Announcements & Overview

- Administrative Stuff
  - **HW #2 is due today (by midnight – *via* Blackboard).**
    - Consult *Homework Guidelines & Tips* handout re HW #2
  - **The mid-term will be pushed back (another week) until March 4**
    - We will review for the mid-term in class on March 1
  - **Our final exam schedule has been announced**
    - Morning Section: 8–10am, April 28 (location TBA)
    - Afternoon Section: 8–10am, April 29 (location TBA)
- **I have posted a bunch of additional LSL symbolization problems, with solutions. See the latest handout on our course website.**
- Today: Unit #2, Finalé
  - Symbolizing English *Arguments* into LSL (one more example)
  - Then: Unit #3 — The (Truth-Functional) Semantics of LSL

Symbolizing Arguments: Example #4

Suppose no two contestants enter; then there will be no contest. No contest means no winner. Suppose all contestants perform equally well. Still no winner. There won't be a winner unless there's a loser. And conversely. Therefore, there will be a loser only if at least two contestants enter and not all contestants perform equally well.

- Step 0: Decide on atomic sentences and letters.

  T: At least two contestants enter. C: There is a contest.
  E: All contestants perform equally well.
  W: There is a winner. L: There is a loser.

- Step 1: Identify (and symbolize) the *conclusion* of the argument:
  - Conclusion: There will be a loser only if at least two contestants enter and not all contestants perform equally well.
    * "Logish": L only if T and not E.
    * LSL: $L \rightarrow (T \,\&\, {\sim}E)$.

– Step 2: Symbolize the premises (here, there are as many as five):

(1) Suppose no two contestants enter; then there will be no contest.
  - "Logish": Suppose that not T; then it is not the case that C.
  - LSL: '$\sim T \to \sim C$'.

(2) No contest means no winner.
  - "Logish": Not C means not W. [*i.e.*, not C *implies* not W.]
  - LSL: '$\sim C \to \sim W$'.

