Logic

M269 Unit 6

Phil Molyneux

11 April 2021

M269 Logic Topics
Agenda

- Welcome & Introductions
- Logic topics:
  - Propositional and predicate logic
  - Truth tables, logical equivalences and *valid* arguments
  - Truth and *interpretations* in logic
  - *Justified* arguments and *Natural Deduction*
- Exercises similar to CMAs and exam
- *Key aim:* Identify where people have problems and how to overcome them.
- Slides http://www.pmolyneux.co.uk/OU/M269FolderSync/M269TutorialNotes/M269TutorialLogic/
- *Adobe Connect* — if you or I get cut off, wait till we reconnect (or send you an email)

M269 Tutorial

Introductions — Me

- *Name* Phil Molyneux
- *Background* Physics and Maths, Operational Research, Computer Science
- *First programming languages* Fortran, BASIC, Pascal
- *Favourite Software*
  - Haskell — pure functional programming language
  - Text editors TextMate, Sublime Text — previously Emacs
  - Word processing in LaTeX
  - Mac OS X
- *Learning style* — I read the manual before using the software (really)

M269 Tutorial

Introductions — You

- *Name* ?
- *Position in M269* ? Which part of which Units and/or Reader have you read ?
- *Particular topics you want to look at* ?
- *Learning Syle* ?

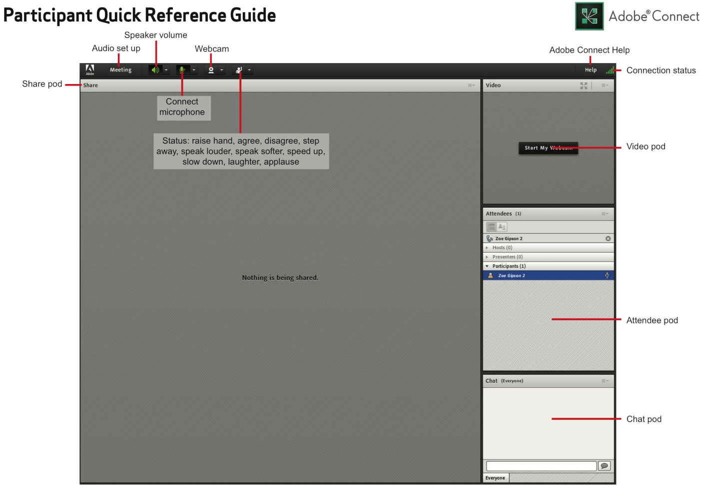
M269 Logic

References

- M269 Unit 6 Section 1.2 Reading 6.2 — Chapter 2 of *Logic and the limits of computing*, *Propositional logic*
- M269 Unit 6 Section 2 Reading 6.3 — Chapter 3 of *Logic and the limits of computing*, *Relations and predicate logic*
- The above two introduce the idea of a *valid* argument
- M269 Unit 7 Section 2 *Logic revisited* — Section 2.3 *A proof system* introduces the idea of *justified* arguments and *Natural Deduction* proofs
- Material based on Allan Grimley's notes for M269 on *Natural Deduction*
- Calculating with logic — manipulating truth tables and finding equivalent propositions — logic puzzles (optional)

