Logic I: Fast Lecture 06

s.butterfill@warwick.ac.uk

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

1. Quantifier Equivalences: $\neg\forall x\ \text{Created}(x) \dashv\vDash \exists x\ \neg\text{Created}(x)$

Reading: §10.1, §10.3, §10.4

2. Proof Example: $\exists x\ \text{Dead}(x) \vdash \neg\forall x\neg\ \text{Dead}(x)$.

1.	$\exists x\ F(x)$	
2.	$\boxed{a}\ F(a)$	
3.	$\forall y \neg F(y)$	
4.	$\neg F(a)$	$\forall$Elim: 3
5.	$\bot$	$\bot$Intro: 2,4
6.	$\neg\forall y \neg F(y)$	$\neg$Intro: 3-5
7.	$\neg\forall y \neg F(y)$	$\exists$Elim: 2-6

3. Proof Example: $\neg\forall x\ \text{Dead}(x) \vdash \exists x\neg\ \text{Dead}(x)$.

1.	$\neg\forall x\ \text{Dead}(x)$	
2.	$\neg\exists x \neg\text{Dead}(x)$	
3.	$\boxed{a}$	
4.	$\neg F(a)$	
5.	$\exists x \neg\text{Dead}(x)$	$\exists$Intro: 4
6.	$\bot$	$\bot$Intro: 2,5
7.	$\neg\neg F(a)$	$\neg$Intro: 4-6
8.	$F(a)$	$\neg$Elim: 8
9.	$\forall x\ \text{Dead}(x)$	$\forall$Intro: 3-8
10.	$\bot$	$\bot$Intro: 1,9
11.	$\neg\neg\exists x \neg\text{Dead}(x)$	$\neg$Intro: 2-10
12.	$\exists x \neg\text{Dead}(x)$	$\neg$Elim: 11

4. Quantifier Equivalences: $\forall x(\text{Square}(x) \rightarrow \text{Broken}(x)) \dashv\vDash \forall x(\neg\text{Broken}(x) \rightarrow \neg\text{Square}(x))$

Reading: §10.3

P	Q	P→Q	¬Q→¬P
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

5. There Does Not Exist

Something is not dead:

$\exists x\ \neg\text{Dead}(x)$

Nothing is dead:

$\neg\exists x\ \text{Dead}(x)$

Everything is not broken:

$\forall x\ \neg\text{Broken}(x)$

Not everything is broken:

$\neg\forall x\ \text{Broken}(x)$

1.		
2.	$a=a$	=Intro
3.	$\exists x\ (x=x)$	$\exists$Intro: 2

1.	$\neg\exists x\ \text{Dead}(x)$	
2.	$\text{Dead}(a)$	
3.	$\exists x\ \text{Dead}(x)$	$\exists$Intro: 2
4.	$\bot$	$\bot$Intro: 1,3
5.	$\neg\text{Dead}(a)$	$\neg$Intro: 2-4
6.	$\exists x\ \neg\text{Dead}(x)$	$\exists$Intro: 5

1. ∃x ¬Dead(x)
2. ¬∃x Dead(x)

Counterexample:
Domain: {a,b}
Dead : {b}

6. Two Things Are Broken

Reading: §14.1

To translate sentences involving number into FOL, use identity. For example,

'Two things are broken' might be translated as:

∃x ∃y (Broken(x) ∧ Broken(y) ∧ ¬(x=y))

7. There Is Exactly One

There is one creator (at least one, maybe more).

∃x Creator(x)

Brian is the one and only creator.

Creator(b) ∧ ∀x(Creator(x) ⟶ x=b)

All squares are broken.

∀x(Sqr(x) ⟶ Brkn(x))

There is one and only one creator.

∃y(Creator(y) ∧ ∀x(Creator(x) ⟶ x=y))

or:

∃y(∀x(Creator(x) ↔ x=y))

8. Variables

Names : a, b, c, …

Variables : x, y, z, w, …

Variables are for saying several things about one thing even without specifying which thing it is

NB: 'Some x is a horse and x is dead' ain't English.

