Philosophy 148 — Announcements & Such

• Administrative Stuff
  – See the course website for all administrative information (also, note that lecture notes are posted the morning prior to each class):
    
    http://socrates.berkeley.edu/~fitelson/148/
  – Section times (II): Those of you who can’t make Tu @ 10–11, please fill-out an index card. New times: Mon or Wed 9–10 or 10–11.

• Last Time: Review of Boolean (Truth-Functional) Sentential Logic
  – Truth-Table definitions of connectives
  – Semantical (Metatheoretic) notions
    ∗ Individual Sentences: Logical Truth, Logical Falsity, etc.
    ∗ Sets of Sentences: Entailment (\(\models\)), Consistency, etc.

• Today: Finite Propositional Boolean Algebras, Review of Boolean (Truth-Functional) Predicate Logic, and a General Boolean Framework
Finite Propositional Boolean Algebras

- Sentences express *propositions*. We individuate propositions according to their logical content. If two sentences are logically equivalent, then they express the same proposition. [*E.g.*, “\(A \rightarrow B\)” and “\(\sim A \lor B\)”]

- A *finite propositional Boolean algebra* is a finite set of *propositions* which is *closed* under the (Boolean) logical operations.

- A set \(S\) is *closed* under a logical operation \(\lambda\) if applying \(\lambda\) to a member (or pair of members) of \(S\) always yields a member of \(S\).

- Example: consider a sentential language \(\mathcal{L}\) with three atomic letters “\(X\)”, “\(Y\)”, and “\(Z\)”. The set of propositions expressible using the logical connectives and these letters is a finite Boolean algebra of propositions.

- This Boolean algebra has \(2^3 = 8\) *atomic propositions* or *states* (*i.e.*, the rows of a 3-atomic sentence truth-table!). Question: How many propositions does it contain *in total*? [A: \(2^8 = 256\) — explanation later]
### Propositional Boolean Algebras: States, Truth-Tables, and Venn Diagrams

- **A literal** is either an atomic sentence or the negation of an atomic sentence (e.g., “A” and “\( \sim A \)” are the literals involving the atom “A”).

- **A state** of a Boolean algebra \( \mathcal{B} \) is a proposition expressed by a *state description* — a *maximal* conjunction of literals in a language \( \mathcal{L}_B \) describing \( \mathcal{B} \) (maximal: having exactly one literal for each atom of \( \mathcal{L}_B \)).

- Consider an algebra \( \mathcal{B} \) described by a 3-atom language \( \mathcal{L}_B \) (\( X, Y, Z \)). The states of \( \mathcal{B} \) are described by the \( 2^3 = 8 \) *state descriptions* of \( \mathcal{L}_B \):

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_1 )</td>
<td>( X &amp; Y &amp; Z )</td>
</tr>
<tr>
<td>( s_2 )</td>
<td>( X &amp; Y &amp; \sim Z )</td>
</tr>
<tr>
<td>( s_3 )</td>
<td>( X &amp; \sim Y &amp; Z )</td>
</tr>
<tr>
<td>( s_4 )</td>
<td>( X &amp; \sim Y &amp; \sim Z )</td>
</tr>
<tr>
<td>( s_5 )</td>
<td>( \sim X &amp; Y &amp; Z )</td>
</tr>
<tr>
<td>( s_6 )</td>
<td>( \sim X &amp; Y &amp; \sim Z )</td>
</tr>
<tr>
<td>( s_7 )</td>
<td>( \sim X &amp; \sim Y &amp; Z )</td>
</tr>
<tr>
<td>( s_8 )</td>
<td>( \sim X &amp; \sim Y &amp; \sim Z )</td>
</tr>
</tbody>
</table>
• We can “visualize” the states of $\mathcal{B}$ using a truth table or a Venn Diagram.

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
<th>$Z$</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>$s_1$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>$s_2$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>$s_3$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>$s_4$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>$s_5$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>$s_6$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>$s_7$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>$s_8$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>$s_2$</td>
<td>$s_3$</td>
</tr>
<tr>
<td>$s_4$</td>
<td>$s_5$</td>
<td>$s_6$</td>
</tr>
<tr>
<td>$s_7$</td>
<td>$s_8$</td>
<td></td>
</tr>
</tbody>
</table>

• Every proposition expressible in the sentential language $\mathcal{L}_B$ can be expressed as a disjunction of state descriptions (how does this work?).