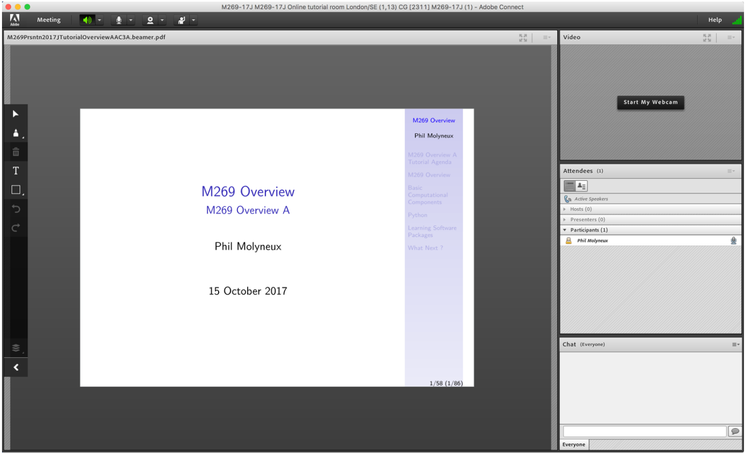
Adobe Connect

Interface — Student Quick Reference



Adobe Connect

Interface — Student View



Adobe Connect

Settings

- **Everybody: Audio Settings** Meeting ⟩ Audio Setup Wizard...
- **Audio** Menu bar ⟩ Audio ⟩ Microphone rights for Participants ✔
- Do not *Enable single speaker mode*
- **Drawing Tools** Share pod menu bar ⟩ Draw (1 slide/screen)
- Share pod menu bar ⟩ Menu icon ⟩ Enable Participants to draw ✔ gray
- Meeting ⟩ Preferences ⟩ Whiteboard ⟩ Enable Participants to draw ✔
- Cancel hand tool ... Do not enable green pointer...
- Meeting ⟩ Preferences ⟩ Attendees Pod ✘ *Raise Hand notification*
- Meeting ⟩ Preferences ⟩ Display Name Display First & Last Name
- **Cursor** Meeting ⟩ Preferences ⟩ General tab ⟩ Host Cursors ⟩ Show to all attendees ✔ (default *Off*)
- Meeting ⟩ Preferences ⟩ Screen Share ⟩ Cursor ⟩ Show Application Cursor
- **Webcam** Menu bar ⟩ Webcam ⟩ Enable Webcam for Participants ✔
- **Recording** Meeting ⟩ Record Meeting... ✔

Adobe Connect

Access

- **Tutor Access**

  TutorHome › M269 Website › Tutorials

  Cluster Tutorials › M269 Online tutorial room

  Tutor Groups › M269 Online tutor group room

  Module-wide Tutorials › M269 Online module-wide room

- **Attendance**

  TutorHome › Students › View your tutorial timetables

- **Beamer Slide Scaling** 440% (422 x 563 mm)

- **Clear Everyone's Status**

  Attendee Pod › Menu › Clear Everyone's Status

- **Grant Access** and send link via email

  Meeting › Manage Access & Entry › Invite Participants...

- **Presenter Only Area**

  Meeting › Enable/Disable Presenter Only Area

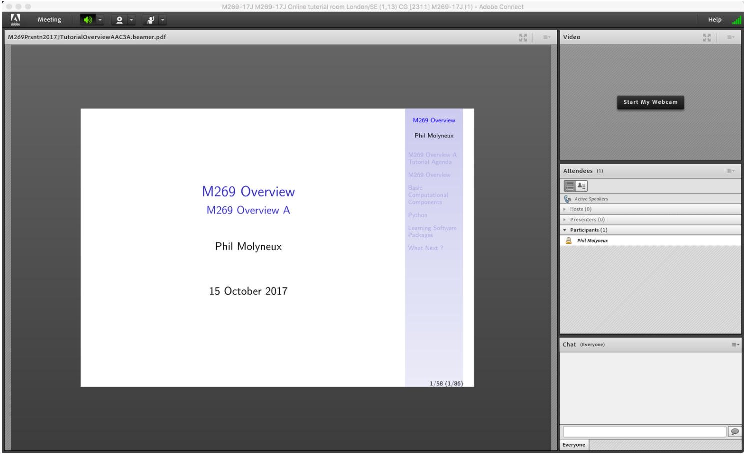
Adobe Connect

Keystroke Shortcuts

- Keyboard shortcuts in Adobe Connect
- **Toggle Mic** ⌘ + M (Mac), Ctrl + M (Win) (On/Disconnect)
- **Toggle Raise-Hand status** ⌘ + E
- **Close dialog box** ⎋ (Mac), Esc (Win)
- **End meeting** ⌘ + \

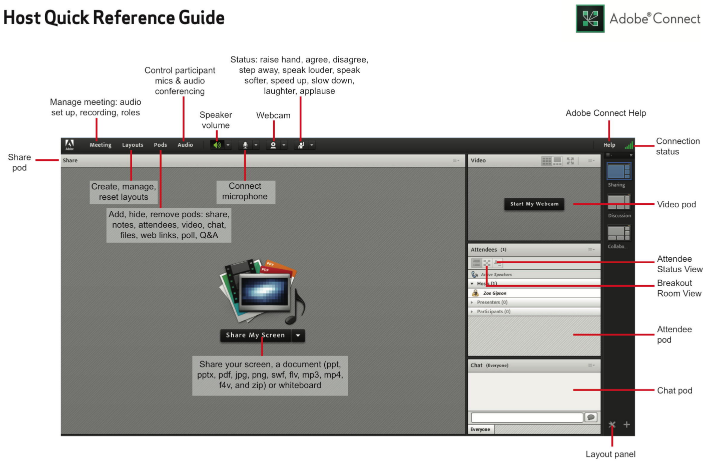
Adobe Connect Interface

Student View (default)