9. More Dead Horse

Reading: §11.4, §11.5

1	∃y∀x StoreFor(y,x)	
2	a ∀x StoreFor(a,x)	
3	StoreFor(a,a)	∀Elim: 2
4	∃x StoreFor(x,x)	∃Intro: 3
5	∃x StoreFor(x,x)	∃Elim: 2-4

"Tesco is a store for everything"

∀x StoreFor(b,x)

Tesco is a store for everything except dead horses

∀x (¬DeadHorse(x) ⟶ StoreFor(b,x))

Tesco is a store for everything except Tesco

∀x (¬x=b ⟶ StoreFor(b,x))

There is a store for everything except itself

∃y ∀x (¬x=y ⟶ StoreFor(y,x))

10. ↔ : 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

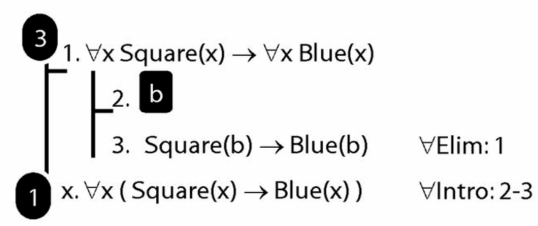
Q
⋮
P

▷ P ↔ Q

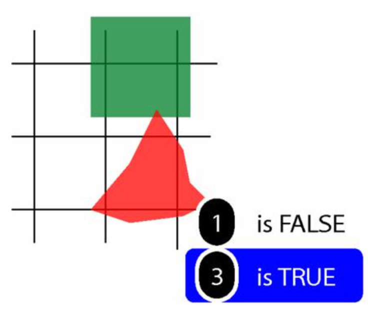
11. ∀Intro: An Incorrect Proof

Reading: §13.1, §13.2

This proof is wrong, but why?:

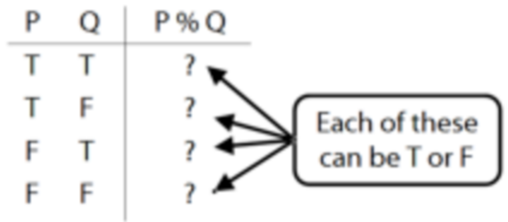


There is a counterexample to the argument:



12. How Big Is a Truth-Table?

How many truth-functions can be constructed using 2 sentence letters?



13. Truth-functional completeness

Reading: §7.4

'A set of truth-functors is said to be *expressively adequate* (or sometimes *functionally complete*) iff, for every truth-function whatever, there is a formula containing only those truth-functors which express that truth-function, i.e. which has as its truth-table the truth-table specifying that function.' (Bostock, *Intermediate Logic* p. 45)

Illustration of the proof that {¬, ∧, ∨} is truth-functionally complete:

P	Q	P→Q	
T	T	T	[P∧Q] ∨
T	F	F	
F	T	T	[¬P∧Q] ∨
F	F	T	[¬P∧ ¬Q]

[P∧Q] ∨ [¬P∧Q] ∨ [¬P∧ ¬Q]

Exercise assuming {¬,∨,∧} is truth-functionally complete, show that {¬,∨} is.

14. Soundness and Completeness: Statement of the Theorems

Reading: §8.3, §13.4

'A ⊢ B' means there is a proof of B using premises A

'⊢ B' means there is a proof of B using no premises

'A ⊨ B' means B is a logical consequence of A

'⊨ B' means B is a tautology

'A $\vDash_{TT}$ B' means B is a logical consequence of A just in virtue of the meanings of truth-functions (the textbook LPL calls this 'tautological consequence')

Soundness: If A ⊢ B then A ⊨ B

i.e. if you can prove it in Fitch, it's valid

Completeness: If A $\vDash_{TT}$ B then A ⊢ B

i.e. if it's valid just in virtue of the meanings of the truth-functional connectives, then you can prove it in Fitch.

15. Exercises

These exercises will be discussed in seminars the week after this lecture. The numbers below refer to the numbered exercises in the course textbook, e.g. '1.1' refers to exercise 1.1. on page 39 of the second edition of *Language, Proof and Logic*. Exercises marked '*' are optional.

13.12, 13.14, 13.16

13.43–13.45

13.49–13.50

9.18–9.19

11.10, 11.13

14.2

7.25, 7.26, *7.28, 7.29