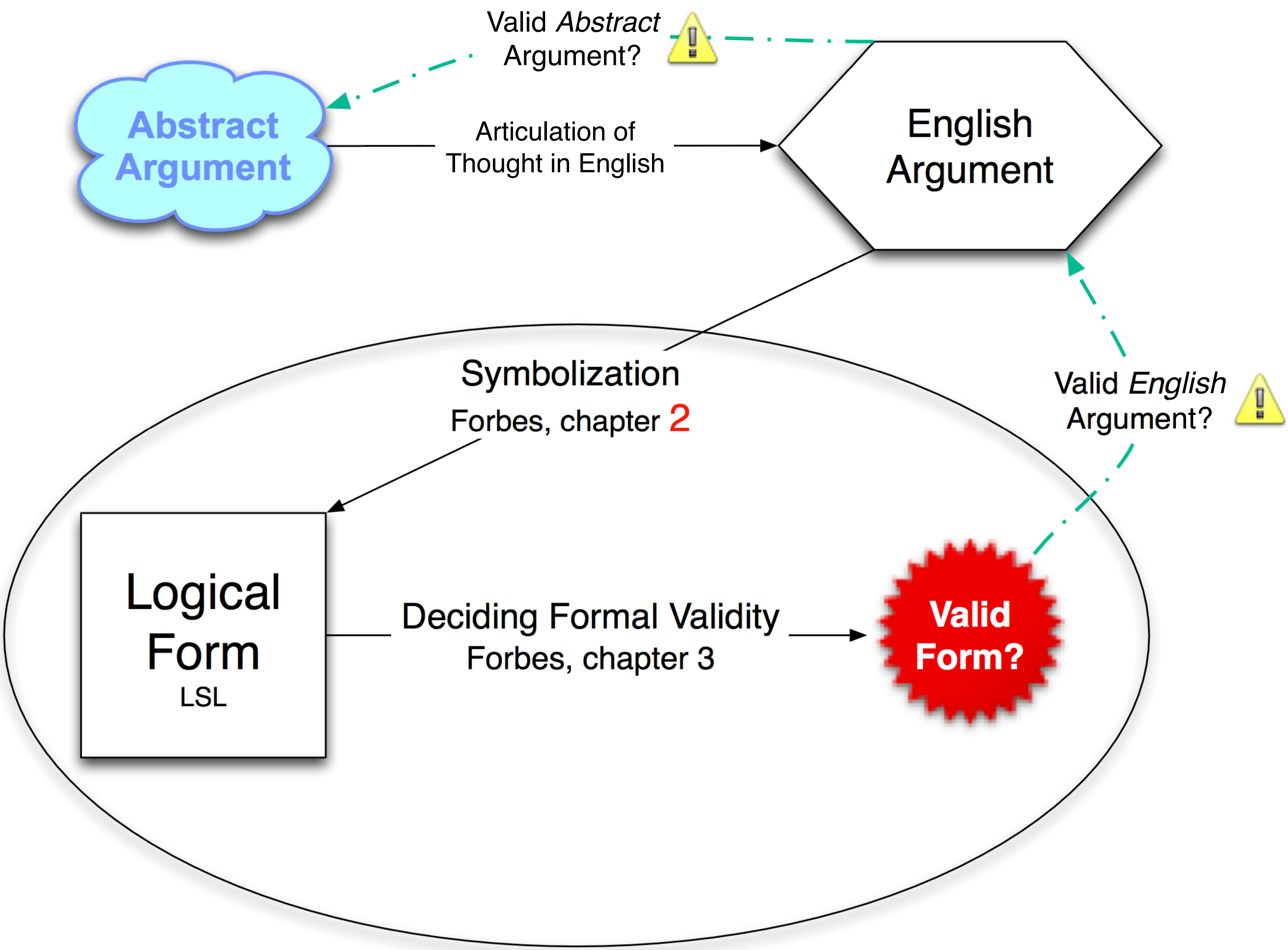
(3) Suppose all contestants perform equally well. Still no winner.
  - "Logish": Suppose E. Still not W. [*i.e.*, E *also* implies not W.]
  - LSL: '$E \to \sim W$'.

(4) There won't be a winner unless there's a loser. And conversely.
  - "Logish": Not W unless L, *and conversely*.
  - LSL: '$(\sim L \to \sim W)\ \&\ (\sim W \to \sim L)$'. [*i.e.*, not W *iff* not L.]

∗ The final product is the following *valid* sentential form:
$\sim T \to \sim C$. $\sim C \to \sim W$. $E \to \sim W$. $\sim L \leftrightarrow \sim W$. Therefore,
$L \to (T\ \&\ \sim E)$.

A Few Final Remarks on Symbolizing Arguments

- We saw the following premise our last argument: ‘There won’t be a winner unless there’s a loser. And conversely.’ I symbolized it as:
  - “Logish”: If not L, then not W, *and conversely*. [*i.e.*, not L *iff* not W.]
  - LSL: ‘$\sim L \leftrightarrow \sim W$’, *equivalently*: ‘$(\sim L \rightarrow \sim W)$ & $(\sim W \rightarrow \sim L)$’.
- One might wonder why I didn’t interpret the “and conversely” to be operating on the *unless* operator itself, rather than the *conditional* operator. This would yield the following *different* symbolization:
  - “Logish”: not W unless L, and L unless not W.
  - LSL: ‘$(\sim L \rightarrow \sim W)$ & $(\sim\sim W \rightarrow L)$’, *equivalently*: ‘$(\sim L \rightarrow \sim W)$ & $(W \rightarrow L)$’.
- Answer: This is a *redundant* symbolization in LSL, since ‘$\sim L \rightarrow \sim W$’ *is equivalent to* ‘$W \rightarrow L$’. Moreover, the resulting argument *isn’t* valid.
- **Principle of Charity**. If an argument $\mathscr{A}$ has two *plausible but semantically distinct* LSL symbolizations (where neither is *obviously* preferable) — and *only one of them is valid — choose the valid one*.


