Q )</th>
<th>( \sim P )</th>
<th>( \cdot )</th>
<th>( Q )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Truth Tables**

- Truth-table for ‘\(\sim P \cdot Q\):'

<table>
<thead>
<tr>
<th>(P)</th>
<th>(Q)</th>
<th>(\sim P \cdot Q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(T)</td>
<td>(T)</td>
<td>(T)</td>
</tr>
<tr>
<td>(T)</td>
<td>(F)</td>
<td>(T)</td>
</tr>
<tr>
<td>(F)</td>
<td>(T)</td>
<td>(F)</td>
</tr>
<tr>
<td>(F)</td>
<td>(F)</td>
<td>(F)</td>
</tr>
</tbody>
</table>
# Truth Tables

- **Truth-table for ‘\( \sim P \cdot Q \):**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Q</td>
<td>( \sim )</td>
<td>P</td>
<td>( \cdot )</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>
Truth Tables

- Truth-table for ‘\( \sim P \cdot Q \)’:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( P )</td>
<td>( Q )</td>
<td>( \sim P )</td>
<td>( \cdot Q )</td>
</tr>
<tr>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
<td>( T )</td>
</tr>
<tr>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
<td>( T )</td>
</tr>
<tr>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
</tr>
<tr>
<td>( F )</td>
<td>( F )</td>
<td>( T )</td>
<td>( F )</td>
</tr>
</tbody>
</table>
## Truth Tables

- Truth-table for ‘\( \sim P \cdot Q \):'

<table>
<thead>
<tr>
<th>( P )</th>
<th>( Q )</th>
<th>( \sim P )</th>
<th>( \cdot Q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
</tr>
<tr>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
<td>( T )</td>
</tr>
<tr>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
</tr>
<tr>
<td>( F )</td>
<td>( F )</td>
<td>( T )</td>
<td>( F )</td>
</tr>
</tbody>
</table>
# Truth Tables

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$\sim P$</th>
<th>$\cdot$</th>
<th>$Q$</th>
</tr>
</thead>
<tbody>
<tr>
<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>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
</tbody>
</table>
**Truth Tables**

- Truth-table for ‘∼(P • Q)’: 

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>∼(P • Q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</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>
**Truth Tables**

- Truth-table for ‘\(\sim (P \cdot Q)\)’:

<table>
<thead>
<tr>
<th>(P)</th>
<th>(Q)</th>
<th>(\sim (P \cdot Q))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Truth Tables

- Truth-table for ‘∼ (P • Q)’:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Q</td>
<td>∼ (P • Q)</td>
</tr>
<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>
Truth Tables

- Truth-table for ‘∼(P • Q)’:

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>∼(P • Q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Truth Tables

- Truth-table for ‘\(\sim (P \cdot Q)\)’:

<table>
<thead>
<tr>
<th>(P)</th>
<th>(Q)</th>
<th>(\sim (P \cdot Q))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(T)</td>
<td>(T)</td>
<td>(T \ T \ T)</td>
</tr>
<tr>
<td>(T)</td>
<td>(F)</td>
<td>(T \ F \ F)</td>
</tr>
<tr>
<td>(F)</td>
<td>(T)</td>
<td>(F \ F \ T)</td>
</tr>
<tr>
<td>(F)</td>
<td>(F)</td>
<td>(F \ F \ F)</td>
</tr>
</tbody>
</table>
**Truth Tables**

- Truth-table for ‘\(~ (P \cdot Q)\)’:

<table>
<thead>
<tr>
<th></th>
<th>Q</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td></td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
OUTLINE

THE LANGUAGE *PL*
Syntax for *PL*
Semantics for *PL*

TRANSLATION FROM *PL TO ENGLISH*
A Translation Guide
Canonical Logical Form

TRANSLATIONS FROM ENGLISH TO *PL*
Translating into Canonical Logical Form
OUTLINE