• Thus, every proposition expressible in $\mathcal{L}_B$ can be “visualized” simply by shading combinations of the 8 state-regions of the Venn Diagram of $\mathcal{B}$.

• How many ways of shading the above Venn diagram are there? $2^8 = 256 = \text{the number of disjunctions of the } s_i (255) — \text{ plus 1 for } \bot$.

• That’s why there are $2^{2^3} = 2^8 = 256$ propositions (in total) in $\mathcal{B}$.
(Finite) Monadic Predicate Logic (MPL) I

• Consider the following two arguments:

\[ \begin{align*}
\text{Socrates is wise.} & \quad \text{Everyone is happy.} \\
& \quad \text{\textbullet} \quad \text{Someone is wise.} \\
& \quad \text{\textbullet} \quad \text{Plato is happy.}
\end{align*} \]

• Intuitively, both ① and ② are valid (why?). But, if we try to translate these into sentential logic, we get the invalid SL forms:

\[ \begin{align*}
\text{(1)SL} \quad S & \quad \text{\textbullet} \quad W \\
\text{(2)SL} \quad H & \quad \text{\textbullet} \quad P
\end{align*} \]

• In SL, we are not able to express what the premises and conclusions of these kinds of arguments have in common.

• If it’s not atomic sentences that the premises and conclusions of such arguments have in common, then what is it?

• This is what monadic predicate logic is about ...
(Finite) Monadic Predicate Logic II

- We need a *richer language* than SL — one which accurately captures the deeper *logical structure* of arguments like ① and ②. New Jargon:
  - A **predicate** is something which *applies to* an object or *is true of* an object. *E.g.*, The predicate *(is) Wise* applies to Socrates.
  - A **proper name** is a word or a phrase which *stands for*, or *refers to*, or *denotes* a person, place, or thing. *E.g.*, ‘Socrates’ is a proper name.
  - **Quantifiers** specify quantities. *E.g.*, ‘someone’ means *at least one* person and ‘everyone’ means *all* people. [think: “Some” and “All”]
  - The collection of objects to which the quantifiers in a statement are *relativized* is called the **domain of discourse** of the statement. *In this course, we’ll only work with finite domains* (e.g., the domain of ravens).
  - As we’ll explain soon, when we are restricted to finite domains, monadic predicate logic is *almost* “sentential logic in disguise”.

UCB Philosophy  
Logical Background, Cont’d  
01/24/08
(Finite) Mondacic Predicate Logic III

• Among the atomic sentences of MPL *in addition to SL sentence letters* are (new) strings of the form ‘$Xn$’, where ‘$X$’ is a (monadic) predicate, and ‘$n$’ is an individual constant (proper name).

• We use the lower-case letters ‘$a$’–‘$s$’ as individual constants (‘$t$’–‘$z$’ are used as *variables* — we won’t say too much about variables).

• Some examples of these new kinds of atomic sentences:
  
  - ‘Branden is tall.’ $\rightarrow$ ‘$Tb$’.
  
  - ‘Honda is an automobile manufacturer.’ $\rightarrow$ ‘$Ah$’.
  
  - ‘New York is a city.’ $\rightarrow$ ‘$Cn$’.

• As in SL, we can *combine* different MPL atomic sentences using the sentential connectives to yield complex sentences. For instance:
  
  - ‘Branden is tall, but Ruth is not tall.’ $\rightarrow$ ‘$Tb \& \sim Tr$’.
Quantifiers in (Finite) Mondaic Predicate Logic

- In finite domains, we can (almost) think of a universal claim $(\forall v)\phi v$ as a conjunction which asserts that the predicate expression $\phi$ applies to each object in the domain [i.e., as $\phi a \land (\phi b \land (\phi c \land \ldots))$].

- Analogously, in finite domains, we can (almost) think of an existential claim $(\exists v)\phi v$ as a disjunction which asserts that $\phi$ applies to at least one object in the domain [i.e., as $\phi a \lor (\phi b \lor (\phi c \lor \ldots))$].