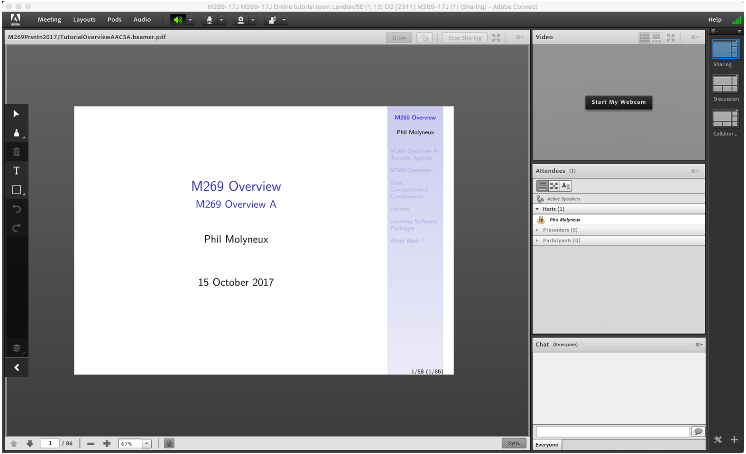
Adobe Connect Interface

Tutor View



Adobe Connect Interface

Tutor View



Adobe Connect Interface

Sharing Screen & Applications

- Share My Screen ⟩ Application tab ⟩ Terminal for Terminal
- Share menu ⟩ Change View ⟩ Zoom in for mismatch of screen size/resolution (Participants)
- (Presenter) Change to 75% and back to 100% (solves participants with smaller screen image overlap)
- Leave the application on the original display
- Beware blued hatched rectangles — from other (hidden) windows or contextual menus
- Presenter screen pointer affects viewer display — beware of moving the pointer away from the application
- First time: System Preferences ⟩ Security & Privacy ⟩ Privacy ⟩ Accessibility

Adobe Connect

Ending a Meeting

- *Notes for the tutor only*
- **Student:** Meeting › Exit Adobe Connect
- **Tutor:**
- **Recording** Meeting › Stop Recording ✔
- **Remove Participants** Meeting › End Meeting. . . ✔
  - Dialog box allows for message with default message:
  - *The host has ended this meeting. Thank you for attending.*
- **Recording availability** *In course Web site for joining the room, click on the eye icon in the list of recordings under your recording* — edit description and name
- **Meeting Information** Meeting › Manage Meeting Information — can access a range of information in Web page.
- **Attendance Report** see course Web site for joining room

Adobe Connect

Invite Attendees

- **Provide Meeting URL** Menu › Meeting › Manage Access & Entry › Invite Participants. . .
- **Allow Access without Dialog** Menu › Meeting › Manage Meeting Information provides new browser window with *Meeting Information* Tab bar › Edit Information
- Check *Anyone who has the URL for the meeting can enter the room*
- Default *Only registered users and accepted guests may enter the room*
- **Reverts to default next session but URL is fixed**
- Guests have blue icon top, registered participants have yellow icon top — same icon if URL is open
- See Start, attend, and manage Adobe Connect meetings and sessions

Adobe Connect

Layouts

- **Creating new layouts** example *Sharing* layout
- Menu › Layouts › Create New Layout... Create a New Layout dialog › Create a new blank layout and name it *PMolyMain*
- New layout has no Pods but does have Layouts Bar open (see Layouts menu)
- **Pods**
- Menu › Pods › Share › Add New Share and resize/position — initial name is *Share n*
- **Rename Pod** Menu › Pods › Manage Pods... Manage Pods › Select › Rename or Double-click & rename
- Add Video pod and resize/reposition
- Add Attendance pod and resize/reposition
- Add Chat pod — name it *PMolyChat* — and resize/reposition

Adobe Connect

Layouts

- Dimensions of **Sharing** layout (on 27-inch iMac)
  - Width of Video, Attendees, Chat column 14 cm
  - Height of Video pod 9 cm
  - Height of Attendees pod 12 cm
  - Height of Chat pod 8 cm
- **Duplicating Layouts** does *not* give new instances of the Pods and is probably not a good idea (apart from local use to avoid delay in reloading Pods)