THE LANGUAGE PL
- Syntax for PL
- Semantics for PL

TRANSLATION FROM PL TO ENGLISH
- A Translation Guide
  - Canonical Logical Form

TRANSLATIONS FROM ENGLISH TO PL
- Translating into Canonical Logical Form
A Translation Guide from PL to English

▶ Submitted for your approval:

\ ~ p \rightarrow \text{It is not the case that } p \\
p \cdot q \rightarrow \text{Both } p \text{ and } q \\
p \lor q \rightarrow \text{Either } p \text{ or } q \\
p \supset q \rightarrow \text{If } p, \text{ then } q \\
p \equiv q \rightarrow p \text{ if and only if } q
‘p ⊆ q’ AND ‘If p, then q’

- If ‘p’ is false, then ‘p ⊆ q’ is automatically true, no matter what statement q represents.
‘p ⊆ q’ AND ‘If p, then q’

- If ‘p’ is false, then ‘p ⊆ q’ is automatically true, no matter what statement q represents.
- But ‘if John Adams was America’s first president, then eating soap cures cancer’ doesn’t seem true.
‘p ⊆ q’ AND ‘If p, then q’

- If ‘p’ is false, then ‘p ⊆ q’ is automatically true, no matter what statement q represents
- But ‘if John Adams was America’s first president, then eating soap cures cancer’ doesn’t seem true.
- I plead guilty
‘\( p \supset q \)’ AND ‘If \( p \), then \( q \)’

- If ‘\( p \)’ is false, then ‘\( p \supset q \)’ is automatically true, *no matter what statement q represents*
- But ‘if John Adams was America’s first president, then eating soap cures cancer’ doesn’t seem true.
- I plead guilty
  - ‘\( p \supset q \)’ isn’t a perfect translation of ‘if \( p \), then \( q \)’
‘\( p \supset q \)’ AND ‘If \( p \), then \( q \)’

- If ‘\( p \)’ is false, then ‘\( p \supset q \)’ is automatically true, \textit{no matter what statement \( q \) represents}
- But ‘if John Adams was America’s first president, then eating soap cures cancer’ doesn’t seem true.
- I plead guilty
  - ‘\( p \supset q \)’ isn’t a perfect translation of ‘if \( p \), then \( q \)’
  - \textit{But}
'p ⊆ q' and 'If p, then q'

- If 'p' is false, then 'p ⊆ q' is automatically true, no matter what statement q represents.
- But 'if John Adams was America’s first president, then eating soap cures cancer' doesn’t seem true.
- I plead guilty
  - 'p ⊆ q' isn’t a perfect translation of 'if p, then q'
  - But it’s not as bad as you might think.
‘\(p \supset q\)’ AND ‘IF \(p\), THEN \(q\)’

- Dmitri: ‘If it’s a weekday, then I’m on campus.’
'p ⊆ q' AND 'if p, then q'

- Dmitri: 'If it’s a weekday, then I’m on campus.'
- Steve: ‘If I’m on campus, then it’s a weekday.'
'p ⊆ q' AND 'IF p, THEN q'

- Dmitri: 'If it’s a weekday, then I’m on campus.'
- Steve: ‘If I’m on campus, then it’s a weekday.’
  - D := ‘Dmitri is on campus’
'p ⊆ q' AND 'IF p, THEN q'

- Dmitri: ‘If it’s a weekday, then I’m on campus.’
- Steve: ‘If I’m on campus, then it’s a weekday.’
  - D := ‘Dmitri is on campus’
  - S := ‘Steve is on campus’
‘$p \supset q$’ AND ‘IF $p$, THEN $q$’

- Dmitri: ‘If it’s a weekday, then I’m on campus.’
- Steve: ‘If I’m on campus, then it’s a weekday.’
  - $D := ‘\text{Dmitri is on campus}’$
  - $S := ‘\text{Steve is on campus}’$
  - $W := ‘\text{It is a weekday}’$
'p ⊨ q' AND 'IF p, THEN q'