- Upshot: when the size of the domain is finite (and, known), we can say everything we need to say about the domain using sentential logic.

- For each $\phi$ and each constant $\tau$, we can construct an atomic sentence $\phi \tau$. And, if we think of these as the atomic sentences of a sentential language, then we can express every claim we need to express using conjunctions, disjunctions, etc. of these atomic sentences.

- So, we can use finite Boolean algebras of propositions for FMPL too…
Consider the language $\mathcal{L}_2^2$, with two monadic predicates $F$ and $G$ and two individual constants $a$ and $b$. $\mathcal{L}_2^2$ has 16 state descriptions:

- $Fa \& Ga \& Fb \& Gb$
- $Fa \& Ga \& \sim Fb \& \sim Gb$
- $Fa \& \sim Ga \& Fb \& Gb$
- $Fa \& \sim Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& Ga \& Fb \& \sim Gb$
- $\sim Fa \& Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& \sim Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& \sim Ga \& \sim Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& \sim Ga \& \sim Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& \sim Ga \& \sim Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& \sim Ga \& \sim Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& \sim Ga \& \sim Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& \sim Ga \& \sim Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& \sim Ga \& \sim Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& \sim Ga \& \sim Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& \sim Ga \& \sim Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& \sim Ga \& \sim Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& \sim Ga \& \sim Ga \& \sim Fb \& \sim Gb$
- $\sim Fa \& \sim Ga \& \sim Ga \& \sim Fb \& \sim Gb$

This characterizes a Boolean algebra $\mathcal{B}$ with 16 states. We cannot easily visualize $\mathcal{B}$ with Venn diagrams. But, we can (and will) use truth-tables.

Note: the total number of propositions in $\mathcal{B}$ is very large ($2^{16} = 65536$). There are 65535 disjunctions of state-descriptions of $\mathcal{L}_2^2$ (plus 1 for $\bot$).
<table>
<thead>
<tr>
<th>$Fa$</th>
<th>$Ga$</th>
<th>$Fb$</th>
<th>$Gb$</th>
<th>State Descriptions ($s_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>$Fa &amp; Ga &amp; Fb &amp; Gb$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>$Fa &amp; Ga &amp; Fb &amp; \sim Gb$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>$Fa &amp; Ga &amp; \sim Fb &amp; Gb$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>$Fa &amp; Ga &amp; \sim Fb &amp; \sim Gb$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>$Fa &amp; \sim Ga &amp; Fb &amp; Gb$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>$Fa &amp; \sim Ga &amp; Fb &amp; \sim Gb$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>$Fa &amp; \sim Ga &amp; \sim Fb &amp; Gb$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>$Fa &amp; \sim Ga &amp; \sim Fb &amp; \sim Gb$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>$\sim Fa &amp; Ga &amp; Fb &amp; Gb$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>$\sim Fa &amp; Ga &amp; Fb &amp; \sim Gb$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>$\sim Fa &amp; Ga &amp; \sim Fb &amp; Gb$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>$\sim Fa &amp; Ga &amp; \sim Fb &amp; \sim Gb$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>$\sim Fa &amp; \sim Ga &amp; Fb &amp; Gb$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>$\sim Fa &amp; \sim Ga &amp; Fb &amp; \sim Gb$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>$\sim Fa &amp; \sim Ga &amp; \sim Fb &amp; Gb$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>$\sim Fa &amp; \sim Ga &amp; \sim Fb &amp; \sim Gb$</td>
</tr>
</tbody>
</table>
(Finite) Relational Predicate Logic (RPL)

- We won’t work much (hardly at all) with relational predicate logic in this course, but a similar trick can be done to use finite relational logical languages to characterize finite Boolean algebras of propositions:

- Consider a language $\mathcal{L}$, which has one 2-place predicate “$R$” and two individual constants “$a$” and “$b$”. $\mathcal{L}$ also has 16 state descriptions:

  $\begin{align*}
  Raa & \land Rab & \land Rba & \land Rbb & \quad Raa & \land Rab & \land \neg Rba & \land Rbb & \quad Raa & \land Rab & \land \neg Rba & \land Rbb \\
  Raa & \land Rab & \land \neg Rba & \land \neg Rbb & \quad Raa & \land \neg Rab & \land Rba & \land Rbb & \quad Raa & \land \neg Rab & \land Rba & \land \neg Rbb \\
  Raa & \land \neg Rab & \land \neg Rba & \land Rbb & \quad Raa & \land \neg Rab & \land \neg Rba & \land \neg Rbb & \quad \neg Raa & \land \neg Rab & \land Rba & \land Rbb \\
  \neg Raa & \land \neg Rab & \land Rba & \land \neg Rbb & \quad \neg Raa & \land \neg Rab & \land \neg Rba & \land \neg Rbb & \quad \neg Raa & \land \neg Rab & \land \neg Rba & \land \neg Rbb \\
  \neg Raa & \land \neg Rab & \land \neg Rba & \land \neg Rbb & \quad \neg Raa & \land \neg Rab & \land \neg Rba & \land \neg Rbb & \\
  \end{align*}$

- The truth-table for $\mathcal{L}$ looks very much like the one for $\mathcal{L}_2^2$, above, since each language ($\mathcal{L}$ and $\mathcal{L}_2^2$) has four atomic sentences. Here’s the table:
<table>
<thead>
<tr>
<th>$Raa$</th>
<th>$Rab$</th>
<th>$Rba$</th>
<th>$Rbb$</th>
<th>State Descriptions ($s_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>$Raa &amp; Rab &amp; Rba &amp; Rbb$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>$Raa &amp; Rab &amp; Rba &amp; \neg Rbb$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>$Raa &amp; Rab &amp; \neg Rba &amp; Rbb$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>$Raa &amp; Rab &amp; \neg Rba &amp; \neg Rbb$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>$Raa &amp; \neg Rab &amp; Rba &amp; Rbb$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>$Raa &amp; \neg Rab &amp; Rba &amp; \neg Rbb$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>$Raa &amp; \neg Rab &amp; \neg Rba &amp; Rbb$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>$Raa &amp; \neg Rab &amp; \neg Rba &amp; \neg Rbb$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>$\neg Raa &amp; Rba &amp; Rbb$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>$\neg Raa &amp; Rab &amp; Rba &amp; \neg Rbb$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>$\neg Raa &amp; Rab &amp; \neg Rba &amp; Rbb$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>$\neg Raa &amp; Rab &amp; \neg Rba &amp; \neg Rbb$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>$\neg Raa &amp; \neg Rab &amp; Rba &amp; Rbb$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>$\neg Raa &amp; \neg Rab &amp; Rba &amp; \neg Rbb$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>$\neg Raa &amp; \neg Rab &amp; \neg Rba &amp; Rbb$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>$\neg Raa &amp; \neg Rab &amp; \neg Rba &amp; \neg Rbb$</td>
</tr>
</tbody>
</table>

Assuming finitely many constant symbols, predicate logics (monadic or relational) characterize finite Boolean algebras of propositions.
Overview of Deductive Logic I

• Deductive Logic provides *formal theories of validity* (following-from). The logician thus *theoretically grounds* our *informal* validity notion(s).

• In English, there are various argument forms or patterns that are intuitively or informally valid. Here’s a simple, *sentential* example:

  Dr. Ruth is a man.
  (1) If Dr. Ruth is a man, then Dr. Ruth is 10 feet tall.
  \[\therefore\text{ Dr. Ruth is 10 feet tall.}\]

• Intuitively, the conclusion of (1) *follows-from* its premises, since *if* the premises of (1) *were* true, then (1)’s conclusion would *have to be* true.

• Our simplest logical theory (SL) correctly classifies this argument form (and many other valid English forms) as *valid* — an SL “success story”:

\[p.\]
(1_{SL}) \text{ If } p, \text{ then } q.
\[\therefore q.\]
Overview of Deductive Logic II

• However, there are many English arguments that are (intuitively, or “absolutely”) valid, but their SL forms are not valid. For instance:

  Socrates is wise.

  (2) ∴ Someone is wise.