Adobe Connect

Chat Pods

- **Format Chat text**
- Chat Pod ❯ menu icon ❯ My Chat Color
- Choices: Red, Orange, Green, Brown, Purple, Pink, Blue, Black
- Note: Color reverts to Black if you switch layouts
- Chat Pod ❯ menu icon ❯ Show Timestamps

Logic

Logics, Logicians, Notations

- A plethora of logics, proof systems, and different notations can be puzzling.
- Martin Davis, Logician When I was a student, even the topologists regarded mathematical logicians as living in outer space. Today the connections between logic and computers are a matter of engineering practice at every level of computer organization
  Davis (1995, Influences of mathematical logic on computer science in *The Universal Turing Machine A Half-Century Survey*, Springer , 1995)
- Various logics, proof systems , were developed well before programming languages and with different motivations,

Logic

Mathematics and Notation

- **Richard Feynman** We could, of course, use any notation we want; do not laugh at notations; invent them, they are powerful. In fact, mathematics is, to a large extent, invention of better notations.
  Feynman et al. (2011, The Feynman Lectures on Physics, 1963, Volume 1, chapter 17 Space-Time, section 17-5 Four-vector algebra)

- **Alfred North Whitehead** It is a profoundly erroneous truism, repeated by all copy-books and by eminent people when they are making speeches, that we should cultivate the habit of thinking of what we are doing. The precise opposite is the case. Civilization advances by extending the number of important operations which we can perform without thinking about them. Operations of thought are like cavalry charges in a battle — they are strictly limited in number, they require fresh horses, and must only be made at decisive moments.
  Whitehead (1911, An Introduction to Mathematics, 1911, chapter 5)

Logic

Logic and Programming Languages

- Turing machines, Von Neumann architecture and procedural languages Fortran, C, Java, Perl, Python, JavaScript — Hoare logic
- Resolution theorem proving and logic programming — Prolog
- Logic and database query languages — SQL (Structured Query Language) and QBE (Query-By-Example) are syntactic sugar for first order logic
- Lambda calculus and functional programming with Miranda, Haskell, ML, Scala
- Programming languages are formal systems — that is, *specialized logics*

Logic

Syntax, Semantics and Proof

- The **syntax** of a logic defines the acceptable strings in the language — *well-formed formulae (WFFs)*
- The **semantics** of a logic associates meaning to a formula
- The **proof theory** is concerned with rules for manipulating formulae.
- Classical logic includes Propositional logic and Predicate logic

Logic

Propositional Logic

- Propositional logic has statements (or propositional constants) which can be True or False
  - It is raining
  - The assignment is due on Thursday
  - The exam is three hours long
- The statements (propositions) can be combined with logical connectives (functions of the propositions)
  - $\neg$ negation ($\neg p$)
  - $\wedge$ conjunction, AND ($p \wedge q$)
  - $\vee$ disjunction, OR ($p \vee q$)
  - $\Rightarrow$ logical implication, IF... THEN... ($p \Rightarrow q$)
  - Only expressions built from the rules are WFFs
- Proof systems including Truth Tables and Natural Deduction
- Note that there was a choice of connectives — see Truth function — the set given is Functionally Complete but is not minimal — see later

Logic

Predicate Logic

- Predicate logic uses quantified variables over sets and predicates indicating relations between objects.
- $\forall x . P(x)$ for all x, $P(x)$ is True
- $\exists x . Q(x)$ for some x, $Q(x)$ is True (or, there exists at least one x)
- Also called *first order logic*
- Higher-order logic quantifies over predicates, sets of sets, . . . semantics more expressive but proof theories more complicated.

Logic

Using Logical Equivalences

- Unit 6 and chapters 2 and 3 of *Logic and the limits of computing* introduce *propositional* and *predicate* logic and some of the *equivalences* used in reasoning about statements.
- The following exercises ask you to prove the equivalence of some logic statements and the later exercises ask you to negate statements
- You can either think about them in English or translate them to statements in predicate logic and use the equivalences
- Which is the easiest ?
- And which is more reliable ?

Logic

Notation and Logical Equivalences

- We could define the notation for predicate calculus in a formal way and it is useful to eventually see we can make many of our definitions mechanical.
- At the start a formal definition can be intimidating until you have seen the usefulness of a formal approach.