<table>
<thead>
<tr>
<th>D</th>
<th>W</th>
<th>if W, then D</th>
<th>W ⊨ D</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S</th>
<th>W</th>
<th>if S, then W</th>
<th>S ⊨ W</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
‘\( p \supset q \)’ AND ‘IF \( p \), THEN \( q \)’

<table>
<thead>
<tr>
<th>( D )</th>
<th>( W )</th>
<th>if ( W ), then ( D )</th>
<th>( W \supset D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( S )</th>
<th>( W )</th>
<th>if ( S ), then ( W )</th>
<th>( S \supset W )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
‘p ⊃ q’ AND ‘IF p, THEN q’

<table>
<thead>
<tr>
<th>D</th>
<th>W</th>
<th>if W, then D</th>
<th>W ⊃ D</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S</th>
<th>W</th>
<th>if S, then W</th>
<th>S ⊃ W</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
'\( p \supset q \)' AND 'IF \( p \), THEN \( q \)'

<table>
<thead>
<tr>
<th>( D )</th>
<th>( W )</th>
<th>if ( W ), then ( D )</th>
<th>( W \supset D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( S )</th>
<th>( W )</th>
<th>if ( S ), then ( W )</th>
<th>( S \supset W )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
'p ⊃ q' AND 'IF p, THEN q'

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>if W, then D</th>
<th>W ⊃ D</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>if S, then W</th>
<th>S ⊃ W</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
'\( p \supset q \)' AND 'IF \( p \), THEN \( q \)'

<table>
<thead>
<tr>
<th>( D )</th>
<th>( W )</th>
<th>if ( W ), then ( D )</th>
<th>( W \supset D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( S )</th>
<th>( W )</th>
<th>if ( S ), then ( W )</th>
<th>( S \supset W )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
‘\( p \supset q \)’ AND ‘IF \( p \), THEN \( q \)’

<table>
<thead>
<tr>
<th>( D )</th>
<th>( W )</th>
<th>if ( W ), then ( D )</th>
<th>( W \supset D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( S )</th>
<th>( W )</th>
<th>if ( S ), then ( W )</th>
<th>( S \supset W )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 'p ⊆ q' AND 'if p, then q'

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>if W, then D</th>
<th>W ⊆ D</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>if S, then W</th>
<th>S ⊆ W</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
'\( p \supset q \)' AND 'IF \( p \), THEN \( q \)'

<table>
<thead>
<tr>
<th>( D )</th>
<th>( W )</th>
<th>if ( W ), then ( D )</th>
<th>( W \supset D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn't lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn't lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn't lie</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( S )</th>
<th>( W )</th>
<th>if ( S ), then ( W )</th>
<th>( S \supset W )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
</tbody>
</table>
‘\( p \supset q \)’ AND ‘IF \( p \), THEN \( q \)’

<table>
<thead>
<tr>
<th>( D )</th>
<th>( W )</th>
<th>if ( W ), then ( D )</th>
<th>( W \supset D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( S )</th>
<th>( W )</th>
<th>if ( S ), then ( W )</th>
<th>( S \supset W )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
</tbody>
</table>
‘$p \supset q$’ AND ‘IF $p$, THEN $q$’

<table>
<thead>
<tr>
<th>$D$</th>
<th>$W$</th>
<th>if $W$, then $D$</th>
<th>$W \supset D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$S$</th>
<th>$W$</th>
<th>if $S$, then $W$</th>
<th>$S \supset W$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
</tbody>
</table>
‘*p ⊇ q’ AND ‘IF *p, THEN *q’

<table>
<thead>
<tr>
<th><em>D</em></th>
<th><em>W</em></th>
<th>if <em>W</em>, then <em>D</em></th>
<th><em>W</em> ⊇ <em>D</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><em>S</em></th>
<th><em>W</em></th>
<th>if <em>S</em>, then <em>W</em></th>
<th><em>S</em> ⊇ <em>W</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
</tbody>
</table>
‘\( \because \ p \supset q \)’ AND ‘IF \( p \) THEN \( q \)’

