Introduction to Logic 4

Last time:

- The meaning of the connectives
- Truth tables
- Tautologies, Inconsistencies and Equivalences
- Arguments and validity

This time:

- Functional completeness
- The Sheffer Stroke
- Logical Equivalences
- Limitations of Truth Tables

Functional Completeness

- Our definition of the PC includes the connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$.
- Our motivation for introducing these connectives came from considering sentences of English
- Perhaps surprisingly, it turns out that it is not necessary, strictly speaking, to have so many connectives.

Proposition: *We can express all of the connectives in terms of negation ($\neg$) and conjunction ($\wedge$).*

Proof (sketch):

We can show that for any sentences A and B:

$$
\begin{aligned}
A \vee B &\equiv \neg(\neg A \wedge \neg B) \\
A \rightarrow B &\equiv \neg(A \wedge \neg B) \\
A \leftrightarrow B &\equiv \neg(A \wedge \neg B) \wedge \neg(B \wedge \neg A)
\end{aligned}
$$

Proof continued:

- *We can show logical equivalence using the method of truth tables.*

  Consider: $A \vee B \equiv \neg(\neg A \wedge \neg B)$*:*

A	B	$\neg A$	$\neg B$	$A \vee B$	$\neg A \wedge \neg B$	$\neg(\neg A \wedge \neg B)$
t	t	f	f	t	f	t
t	f	f	t	t	f	t
f	t	t	f	t	f	t
f	f	t	t	f	t	f

- *We can provide similar demonstrations for the other connectives.*

$\square$

- There are other combinations of connectives that are **functionally complete in this sense** (e.g. $\neg$ and $\vee$).
- It is even possible to find a *single* connective that is functionally complete!

The Sheffer Stroke

- The Sheffer Stroke is a logical connective written as $|$ with the following meaning (i.e. truth function):

$\vert$	t	f
t	f	t
f	t	t

Proposition: *All of the connectives introduced as part of the language of the PC can be expressed in terms of the single connective* $|$.

Proof: *We must show that any sentence of the PC can be re-written as an equivalent sentence involving involving only the connective* $|$.

Note: we can simplify the proof by noting that we have already shown (in sketch) that the connectives can be expressed in terms of $\neg$ and $\wedge$.

Proof continued:

There are two cases to consider:

1. $\neg A$ *is equivalent to* $A|A$

A	$\neg A$	$A\|A$
t	f	f
f	t	t

2. $A \wedge B$ *is equivalent to* $(A|B)|(A|B)$

A	B	$A \wedge B$	$A\|B$	$(A\|B)\|(A\|B)$
t	t	t	f	t
t	f	f	t	f
f	t	f	t	f
f	f	f	t	f

The Sheffer Stroke is functionally complete. $\square$

So What?

- It is interesting to observe that logically speaking, the full set of connectives is not necessary.

  Question: Why is it that natural languages such as English have so many connectives when they could be more economical?

- The observation is also helpful when we want to prove things about the PC itself.
  - Note that this assumption helped to shorten the proof that the Sheffer Stroke was functionally complete.
  - Many other facts about the PC can be proved more easily using the same trick.

More on Logical Equivalence

- We have already noted that different sentences of the PC can be logically equivalent.
- For example, for any sentences A and B we showed (last lecture) that

$$A \rightarrow B \equiv \neg A \vee B$$

- Here are some further examples:

$$\begin{aligned} A \wedge B &\equiv B \wedge A \\ A \vee B &\equiv B \vee A \end{aligned}$$

  These are rather obvious. They show that $\wedge$ and $\vee$ are **commutative** operators.

$$\begin{aligned} A \wedge (B \wedge C) &\equiv (A \wedge B) \wedge C \\ A \vee (B \vee C) &\equiv (A \vee B) \vee C \end{aligned}$$

  These equivalences show that $\wedge$ and $\vee$ are **associative** operators.

- More interesting are the following ‘laws’ of **distributivity**:

$$
\begin{aligned}
A \wedge (B \vee C) &\equiv (A \wedge B) \vee (A \wedge C) \\
A \vee (B \wedge C) &\equiv (A \vee B) \wedge (A \vee C)
\end{aligned}
$$

- The following equivalences are known as **De Morgan’s laws**:

$$
\begin{aligned}
\neg(A \wedge B) &\equiv \neg A \vee \neg B \\
\neg(A \vee B) &\equiv \neg A \wedge \neg B
\end{aligned}
$$

- **Aside:** De Morgan (1806–1871) was a pioneer of the algebraic approach to logic.

- Let us introduce explicit symbols to represent *inconsistency* and *tautology*.
  - $\perp$ represents a proposition that is always **false** (inconsistency).
  - $\top$ represents a proposition that is always **true** (tautology).
- It is not difficult to see that:

$$\begin{array}{lcl} A \wedge \perp & \equiv & \perp \\ A \wedge \top & \equiv & A \\ A \vee \perp & \equiv & A \\ A \vee \top & \equiv & \top \\ A \wedge \neg A & \equiv & \perp \\ A \vee \neg A & \equiv & \top \end{array}$$

- There are many other equivalences (See: Kelly p.12 for a summary of some of the most important)

- The various logical equivalences provide a means of *simplifying* expressions.
- Consider for example:

$$(p \wedge \neg q) \vee (r \wedge (p \wedge q))$$

  We can simplify this as follows:

$$\begin{array}{lll}
 & (p \wedge \neg q) \vee (r \wedge (p \wedge q)) & \\
\equiv & (p \wedge \neg q) \vee ((p \wedge q) \wedge r) & \text{Commutativity} \\
\equiv & (p \wedge \neg q) \vee (p \wedge (q \wedge r)) & \text{Associativity} \\
\equiv & p \wedge (\neg q \vee (q \wedge r)) & \text{Distributivity} \\
\equiv & p \wedge ((\neg q \vee q) \wedge (\neg q \vee r)) & \text{Distributivity} \\
\equiv & p \wedge (\top \wedge (\neg q \vee r)) & \text{Tautology} \\
\equiv & p \wedge (\neg q \vee r) & \text{Identity of } \wedge
\end{array}$$

- Note that the final line is also equivalent to:

$$p \wedge (q \rightarrow r)$$

Limitations of the Method of Truth Tables

- In principle, the method of truth tables can be applied to answer questions about:
  - the validity of arguments
  - consistency and inconsistency
  - logical equivalence
- There are *practical* limitations to this method however.

You may have noticed the following:

- for a sentence with 1 propositional variable, the truth table has two rows.
- for a sentence with 2 (distinct) propositional variables, the truth table has four rows.

Question: *In general, for a sentence with n distinct propositional variables, how many rows does its truth table have?*

- To appreciate what this means in practice, suppose that the sentence has 10 distinct propositional variables.
- In this case, the truth table has $2^{10} = 1024$ rows. (That's rather a lot for a person to work out, but we've got fast computers, right?)
- For a sentence with 50 distinct propositional variables, the number of rows is:

$$2^{50} = 1,125,899,906,842,624$$

- Calculating the truth table at the rate of one million rows per second, would still require **approximately 36 years** to complete the table.

Summary

- The stock of connectives we used to define the language of the PC are not strictly necessary.
- It is possible to find smaller sets of connectives that are functionally complete.
- The Sheffer Stroke is a single functionally complete connective.
- Logical equivalences can be used to simplify sentences of the PC.
- The method of truth tables has practical limitations which restrict its usefulness.