Formula	
$P(t_1, t_2, \ldots, t_n)$	Predicate with arguments
$\neg p$	Negation of formula p
$\forall x$ **in** $X[p]$	Universal quantification
$\exists x$ **in** $X[p]$	Existential quantification
$p \wedge q$	Logical AND, conjunction
$p \vee q$	Logical OR, disjunction
$p \Rightarrow q$	Logical implication
(p)	Brackets

- Truth tables define the meaning of $\neg$, $\wedge$, $\vee$, $\Rightarrow$

Propositional Logic

meaning of $\neg$, $\wedge$, $\vee$, $\Rightarrow$

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

p	$\neg p$
T	F
F	T

p	q	$p \Rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

- **Exercise** Justify the truth table for $\Rightarrow$

Propositional Logic

Justification of Truth Table for $\Rightarrow$

p	q	$p \Rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

- The **True** values in the last two rows give students a lot of trouble
- What is going on ? This is a *negative definition*
- $p \Rightarrow q$ holds unless we have evidence to the contrary
- $\Rightarrow$ is one of the 16 possible truth functions of two boolean inputs
- In a typed programming language
- $\Rightarrow :: (\mathbb{B}, \mathbb{B}) \rightarrow \mathbb{B}$

Logical Equivalences

Negation and De Morgan

Negation

$$p \lor \neg p \equiv True$$

$$p \land \neg p \equiv False$$

$$\neg\neg p \equiv p$$

De Morgan

$$\neg(p \lor q) \equiv \neg p \land \neg q$$

$$\neg(p \land q) \equiv \neg p \lor \neg q$$

$$\neg \forall x[P(x)] \equiv \exists x[\neg P(x)]$$

$$\neg \exists x[P(x)] \equiv \forall x[\neg P(x)]$$

- **Question** Why has the author put the equivalence symbol ($\equiv$) in a different colour ?

Logical Equivalences

Equivalence Symbol ($\equiv$) in a different Colour

- The equivalence symbol ($\equiv$) is not a symbol in Propositional or Predicate Logic (in our notation)
- It is important to realise we have some notation to refer to notation in Logic
- This is common when we have proofs about logical statements
- Sadly most texts just use black and white
- And I haven't had time to do consistent colour coding (and would have to hack the package used for the proof tree layout)

Negation and De Morgan

Alice in Wonderland

- *White King* . . . Just look along the road, and tell me if you can see either of them.
- *Alice* I can see nobody on the road
- *White King* . . . To be able to see Nobody! And at that distance too!
- *Through the Looking Glass and What Alice Found There* Chp 7 *The Lion and the Unicorn*
- What was the day job of *Lewis Carroll* ?

Logical Equivalences

Rewriting $\Rightarrow$

Rewriting $\Rightarrow$

$$p \Rightarrow q \equiv \neg p \vee q$$

$$p \Leftrightarrow q \equiv (p \Rightarrow q) \wedge (q \Rightarrow p)$$

- **Exercise** Use a truth table to prove $p \Rightarrow q \equiv \neg p \vee q$

Logical Equivalences

Truth Table Proof of $p \Rightarrow q \equiv \neg p \vee q$

p	q	$p \Rightarrow q$	$\neg p \vee q$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

Logical Equivalences

$\vee, \wedge$ Laws

Distributive Laws

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

Associative Laws

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

Commutative Laws

$$p \vee q \equiv q \vee p$$
$$p \wedge q \equiv q \wedge p$$

Logical Equivalences

Extended Commutativity

Extended Commutativity

$\forall x[\forall y[P(x,y)]] \equiv \forall y[\forall x[P(x,y)]]$ often written $\forall x, \forall y[P(x,y)]$

$\exists x[\exists y[P(x,y)]] \equiv \exists y[\exists x[P(x,y)]]$ often written $\exists x, \exists y[P(x,y)]$

Logic Exs
Quantifiers Q 1

- Is it the case that $\exists x[\forall y[P(x,y)]] \equiv \forall y[\exists x[P(x,y)]]$?
- If not, give counter examples.
- Does $\forall y[\exists x[P(x,y)]] \Rightarrow \exists x[\forall y[P(x,y)]]$
- or does $\exists x[\forall y[P(x,y)]] \Rightarrow \forall y[\exists x[P(x,y)]]$