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>if ( W ), then ( D )</th>
<th>( W \supset D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>if ( S ), then ( W )</th>
<th>( S \supset W )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td>S</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>lied</td>
<td>S</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>didn’t lie</td>
<td>S</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td>S</td>
</tr>
</tbody>
</table>
‘\( p \supset q \)’ AND ‘IF \( p \), THEN \( q \)’

<table>
<thead>
<tr>
<th></th>
<th>( W )</th>
<th>if ( W ), then ( D )</th>
<th>( W \supset D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>( W )</th>
<th>if ( S ), then ( W )</th>
<th>( S \supset W )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>lied</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>didn’t lie</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td></td>
</tr>
</tbody>
</table>

---

**The Language PL**

**Translation from PL to English**

**Translations from English to PL**
### 'p ⊃ q' AND 'IF p, THEN q'

<table>
<thead>
<tr>
<th>D</th>
<th>W</th>
<th>if W, then D</th>
<th>W ⊃ D</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S</th>
<th>W</th>
<th>if S, then W</th>
<th>S ⊃ W</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>lied</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>didn’t lie</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td>F</td>
</tr>
</tbody>
</table>
‘\( p \supset q \)’ AND ‘IF \( p \), THEN \( q \)’

<table>
<thead>
<tr>
<th>( D )</th>
<th>( W )</th>
<th>if ( W ), then ( D )</th>
<th>( W \supset D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( S )</th>
<th>( W )</th>
<th>if ( S ), then ( W )</th>
<th>( S \supset W )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>lied</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
</tbody>
</table>
‘$p \supset q$’ AND ‘IF $p$, THEN $q$’

<table>
<thead>
<tr>
<th>$D$</th>
<th>$W$</th>
<th>if $W$, then $D$</th>
<th>$W \supset D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>lied</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$S$</th>
<th>$W$</th>
<th>if $S$, then $W$</th>
<th>$S \supset W$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>lied</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>didn’t lie</td>
<td>T</td>
</tr>
</tbody>
</table>
INCLUSIVE AND EXCLUSIVE ‘OR’

The Inclusive ‘or’

*In the inclusive sense, ‘p or q’ means ‘Either p or q or both.’*
Inclusive and Exclusive ‘or’

The Inclusive ‘or’

In the inclusive sense, ‘p or q’ means ‘Either p or q or both.’

▶ ‘Either the elevator or the escalator is working’
Inclusive and Exclusive ‘or’

The Inclusive ‘or’
In the inclusive sense, ‘p or q’ means ‘Either p or q or both.’

▶ ‘Either the elevator or the escalator is working’

The Exclusive ‘or’
In the exclusive sense, ‘p or q’ means ‘Either p or q, but not both.’
Inclusive and Exclusive ‘or’

The Inclusive ‘or’
In the inclusive sense, ‘p or q’ means ‘Either p or q or both.’

▶ ‘Either the elevator or the escalator is working’

The Exclusive ‘or’
In the exclusive sense, ‘p or q’ means ‘ Either p or q, but not both.’

▶ ‘Either you clean your room, or you’re grounded’
Inclusive and Exclusive ‘or’

The Inclusive ‘or’

In the inclusive sense, ‘p or q’ means ‘Either p or q or both.’

- ‘Either the elevator or the escalator is working’

The Exclusive ‘or’

In the exclusive sense, ‘p or q’ means ‘Either p or q, but not both.’

- ‘Either you clean your room, or you’re grounded’

‘∨’ translates to the inclusive ‘or’
OUTLINE

THE LANGUAGE PL
Syntax for PL
Semantics for PL

TRANSLATION FROM PL TO ENGLISH
A Translation Guide
Canonical Logical Form

TRANSLATIONS FROM ENGLISH TO PL
Translating into Canonical Logical Form
Call the phrases on the right-hand-side of the translation guide the *canonical logical expressions* of English.
**Canonical Logical Form**

- Call the phrases on the right-hand-side of the translation guide the *canonical logical expressions* of English.
- If the logical structure of an English statement is written in this form, then that statement is in *canonical logical form*. 

