Logic I – Session 11
Plan for today

- Damien’s comments on quiz
- My comments on teaching feedback
- A bit more on the TF-completeness of SL
- Recap of proof of soundness of SD:
  \[ \Gamma \vdash P \text{ in SD, then } \Gamma \models P \]
- Begin to prove completeness of SD:
  \[ \Gamma \models P, \text{ then } \Gamma \vdash P \text{ in SD} \]
**TF-completeness**

- We can express any truth-function in SL.

- Find a sentence that expresses the TF for this TT schema:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
<td>A&amp;B</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td>A&amp;~B</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td></td>
<td>~A&amp;B</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td>~A&amp;~B</td>
</tr>
</tbody>
</table>

- We want an iterated disjunction of CSs for the T rows: 1 and 3.

  (A&B) v (~A&B).
Strictly, we haven’t yet proven that SL is TF-complete. We’d need to show that our algorithm always yields a sentence that expresses the truth-function we want. See 6.1E (1d) and 6.2E (1).

Not only is SL truth-functionally complete, but so is any language that contains formulae TF-equivalent to every sentence of SL.

E.g. \{&,v,\sim\}. (After all, that’s all we use in our algorithm!)

In fact, we can achieve TF-completeness with a single binary connective, ‘\|’.

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td>T</td>
</tr>
</tbody>
</table>
To see this, just add a step to our algorithm: translate the old sentence into one that only contains ‘|’.

The new one will be equivalent, so it will have the same TT, so it will expresses the same truth-function.

In our example, our algorithm generated \((A \& B) \lor (\sim A \& B)\).

To find an equivalent sentence, make replacements in stages.
TF-completeness with ‘|’

- We start with \((A \& B) \lor (\sim A \& B)\), which is of the form \(P \lor Q\).

- Now, \(P \lor Q \iff (P \mid P) \mid (Q \mid Q)\).

- Substitute \((A \& B)\) and \((\sim A \& B)\) for \(P\) and \(Q\)

- \((A \& B) \lor (\sim A \& B)\)
  \(\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad ((A \& B)(A \& B)) \mid ((\sim A \& B)(\sim A \& B))\)

- Now replace the remaining sub-sentences.

- \((A \& B) \iff (A \mid B)(A \mid B)\). And \((\sim A \& B) \iff ((A \mid A)\mid B)((A \mid A)\mid B)\).

- So we get:
  \(((A \mid B)(A \mid B) \mid (A \mid B)(A \mid B)) \mid (((A \mid A)\mid B)((A \mid A)\mid B) \mid ((A \mid A)\mid B)((A \mid A)\mid B))\)
We’ve just looked at one sentence. We haven’t yet proven that a language L with just ‘|’ is TF-complete.

To do that, we need to prove that for any sentence of SL, there is an equivalent sentence in L.

Provide an algorithm Z that makes step-by-step replacements like we did. Then prove that:

- Each step of Z preserves TV, and
- For any $P_{SL}$ of SL, Z turns $P_{SL}$ into a sentence $P_L$ of L.
Soundness of SD

Basic strategy to show soundness of SD: Use MI to prove that (*) holds for any line n of any SD derivation:

(*) If $P_n$ is the sentence on line n and $P_n$ is in the scope of only the assumptions in $\Gamma_n$, then $\Gamma_n \models P_n$.

So for our induction sequence, we use lines of SD derivations.

For basis clause: (*) holds for n=1.

For inductive clause: if (*) holds up to line n, it holds for n+1.

$P_{n+1}$ had to be justified by applying some SD rule to earlier lines. So, prove for each SD rule X: If $P_{n+1}$ is justified by X and (*) holds up to the nth line, then (*) holds for the n+1st.
Soundness of SD

(*) If $p_n$ is the sentence on line $n$ and $p_n$ is in the scope of only the assumptions in $\Gamma_n$, then $\Gamma_n \models p_n$.

Most of the proof involves the last step, going through each rule to prove this:

For each SD rule $X$: If $p_{n+1}$ is justified by $X$ and (*) holds up to the $n$th line, then (*) holds for the $n+1$st.

Last time, we went through &E and $\sim$I. Let's do one more: $\supset$I.