Go to Quantifiers Soln 1

Logic Exs

Quantifiers Soln 1

- It is a common error to think they are equivalent
- See Maths Stack Exchange: Is $\forall x \exists y Q(x, y)$ the same as $\exists y \forall x Q(x, y)$?
- See Maths Stack Exchange: What does $\forall x \exists y (x + y = 0)$ mean ?
- Let $P(x, y)$ be $x + y = 0$
- Then $\forall x[\exists y[P(x, y)]]$ is true — say this in English
- but $\exists y[\forall x[P(x, y)]]$ is not true

Go to Quantifiers Q 1

Logical Equivalences

Other Equivalences

Identity Laws

$$p \vee \text{False} \equiv p$$
$$p \wedge \text{True} \equiv p$$
$$p \vee \text{True} \equiv \text{True}$$
$$p \wedge \text{False} \equiv \text{False}$$

Idempotent Laws

$$p \vee p \equiv p$$
$$p \wedge p \equiv p$$

Absorption Laws

$$p \vee (p \wedge q) \equiv p$$
$$p \wedge (p \vee q) \equiv p$$

Logic Exs

Absorption Laws Q 1

- Prove the *Absorption Laws* using truth tables
- Prove the *Absorption Laws* using other equivalences

Go to Absorption Laws Soln 1

Logic Exs
Absorption Laws Soln 1

- Truth table for $p \vee (p \wedge q) \equiv p$

p	q	$p \wedge q$	$p \vee (p \wedge q)$
T	T	T	T
T	F	F	T
F	T	F	F
F	F	F	F

Go to Absorption Laws Q 1

Logic Exs

Absorption Laws Soln 1 (B)

- Equivalences proof for $p \vee (p \wedge q) \equiv p$
- $p \vee (p \wedge q)$
- $\rightarrow (p \vee p) \wedge (p \vee q)$ by *Distributive laws*
- $\rightarrow p \wedge (p \vee q)$ by *Idempotent laws*
- This could go round in circles — *start again*.

Go to Absorption Laws Q 1

Logic Exs

Absorption Laws Soln 1 (C)

- Equivalences proof for $p \vee (p \wedge q) \equiv p$
- $p \vee (p \wedge q)$
- $\rightarrow (p \wedge T) \vee (p \wedge q)$ by *Identity laws* *Eureka step*
- $\rightarrow p \wedge (T \vee q)$ by *Distributive laws*
- $\rightarrow p \wedge T$ by *Identity & Commutative laws*
- $\rightarrow p$ by *Identity laws*

Go to Absorption Laws Q 1

Truth Function

Truth Function References

- The following notes illustrate the 16 binary functions of two Boolean variables
- See Truth function
- See Functional completeness
- See Sheffer stroke
- See Logical NOR

Truth Function

Table of Binary Truth Functions

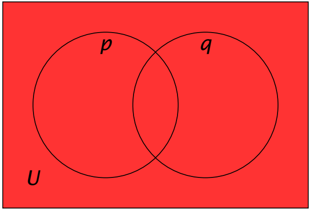
p	q	$\top$	$p \vee q$	$p \Leftarrow q$	p	$p \Rightarrow q$	q	$p \Leftrightarrow q$	$p \wedge q$
T	T	T	T	T	T	T	T	T	T
T	F	T	T	T	T	F	F	F	F
F	T	T	T	F	F	T	T	F	F
F	F	T	F	T	F	T	F	T	F

p	q	$\bot$	$p \mathbin{\overline{\vee}} q$	$p \nLeftarrow q$	$\neg p$	$p \nRightarrow q$	$\neg q$	$p \nLeftrightarrow q$	$p \mathbin{\overline{\wedge}} q$
T	T	F	F	F	F	F	F	F	F
T	F	F	F	F	F	T	T	T	T
F	T	F	F	T	T	F	F	T	T
F	F	F	T	F	T	F	T	F	T

Truth Function

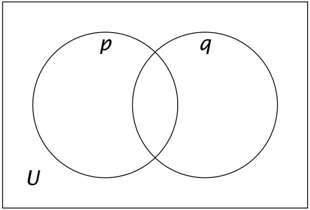
Tautology/Contradiction

- Tautology True, $\top$, *Top*



p	q	$\top$
T	T	T
T	F	T
F	T	T
F	F	T

- Contradiction False, $\perp$, *Bottom*

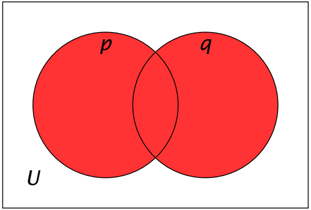


p	q	$\perp$
T	T	F
T	F	F
F	T	F
F	F	F

Truth Function

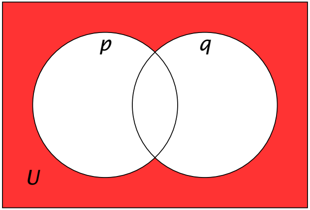
Disjunction/Joint Denial

- Disjunction OR, $p \vee q$



p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

- Joint Denial NOR, $p \mathbin{\overline{\vee}} q$, $p \downarrow q$, *Pierce's arrow*