Valid *Abstract* Argument?
Abstract Argument
Articulation of Thought in English
English Argument
Symbolization
Forbes, chapter 2
Valid *English* Argument?
Logical Form
LSL
Deciding Formal Validity
Forbes, chapter 3
Valid Form?

Chapter 3 — Semantics of LSL: Truth Functions I

- The semantics of LSL is *truth-functional* — the truth value of a compound statement is a *function* of the truth values of its parts.
- Truth-conditions for each of the five LSL statement forms are given by *truth tables*, which show how the truth value of each type of complex sentence depends on the truth values of its constituent parts.
- Truth-tables provide a very precise way of thinking about *logical possibility*. Each row of a truth-table can be thought of as a *way the world might be*. The actual world falls into *exactly one* of these rows.
- In this sense, truth-tables provide a way to "see" *logical space*.
- Truth-tables will also provide us with a rigorous way to establish whether an argument form in LSL is valid (*i.e.*, sentential validity).
- We just look for rows of a salient truth-table in which all the premises are true and the conclusion is false. That's where we're headed.

Chapter 3 — Semantics of LSL: Truth Functions II

- We begin with negations, which have the simplest truth functions. The truth table for negation is as follows (we use $\top$ and $\bot$ for true and false):

p	$\sim p$
$\top$	$\bot$
$\bot$	$\top$

- In words, this table says that if p is true than $\sim p$ is false, and if p is false, then $\sim p$ is true. This is quite intuitive, and corresponds well to the English meaning of 'not'. Thus, LSL negation is like English negation.
- Examples:
  - It is not the case that Wagner wrote operas. ($\sim W$)
  - It is not the case that Picasso wrote operas. ($\sim P$)
- '$\sim W$' is false, since 'W' is true, and '$\sim P$' is true, since 'P' is false (like English).

Chapter 3 — Semantics of LSL: Truth Functions III

p	q	$p \& q$
⊤	⊤	⊤
⊤	⊥	⊥
⊥	⊤	⊥
⊥	⊥	⊥

- Notice how we have four (4) rows in our truth table this time (not 2), since there are four possible ways of assigning truth values to p and q.
- The truth-functional definition of & is very close to the English 'and'. A LSL conjunction is true if *both* conjuncts are true; it's false otherwise.
  - Monet and van Gogh were painters. ($M \& V$)
  - Monet and Beethoven were painters. ($M \& B$)
  - Beethoven and Einstein were painters. ($B \& E$)
- '$M \& V$' is true, since both 'M' and 'V' are true. '$M \& B$' is false, since 'B' is false. And, '$B \& E$' is false, since 'B' and 'E' are both false (like English).

Chapter 3 — Semantics of LSL: Truth Functions IV

p	q	$p \vee q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\top$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$

- Our truth-functional $\vee$ is not as close to the English 'or'. An LSL disjunction is true if *at least one* disjunct is true (false otherwise).
- In English, 'A or B' often implies that 'A' and 'B' are *not both true*. That is called *exclusive* or. In LSL, '$A \vee B$' is *not* exclusive; it is *inclusive* (true if both disjuncts are true). But, we *can* express exclusive or in LSL. How?
  - Either Jane austen or René Descartes was novelist. ($J \vee R$)
  - Either Jane Austen or Charlotte Bronte was a novelist. ($J \vee C$)
  - Either René Descartes or David Hume was a novelist. ($R \vee D$)
- The first two disjunctions are true because at least one their disjuncts is true, but the third is false, since both of its disjuncts are false.

Chapter 3 — Semantics of LSL: Truth Functions V

p	q	$p \to q$
⊤	⊤	⊤
⊤	⊥	⊥
⊥	⊤	⊤
⊥	⊥	⊤

- Our truth-functional $\to$ is farther from the English 'only if'. An LSL conditional is false iff its antecedent is true and its consequent is false.
- Consider the following English conditionals. [M = 'the moon is made of green cheese', O = 'life exists on other planets', and E = 'life exists on Earth']
  - If the moon is made of green cheese, then life exists on other planets.
  - If life exists on other planets, then life exists on earth.
- The LSL translations of these sentences are both true. '$M \to O$' is true because its antecedent 'M' is false. '$O \to E$' is true because its consequent 'E' is true. This seems to deviate from the English 'if'.