So suppose $p_{n+1}$ is justified by applying $\supset$I, and that (*) holds through line $n$. Then $p_{n+1}$ is of the form $Q_i \supset R_k$.

So, to prove: If $Q_i \supset R_k$ on line $n+1$ is justified by $\supset$I and is in the scope only of assumptions in $\Gamma_{n+1}$, then $\Gamma_{n+1} \models Q_i \supset R_k$. 
Soundness of SD

Since $Q_i \supset R_k$ is justified by $\supset I$, we have a subderivation from an auxiliary assumption $Q_i$ on line $i$ to $R_k$ on line $k$, where $i < k < n+1$.

And since (*) applies for all $n < n+1$, it applies to $i$ and $k$.

So $\Gamma_k \models R_k$.

Now note that since $Q_i \supset R_k$ on line $n+1$ is justified by applying $\supset I$ to the subderivation on $i-k$, no assumptions in $\Gamma_k$ can have been closed before $n+1$ except $Q_i$.

In other words, every assumption open at $k$, apart from $Q_i$, must still be open at $n+1$.

So $\Gamma_k \subseteq \Gamma_{n+1} \cup \{Q_i\}$. 
Soundness of SD

So far we have:

(a) $\Gamma_k \subseteq \Gamma_{n+1} \cup \{Q_i\}$, and

(b) $\Gamma_k \models R_k$.

Now remember from last time that for any sets $\Gamma_1$ and $\Gamma_2$:

If $\Gamma_1 \subseteq \Gamma_2$, then if $\Gamma_1 \models S$, then $\Gamma_2 \models S$.

So in particular, from (a), we know that since $\Gamma_k \subseteq \Gamma_{n+1} \cup \{Q_i\}$:

(c) If $\Gamma_k \models R_k$ then $\Gamma_{n+1} \cup \{Q_i\} \models R_k$.

So putting together (b) and (c): $\Gamma_{n+1} \cup \{Q_i\} \models R_k$.

So $\Gamma_{n+1} \models Qi \supset R_k$. I.e. $\Gamma_{n+1} \models P_{n+1}$. 
Completeness of SD

To prove: If $\Gamma \models P$, then $\Gamma \vdash P$ (in SD).

By contraposition, this is equivalent to:

- $\Gamma \nvdash P$ then $\Gamma \nvdash P$.

So we can assume $\Gamma \nvdash P$ and try to prove $\Gamma \nvdash P$.

We need lots of intermediate steps to do it...

...and an important new notion: maximal consistency

- $\Gamma$ is maximally consistent in SD (MC-SD) iff $\Gamma$ is consistent in SD and $\Gamma$ would become inconsistent if any additional sentence were added to it.
Plan for proving completeness

1. \( \Gamma \not\vdash P \)

2. \( \Gamma \vdash \sim P \)

3. \( \Gamma \vdash \sim P \)

4. \( \Gamma \cup \{\sim P\} \) is C-SD

5. \( \Gamma \cup \{\sim P\} \subseteq \Gamma^* \) (for some \( \Gamma^* \) that’s MC-SD) (6.4.5)

6. For any \( \Gamma^* \) that’s MC-SD, \( \Gamma^* \) is TF-C (6.4.8)

7. \( \Gamma \vdash P \) then \( \Gamma \not\vdash P \).

8. If \( \Gamma \models P \), then \( \Gamma \vdash P \).
Completeness of SD

To prove: If $\Gamma \not\vdash P$, then $\Gamma \cup \{\sim P\}$ is C-SD

Suppose $\Gamma \cup \{\sim P\}$ is NOT C-SD. Then it’s inconsistent in SD.

Then, by def., some $Q$ and $\sim Q$ are derivable from it.

But that means we can derive $Q$ and $\sim Q$ in a sub-derivation from $\Gamma$ together with the assumption $\sim P$.

We could then perform $\sim E$ on the subderivation, yielding $P$.

So we could get $P$ in the scope of only the assumptions in $\Gamma$.

So if $\Gamma \cup \{\sim P\}$ is NOT C-SD, then $\Gamma \vdash P$.