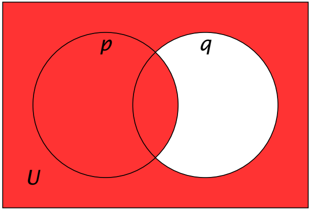


p	q	$p \downarrow q$
T	T	F
T	F	F
F	T	F
F	F	T

Truth Function

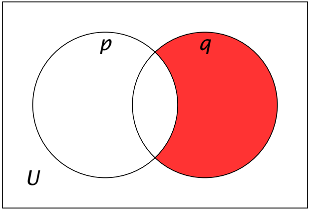
Converse Implication/Converse Nonimplication

- Converse Implication $p \Leftarrow q$



p	q	$p \Leftarrow q$
T	T	T
T	F	T
F	T	F
F	F	T

- Converse Nonimplication $p \nLeftarrow q$

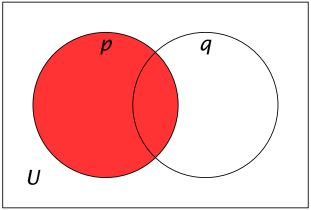


p	q	$p \nLeftarrow q$
T	T	F
T	F	F
F	T	T
F	F	F

Truth Function

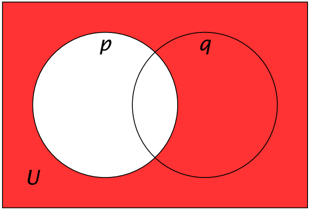
Proposition p/Negation of p

- Proposition p p



p	q	p
T	T	T
T	F	T
F	T	F
F	F	F

- Negation of p $\neg p$

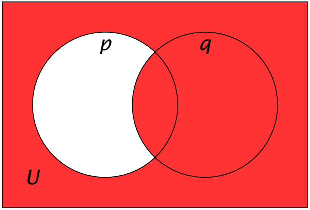


p	q	$\neg p$
T	T	F
T	F	F
F	T	T
F	F	T

Truth Function

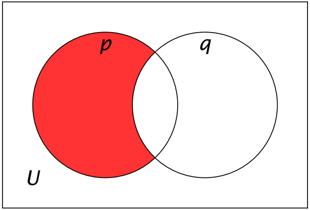
Material Implication/Material Nonimplication