• Intuitively, argument (2) is (“absolutely”) valid. But, if we try to translate this argument into SL, we get the following invalid SL form:

  (2_SL) p

  ∴ q

• This motivates the richer logical language MPL, which subsumes SL, and which adds additional structure that allows us to “see” its validity:

  (2_MPL) Ws

  ∴ (∃x)Wx
Overview of Deductive Logic III

• The relational predicate logical language is even richer than either SL or MPL. Indeed, it subsumes both SL and MPL. This language (RPL) can formalize many more English validities. Example:

   Everybody loves John.
   \( (3) \quad \therefore \text{Someone is loved by everyone.} \)

• Adequately formalizing this argument requires the use of a two-place predicate/relation: ‘\( L_{xy} \)’, which reads ‘\( x \) loves \( y \)’ or ‘\( y \) is loved by \( x \)’.

   \( (3_{\text{RPL}}) \quad (\forall x)Lxj \therefore (\exists y)(\forall x)Lxy \)

• Exercise: try symbolizing (3) in MPL. You’ll get something like this:

   \( (3_{\text{MPL?}}) \quad (\forall x)Jx \therefore (\exists x)Ex \)

• We will not use RPL much in this course. Mainly, we’ll use just SL/MPL.
Overview of Deductive Logic IV

• Logical theories replace imprecise, informal notions of following-from with precise, theoretical validity concepts (in formal logical languages).

  - One Informal Validity Notion.
    ∗ If $\mathcal{A}$’s premises are true, then $\mathcal{A}$’s conclusion must also be true.

  - Some Corresponding Theoretical Validity Concepts.
    ∗ An English argument $\mathcal{A}$ is SL-valid if there is no SL interpretation in which all of the premises of $\mathcal{A}_{SL}$ are T and the conclusion of $\mathcal{A}_{SL}$ is F, where $\mathcal{A}_{SL}$ is the SL form of the English argument $\mathcal{A}$.
    ∗ An English argument $\mathcal{A}$ is MPL-valid if there is no MPL interpretation in which all of the premises of $\mathcal{A}_{MPL}$ are T and the conclusion of $\mathcal{A}_{MPL}$ is F, where $\mathcal{A}_{MPL}$ is the MPL form of $\mathcal{A}$.
    ∗ Similarly for RPL-validity …

• The story of formal deductive logic does not end with RPL…
Overview of Deductive Logic V

- The full theory of first-order logic (LFOL) includes RPL, plus $n$-place predicates, the identity relation $=$, and also function symbols.

- LFOL can capture even more valid arguments than RPL. For instance, LFOL can capture arguments like the following mathematical one:

$$2 + 4 = 6$$

(4) \hspace{1cm} 3 \times 2 = 6$$

$$\therefore 2 + 4 = 3 \times 2$$

- Indeed, LFOL can capture just about any argument in just about any branch of modern mathematics. That’s a lot of expressive power.

- In PHIL 140A, we study the full theory of first-order logic (LFOL). There, we give a semantics for LFOL, and we show that there is a sound and complete proof theory for LFOL (but, no decision procedure for $\vdash$).

- Of course, even full first-order logic (LFOL) has its limitations...
Overview of Deductive Logic VI

- Some arguments involve quantification over not only objects but *properties*. These arguments are *second-order* and ∴ beyond LFOL.

- Leibniz (sometimes) talked as if the following argument were valid:
  
  (5) \(a\) and \(b\) have exactly the same (monadic) properties.
  
  \[\therefore a\) and \(b\) are identical.\]

- In second-order logic (SOL), (5) would be formalized as follows:

  \(5_{SOL} (\forall P)(Pa \leftrightarrow Pb).\)

  \[\therefore a = b.\]

- Note that the premise of (5) quantifies over (monadic) *predicates*.

- This is something that LFOL is not designed to do.

- We could also have an SOL which allows quantification over *relations*.

- Second-order logic is beyond 140A. It is touched upon (a little) in 140B.
Overview of Deductive Logic VII

• All the logics I’ve mentioned are *classical* deductive logics. Not all logicians think classical logics capture our intuitive validity notions.

• Classical logics all share the following two properties:
  (i) All arguments with contradictory premises (*e.g.*, \( p \& \sim p \)) are valid.
  (ii) All arguments with tautological conclusions (*e.g.*, \( p \lor \sim p \)) are valid.