Example:

If both John loves Andrew and it is not the case that Andrew loves John, then it is not the case that John and Andrew will be friends.

Expressions in this form are easily translated into PL. Let $J = 'John loves Andrew$', $A = 'Andrew loves John$', and $F = 'John and Andrew will be friends'$. Then, $(J \land \neg A) \rightarrow \neg F$.
**Canonical Logical Form**

- Call the phrases on the right-hand-side of the translation guide the *canonical logical expressions* of English.
- If the logical structure of an English statement is written in this form, then that statement is in *canonical logical form*.
- E.g.,
Call the phrases on the right-hand-side of the translation guide the *canonical logical expressions* of English.

If the logical structure of an English statement is written in this form, then that statement is in *canonical logical form*.

E.g.,

*If both John loves Andrew and it is not the case that Andrew loves John, then it is not the case that John and Andrew will be friends.*
**Canonical Logical Form**

- Call the phrases on the right-hand-side of the translation guide the *canonical logical expressions* of English.
- If the logical structure of an English statement is written in this form, then that statement is in *canonical logical form*.
- E.g.,

  *If both John loves Andrew and it is not the case that Andrew loves John, then it is not the case that John and Andrew will be friends.*

- Expressions in this form are easily translated into PL. Let $J = 'John loves Andrew'$, $A = 'Andrew loves John'$, and $F = 'John and Andrew will be friends'$. Then,

  $$(J \land \neg A) \supset \neg F$$
English Sentences Not in Canonical Logical Form

John and Andrew won't be friends if John loves Andrew but Andrew doesn't love him back.

We'll have to say a bit more about how to translate sentences like this into PL.
English sentences not in canonical logical form

- John and Andrew won’t be friends if John loves Andrew but Andrew doesn’t love him back.
English Sentences Not in Canonical Logical Form

- John and Andrew won’t be friends if John loves Andrew but Andrew doesn’t love him back.
- We’ll have to say a bit more about how to translate sentences like this into PL.
OUTLINE

THE LANGUAGE PL
Syntax for PL
Semantics for PL

TRANSLATION FROM PL TO ENGLISH
A Translation Guide
Canonical Logical Form

TRANSLATIONS FROM ENGLISH TO PL
Translating into Canonical Logical Form
OUTLINE

THE LANGUAGE PL
Syntax for PL
Semantics for PL

TRANSLATION FROM PL TO ENGLISH
A Translation Guide
Canonical Logical Form

TRANSLATIONS FROM ENGLISH TO PL
Translating into Canonical Logical Form
TRANSLATING INTO CANONICAL LOGICAL FORM

- Consider:
Translating into Canonical Logical Form

- Consider:
  
  \[
  \text{Harry doesn't like chestnuts}
  \]
Translating into Canonical Logical Form

- Consider:
  
  \( \text{Harry doesn't like chestnuts} \)

- This has the same meaning as
TRANSLATING INTO CANONICAL LOGICAL FORM

▶ Consider:

Harry doesn’t like chestnuts

▶ This has the same meaning as

It is not the case that Harry likes chestnuts.
Translating into Canonical Logical Form

- Consider:

  *Harry doesn’t like chestnuts*

- This has the same meaning as

  *It is not the case that Harry likes chestnuts.*

- So, if we let $H = ‘Harry likes chestnuts’, then we can translate it into $PL$ as follows:

  $\sim H$
TRANSLATING INTO CANONICAL LOGICAL FORM

- A general strategy:
TRANSLATING INTO CANONICAL LOGICAL FORM

▶ A general strategy:
  ▶ for any sentence of English, find another sentence of English which has the same meaning as the first, and which is in canonical logical form.
Translating into Canonical Logical Form