  [Soon, we'll *prove* the following *equivalence*: $\ulcorner p \to q \urcorner \dashv\vDash \ulcorner {\sim} p \vee q \urcorner$.]

Chapter 3 — Semantics of LSL: Truth Functions VI

p	q	$p \leftrightarrow q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\top$

- Our truth-functional $\leftrightarrow$ is also farther from the English 'if and only if'. An LSL biconditional is true iff both sides have the same truth value.
- Consider these two biconditionals. [M = 'the moon's made of green cheese', U = 'there are unicorns', E = 'life exists on Earth', and S = 'the sky is blue']
  - The moon is made of green cheese if and only if there are unicorns.
  - Life exists on earth if and only if the sky is blue.
- The LSL translations of these sentences are true. $M \leftrightarrow U$ is true because M and U are false. $E \leftrightarrow S$ is true because E and S are true. This seems to deviate from the English 'iff'. Soon, we'll *prove* the following:

$$\ulcorner p \leftrightarrow q \urcorner \dashv\vDash \ulcorner (p \,\&\, q) \vee (\sim p \,\&\, \sim q) \urcorner$$

Chapter 3 — Semantics of LSL: Truth Functions VII

- If our truth-functional semantics for '→' doesn't perfectly capture the English meaning of 'if ... then ...', then why do we define it this way?
- The answer has two parts. First, our semantics is *truth-functional*. This is an *idealization* — it yields the *simplest* ("Newtonian") semantics.
- And, there are only $2^4 = 16$ possible binary truth-functions. Why?
- So, unless one of the *other* 15 binary truth-functions is *closer* to the English conditional than '→' is, it's *the best we can do*, *truth-functionally*.
- More importantly, there are certain *logical properties* that the conditional *must* have. It can be shown that our definition of '→' is the *only* binary truth-function which satisfies all three of the following:

(1) *Modus Ponens* [p and $\ulcorner p \to q \urcorner \therefore q$] is a valid sentential form.

(2) Affirming the consequent [q and $\ulcorner p \to q \urcorner \therefore p$] is *not* a valid form.

(3) All sentences of the form $\ulcorner p \to p \urcorner$ are logical truths.

Chapter 3 — Semantics of LSL: Truth Functions VIII

- Here are all of the 16 possible binary truth-functions. I've given them all names or descriptions. [Only a few of these names were made up by me.]

p	q	$\top$	NAND	$\rightarrow$	$\sim p$	FI ($\leftarrow$)	$\sim q$	$\leftrightarrow$	NOR	$\vee$	NIFF	q	NFI	p	NIF	&	$\bot$
$\top$	$\top$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$
$\top$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
(1)?				Yes													
(2)?				Yes													
(3)?				Yes													

- Exercise: fill-in the three rows at the bottom (except for $\rightarrow$, which I have done for you already) concerning (1), (2), and (3) from the previous slide.

- You should be able to do this pretty soon (within the next week) ...

Chapter 3 — Semantics of LSL: Additional Remarks on →

- Above, I explained *why* our conditional → behaves "like a disjunction":
  1. We want a *truth-functional* semantics for →. This is a simplifying *idealization*. Truth-functional semantics are the simplest compositional semantics for sentential logic. [A "Newtonian" semantic model.]
  2. Given (1), the *only* way to define → is *our* way, since it's the *only* binary truth-function that has the following three essential *logical* properties:
     (i) *Modus Ponens* [p and $\ulcorner p \to q \urcorner \therefore q$] is a valid sentential form.
     (ii) Affirming the consequent [q and $\ulcorner p \to q \urcorner \therefore p$] is *not* a valid form.
     (iii) All sentences of the form $\ulcorner p \to p \urcorner$ are logical truths.
- There are *non*-truth-functional semantics for the English conditional.
- These may be "closer" to the English *meaning* of "if". But, they agree with our semantics for →, when it comes to the crucial *logical* properties (i)–(iii). Indeed, our → captures *most* of the (intuitive) *logical* properties of "if".

