Logic I: Lecture 10

s.butterfill@warwick.ac.uk

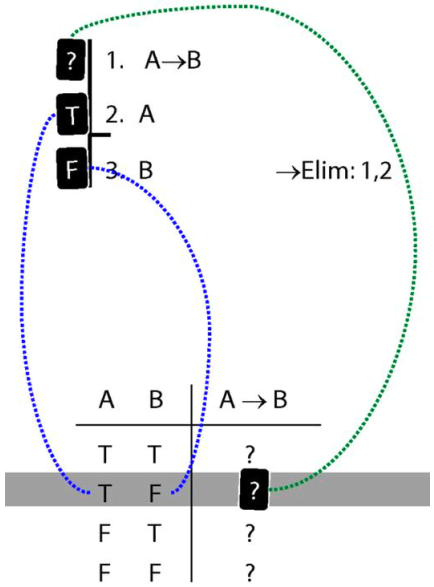
Readings refer to sections of the course textbook, *Language, Proof and Logic.*

1. What does '→' mean?

Reading: §7.1

Assuming that the rules of Fitch are such that it is impossible to prove an argument which is not logically valid, the truth table for → is fixed if we accept →Elim and →Intro.

How do the rules of proof for → fix its truth table?



2. Not If

If she has seen it, I am dead.
A → B

That's not true.
¬(A → B)

If she has seen it, I am not dead.
A → ¬B

A	B	A → B	¬(A → B)	A → ¬B
T	T	T	F	F
T	F	F	T	T
F	T	T	F	T
F	F	T	F	T

3. Fubar Rules

Reading: §8.3

Consider this made-up rule:

∧Fubar:
*
…
* ∧ #

Q1. What would be wrong with adding ∧Fubar to Fitch?
Q2. What would be wrong with having ∧Fubar in any system of proof?

4. ↔ : truth tables and rules

A	B	A ↔ B
T	T	T
T	F	F
F	T	F
F	F	T

Biconditional Elimination (↔ Elim)

P ↔ Q (or Q ↔ P)
⋮
P
⋮
▷ Q

Biconditional Introduction (↔ Intro)

P
⋮
Q
Q
⋮
P
▷ P ↔ Q

5. Translation with Quantifiers

Reading: §9.5, §9.6

All discordians weep:
∀x(Dscrdn(x) ⟶ Wps(x))

All **quadrumanous** discordians weep:
∀x((**Quadr(x)** ∧ Dscrdn(x)) ⟶ Wps(x))

All quadrumanous discordians weep **and wail**:
∀x((Quadr(x) ∧ Dscrdn(x)) ⟶ (Wps(x) ∧ **Wls(x)**))

All quadrumanous discordians weep and wail **except Gillian Deleude**:
∀x((Quadr(x) ∧ Dscrdn(x) ∧ ¬**(x=a)**) ⟶ (Wps(x) ∧ Wls(x)))

6. ∃Intro

Reading: §13.2

Disjunction Introduction (∨ Intro)

	P_i
	⋮
▷	$P_1 \vee \ldots \vee P_i \vee \ldots \vee P_n$

4.	YelBk(a)	
5.	∃x YelBk(x)	

7. What does ∃ mean?

Reading: §9.4

We give the meaning of ∃ by specifying what it takes for a sentence containing ∃ to be true:

1. Give every object a name.
2. For each name in turn, create a new sentence like this: delete the quantifier and replace all instances of the variable it binds with that name.
3. If ANY of the new sentences are true, so is the original sentence.

8. There Does Not Exist

Something is not dead:

∃x ¬Dead(x)

Nothing is dead:

¬∃x Dead(x)

Everything is not broken:

∀x ¬Broken(x)

Not everything is broken:

¬∀x Broken(x)

1.		
2.	a=a	=Intro
3.	∃x (x=x)	∃Intro: 2

1.	¬∃x Dead(x)	
2.	Dead(a)	
3.	∃x Dead(x)	∃Intro: 2
4.	⊥	⊥Intro: 1,3
5.	¬Dead(a)	¬Intro: 2-4
6.	∃x ¬Dead(x)	∃Intro: 5

9. Quantifier Equivalences: ¬∀x Created(x) ⊣⊨ ∃x ¬Created(x)

Reading: §10.1, §10.3, §10.4