- A general strategy:
  - for any sentence of English, find another sentence of English which has the same meaning as the first, and which is in canonical logical form.
Translating into Canonical Logical Form

- A general strategy:
  - for any sentence of English, find another sentence of English which has the same meaning as the first, and which is in canonical logical form.
  - Then, translate the sentence in canonical logical form into PL following the translation guide:

\[
\begin{align*}
\text{It is not the case that } p & \quad \rightarrow \quad \sim p \\
\text{Both } p \text{ and } q & \quad \rightarrow \quad p \land q \\
\text{Either } p \text{ or } q & \quad \rightarrow \quad p \lor q \\
\text{If } p, \text{ then } q & \quad \rightarrow \quad p \rightarrow q \\
p \text{ if and only if } q & \quad \rightarrow \quad p \equiv q
\end{align*}
\]
# SOME ISSUES: NEGATION

<table>
<thead>
<tr>
<th>THE LANGUAGE PL</th>
<th>TRANSLATION FROM PL TO ENGLISH</th>
<th>TRANSLATIONS FROM ENGLISH TO PL</th>
</tr>
</thead>
</table>

H


Some Issues: Negation

- I hate not getting what I want and I hate getting what I want.
SOME ISSUES: NEGATION

- I hate not getting what I want and I hate getting what I want.
 SOME ISSUES: NEGATION

- I hate not getting what I want and I hate getting what I want.
- It is not the case that I hate getting what I want, and I hate getting what I want.

Some Issues: Negation

- I hate not getting what I want and I hate getting what I want.
- It is not the case that I hate getting what I want, and I hate getting what I want.
**Some Issues: Negation**

- I hate not getting what I want and I hate getting what I want.
- It is not the case that I hate getting what I want, and I hate getting what I want.
- \(\sim\) I hate getting what I want • I hate getting what I want.
SOME ISSUES: NEGATION

- I hate not getting what I want and I hate getting what I want.
- It is not the case that I hate getting what I want, and I hate getting what I want.
- ~ I hate getting what I want  •  I hate getting what I want.
- ~H  •  H
SOME ISSUES: NEGATION

<table>
<thead>
<tr>
<th>H</th>
<th>~H</th>
<th>•H</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
SOME ISSUES: CONJUNCTION

Hannes loves peaches and he loves apples.

P = 'Hannes loves peaches' and A = 'Hannes loves apples'
Some Issues: Conjunction

- Hannes loves peaches and he loves apples.
  Hannes loves peaches but he loves apples.
Some Issues: Conjunction

- Hannes loves peaches and he loves apples.
  Hannes loves peaches but he loves apples.
- \( P = '\text{Hannes loves peaches}' \) and \( A = '\text{Hannes loves apples}' \)
Some Issues: Conjunction

- Hannes loves peaches and he loves apples.
  Hannes loves peaches but he loves apples.
- \( P = \) ‘Hannes loves peaches’ and \( A = \) ‘Hannes loves apples’
- \( P \land A \)
Some Issues: Conjunction

\[
p \text{ and } q \\
p, \text{ but } q \\
p; \text{ however, } q \\
p, \text{ though } q \\
p \text{ as well as } q
\]

\[\rightarrow p \cdot q\]
**Some Issues: Disjunction**

\[
p \text{ or } q \quad \left\{ \begin{array}{l} 
p \text{ unless } q \end{array} \right\} \rightarrow p \lor q
\]
Some Issues: The Material Conditional

If $p$, then $q$
- $p$ only if $q$
- $q$ if $p$
- $p$ is sufficient for $q$
- $q$ is necessary for $p$

$\rightarrow p \supset q$
**Some Issues: The Material Biconditional**

\[
p \text{ if and only if } q \equiv p \iff q
\]

\[p \text{ is necessary and sufficient for } q \]\