Constructing Truth-Tables for LSL Sentences

- With the truth-table definitions of the five connectives in hand, we can now construct truth tables for arbitrary compound LSL statements.
- The procedure for constructing the truth-table of p is as follows:
  1. Determine the number of rows in the truth-table. This is 2^n, where n is the number of atomic sentences in the compound statement p.
  2. The table will have $n + 1$ main columns: n columns for the atomic sentences in p, and one for the truth-values of p itself.
  3. The table will also have some "quasi-columns" — one for each LSL statement occurring in the compound p — which needn't be drawn explicitly, but which go into the determination of p's truth values.
  4. Place the atomic letters in the left most columns, in alphabetical order from left to right. And, place p in the right most column.
  5. Write in all possible combinations of truth-values for the atomic statements. There are 2^n of these — one for each row of the table.

6. Convention: start on the nth column (farthest down the alphabet) with the pattern $\top\bot\top\bot\ldots$ repeated until the column is filled. Then, go $\top\top\bot\bot\ldots$ in the $n-1$st column, $\top\top\top\top\bot\bot\bot\bot\ldots$ in the $n-2$nd column, etc. . . , until the very first column has been completed.
7. Finally, we compute the truth-values of p in each row of the table. Here, we start from the inside-out. We first copy the truth-values of the atoms, then we compute the negations, conjunctions, etc. which compose p. Finally, we will be in a position to compute the value of the main connective of p, at which point we'll be done with the table.

- Example: Step-By-Step Truth-Table Construction of '$A \leftrightarrow (B \,\&\, A)$.'

A	B	A	$\leftrightarrow$	$(B$	$\&$	$A)$
$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\top$
$\bot$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\bot$
$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$

Interpretations and the Relation of Logical Consequence

- An *interpretation* of an LSL formula p is an assignment of truth-values to all of the sentence letters in p — *i.e.*, a row in p's truth-table.
- A formula p is a *logical consequence* of a set of formulae S [written $S \vDash p$] just in case there is no interpretation (*i.e.*, no row in the joint truth-table of S and p) on which all the members of S are $\top$ but p is $\bot$.
- $S \vDash p$ is another way of saying that the argument from S to p is *valid*.
- Two LSL sentences p and q are said to be *logically equivalent* [written $p \dashv\vDash q$] iff they have the same truth-value on all (joint) interpretations.
- That is, p and q are logically equivalent iff *both* $p \vDash q$ *and* $q \vDash p$.
- I will often express $\ulcorner p \vDash q \urcorner$ by saying that $\ulcorner p$ *entails* $q \urcorner$. This is easier than saying that $\ulcorner q$ is a logical consequence of $p \urcorner$.
- The logical consequence relation $\vDash$ is our central theoretical relation.

Logical Truth, Logical Falsity, and Contingency: Definitions

- A statement is said to be logically true (or tautologous) if it is $\top$ on all interpretations. *E.g.*, any statement of the form $p \leftrightarrow p$ is tautological.

p	p	$\leftrightarrow$	p
$\top$	$\top$	$\top$	$\top$
$\bot$	$\bot$	$\top$	$\bot$

- A statement is logically false (or self-contradictory) if it is $\bot$ on all interpretations. *E.g.*, any statement of the form $p \mathbin{\&} {\sim}p$ is logically false:

p	p	$\&$	$\sim$	p
$\top$	$\top$	$\bot$	$\bot$	$\top$
$\bot$	$\bot$	$\bot$	$\top$	$\bot$

- A statement is contingent if it is *neither* tautological *nor* self-contradictory. Example: 'A' (or *any* basic sentence) is contingent.

A	A
$\top$	$\top$
$\bot$	$\bot$