So if $\Gamma \not\vdash P$, then $\Gamma \cup \{\sim P\}$ is C-SD.
Plan for proving completeness

- $\Gamma \not\models P$
- $\Downarrow \checkmark$
- $\Gamma \cup \{\neg P\}$ is C-SD
- $\Gamma \cup \{\neg P\} \subseteq \Gamma^*$ (for some $\Gamma^*$ that's MC-SD) \hspace{1cm} (6.4.5)$
- $\Gamma \cup \{\neg P\}$ is TF-C \hspace{1cm} (6.4.8)$
- $\rightarrow \Gamma \models \neg P$

- If $\Gamma \models P$, then $\Gamma \not\models P$.
- If $\Gamma \models P$, then $\Gamma \models P$. 
Completeness of SD

Next, let's prove:

If $\Gamma \cup \{\sim P\}$ is TF-consistent (TF-C), then $\Gamma \not\models P$.

So assume $\Gamma \cup \{\sim P\}$ is TF-consistent (TF-C).

By def., there's a TVA that m.e.m. $\Gamma \cup \{\sim P\}$ true.

A TVA m.e.m. true $\Gamma \cup \{\sim P\}$ iff it m.e.m. $\Gamma$ true and $P$ false.

So there's a TVA that m.e.m. $\Gamma$ true and $P$ false.

So by def., $\Gamma \models P$ iff there's NO TVA that does that.

So $\Gamma \not\models P$. 
Plan for proving completeness

- $\Gamma \not\vdash P$
  - $\Gamma \cup \{\neg P\}$ is C-SD
  - $\Gamma \cup \{\neg P\} \subseteq \Gamma^*$ (for some $\Gamma^*$ that's MC-SD) (6.4.5)
  - For any $\Gamma^*$ that's MC-SD, $\Gamma^*$ is TF-C (6.4.8)
  - $\Gamma \cup \{\neg P\}$ is TF-C
  - $\Gamma \not\models P$

- $\Gamma \not\vdash P$ then $\Gamma \not\models P$.

- If $\Gamma \models P$, then $\Gamma \vdash P$.
Completeness of SD

Next, let’s prove:

If $\Gamma \cup \{\neg P\} \subseteq \Gamma^*$ for some $\Gamma^*$ that’s MC-SD and for any $\Gamma^*$ that’s MC-SD, $\Gamma^*$ is TF-C, then $\Gamma \cup \{\neg P\}$ is TF-C.

So assume $\Gamma \cup \{\neg P\} \subseteq \Gamma^*$ for some $\Gamma^*$ that’s MC-SD and for any $\Gamma^*$ that’s MC-SD, $\Gamma^*$ is TF-C.

Suppose $\Gamma \cup \{\neg P\}$ is NOT TF-C.

Then there’s no TVA that m.e.m. $\Gamma \cup \{\neg P\}$ true.

But since $\Gamma \cup \{\neg P\} \subseteq \Gamma^*$, any TVA that m.e.m. $\Gamma^*$ true would also m.e.m. $\Gamma \cup \{\neg P\}$ true.

So there’s no TVA that m.e.m. $\Gamma^*$ true. I.e.: $\Gamma^*$ is NOT TF-C.

But since $\Gamma^*$ is MC-SD, and for any $\Gamma^*$ that’s MC-SD, $\Gamma^*$ is TF-C, $\Gamma^*$ is TF-C.

Our assumption led to a contradiction. So $\Gamma \cup \{\neg P\}$ is TF-C.
Plan for proving completeness

- $\Gamma \not\models P$
  - $\therefore$ $\Gamma \cup \{\neg P\}$ is C-SD
  - $\therefore$ $\Gamma \cup \{\neg P\} \subseteq \Gamma^*$ (for some $\Gamma^*$ that's MC-SD) \hspace{1cm} (6.4.5)
  - $\because$ For any $\Gamma^*$ that's MC-SD, $\Gamma^*$ is TF-C \hspace{1cm} (6.4.8)
    - $\therefore$ $\Gamma \cup \{\neg P\}$ is TF-C
      - $\therefore$ $\Gamma \not\models P$

- $\Gamma \not\models P$ then $\Gamma \not\models P$.

- If $\Gamma \models P$, then $\Gamma \vdash P$. 