• Some logicians think (i) and/or (ii) are *counterexamples* to the classical theory of validity (as an explication of our informal “following-from”).

• Such logicians propose alternative formal theories of validity (\( \vdash^* \)).

• Usually, non-classical logicians reject the classical (truth-functional) theory of the *conditional*. They adopt a non-classical conditional (\( \rightarrow^* \)) which obeys constraints like the deduction theorem (relative to \( \vdash^* \)).

\[ p \vdash^* q \text{ if and only if } \vdash^* p \rightarrow^* q \]

• These and other foundational questions about deductive logic are addressed in Philosophical Logic (142). We’ll see *some* overlap here…
Abstract Argument
Logical Form
Symbolization
Articulation of Thought in English
Valid Abstract Argument?
Deciding Formal Validity
Valid English Argument?
Valid Abstract Argument?
Logical Background, Cont’d
01/24/08
Why study logic *formally or symbolically*?

- In ordinary contexts, we want to know if arguments expressed *in English* are valid or invalid. But, in formal logic, we only study arguments expressible in *formal* languages (SL, MPL, etc.). *Why?*

- Analogous question: What we want from natural science is to understand natural systems. But, our theories (strictly) apply only to systems faithfully describable in mathematical terms.

- Although formal models are *idealizations* which abstract away some aspects of natural systems, they are *useful idealizations* that help us understand *many* natural relationships and regularities.

- Studying arguments expressible in formal languages allows us to develop and use powerful tools for testing validity, etc. We can’t capture *all* valid arguments this way. But, we can grasp *many*.

- We will take the same attitude toward inductive logic as well …
Inductive Logic — Basic Motivation and Ideas

• Intuitively, not all “logically good” arguments are deductively valid. Some invalid arguments seem (intuitively) logically better than others:

(6)  
\[ p. \text{Someone is wise.} \quad r. \text{Someone is either wise or unwise.} \]
\[ \therefore q. \text{Socrates is wise.} \]

(7)  
\[ r. \text{Someone is either wise or unwise.} \]
\[ \therefore q. \text{Socrates is wise.} \]

• Inductive logic should theoretically ground our intuition that (6) is a logically stronger argument than (7) is. Neither argument is valid.

• More ambitiously, an inductive logician might aim for a theory of “the degree to which the premises of an argument confirm its conclusion”.

• This ambitious project would aim to characterize a function \( c(C, P) \). And, an intuitive requirement would be that this function be such that:

\[ c(q, p) > c(q, r) \]

• This course is (mainly) about inductive logic. We will examine how probabilities might be used to quantitatively generalize deductive logic.
Logic and Epistemology — A Prelude I

As I mentioned, some have worried about the adequacy of classical logic as a formal explication of our informal “following-from” relation.

Here’s a fact about classical deductive logic that may seem “odd”:

(†) If $p$ and $q$ are (classically) logically inconsistent, then the argument from $p$ and $q$ to $r$ is (classically) valid — for any $r$.

There’s something “odd” about the fact that everything follows-from inconsistent premises, according to the classical formal explication of following-from. But, what, exactly, is supposed to be “odd” about it?

Here’s an epistemological principle that is downright crazy:

(‡) If one’s beliefs are inconsistent (and one knows that they are), then one should believe everything (i.e., every proposition).

It is clear that (‡) is false. There are things I know to be false, and I shouldn’t believe those things — no matter what else is true of me.
OK, (‡) is clearly false. So? What does that have to do with (†)?

After all, (†) is a about logic, and (‡) is about epistemology.

Perhaps those worried about (†) are assuming that logic and epistemology are connected, or bridged by something like:

(*) If an agent S’s belief set B is such that $B \models p$ (and S knows that $B \models p$), then it would be reasonable for S to infer/believe p.

If (*) were true, then (†) would imply (‡), and — as a result — classical logicians who accepted (*) would seem to be stuck with (‡) too.

More precisely, classical logicians who believe (*) should find it reasonable to believe (‡). But, they don’t (at least, they shouldn’t!).

But, this doesn’t force classical logicians to give up (†). They could give up (*) instead. In such contexts, logic (alone) doesn’t seem to tell us whether to infer something new, or reject something we already believe.