- Material Implication $p \Rightarrow q$



p	q	$p \Rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

- Material Nonimplication $p \not\Rightarrow q$

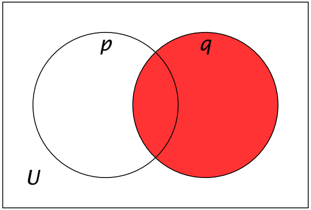


p	q	$p \not\Rightarrow q$
T	T	F
T	F	T
F	T	F
F	F	F

Truth Function

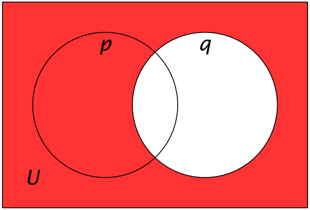
Proposition q/Negation of q

- Proposition q q



p	q	q
T	T	T
T	F	F
F	T	T
F	F	F

- Negation of q $\neg q$

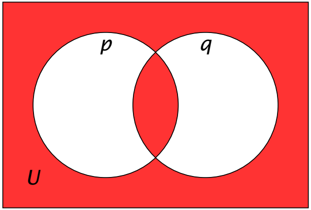


p	q	$\neg q$
T	T	F
T	F	T
F	T	F
F	F	T

Truth Function

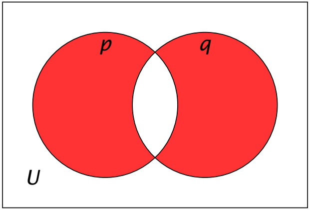
Biconditional/Exclusive disjunction

- **Biconditional** If and only if, IFF, $p \Leftrightarrow q$



p	q	$p \Leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

- **Exclusive disjunction XOR**, $p \not\Leftrightarrow q$, $p \veebar q$

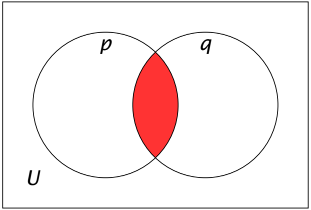


p	q	$p \not\Leftrightarrow q$
T	T	F
T	F	T
F	T	T
F	F	F

Truth Function

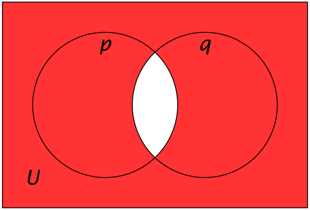
Conjunction/Alternative denial

- Conjunction AND, $p \wedge q$



p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

- Alternative denial NAND, $p \barwedge q$, $p \uparrow q$, *Sheffer stroke*



p	q	$p \uparrow q$
T	T	F
T	F	T
F	T	T
F	F	T

Propositional Calculus

Functional Completeness, Boolean Programming

- **Functionally complete** set of connectives is one which can be used to express all possible connectives
- $p \Rightarrow q \equiv \neg p \vee q$ so we could just use $\{\neg, \wedge, \vee\}$
- **Boolean programming** — we have to have a functionally complete set but choose more to make the programming easier
- *Expressiveness* is an issue in programming language design

Propositional Calculus

NAND, NOR

- **NAND** $\boldsymbol{p \barwedge q}$, $\boldsymbol{p \uparrow q}$, Sheffer stroke
- **NOR** $\boldsymbol{p \mathbin{\overline{\vee}} q}$, $\boldsymbol{p \downarrow q}$, Pierce's arrow
- Both $\{\uparrow\}, \{\downarrow\}$ are functionally complete — verify:

$$
\begin{aligned}
\neg p &\equiv p \uparrow p \\
p \wedge q &\equiv \neg(p \uparrow q) \equiv (p \uparrow q) \uparrow (p \uparrow q) \\
p \vee q &\equiv (p \uparrow p) \uparrow (q \uparrow q) \\
p \Rightarrow q &\equiv ((p \uparrow p) \uparrow (p \uparrow p)) \uparrow (q \uparrow q) \\
\neg p &\equiv p \downarrow p \\
p \wedge q &\equiv (p \downarrow p) \downarrow (q \downarrow q) \\
p \vee q &\equiv \neg(p \downarrow q) \equiv (p \downarrow q) \downarrow (p \downarrow q) \\
p \Rightarrow q &\equiv ((p \downarrow p) \downarrow q) \downarrow ((p \downarrow p) \downarrow q)
\end{aligned}
$$

- Not a novelty — the Apollo Guidance Computer was implemented in NOR gates alone.

Logic Exs

Negation Qs

- In each of the following questions $P(x, y, \ldots)$ denotes a statement involving objects $x, y, \ldots$. Construct the negation of each of the following propositions.
  1. $P(x)$ is true for all x.
  2. $P(x, y)$ is true for all x and all y.
  3. There is at least one x such that $P(x, y)$ is true for all y.
  4. Given any x there is at least one y such that $P(x, y)$ is false.
  5. Given any x there is at least one y such that $P(x, y, z)$ is true for all z.
  6. Given any x there is precisely one y such that $P(x, y, z)$ is true for at least one z.
  7. Given any x there is at least one y such that $P(x, y, z)$ is true for at most one z.

Go to Negation Solns

Logic Exs

Negation Solns

- Our strategy:
- Translate the English statements into our *formal language*
- Use the equivalence rules to simplify the negation
- Finally translate back into English

Go to Negation Qs

Using Logical Equivalences

Answers — 1

- Our strategy is to translate the English statements into our formal language, use the equivalence rules to simplify the negation and finally translate back to English
- $P(x)$ is true for all x.
- **Translate** $\forall x[P(x)]$
- **Negation** $\neg(\forall x[P(x)])$
- **Simplify** $\exists x[\neg P(x)]$
- **Translate** $P(x)$ is false for at least one x

Go to Negation Qs

Using Logical Equivalences

Answers — 2

- $P(x, y)$ is true for all x and all y.
- **Translate** $\forall x, \forall y[P(x, y)]$
- **Negation** $\neg(\forall x, \forall y[P(x, y)])$
- **Simplify** $\exists x, \exists y[\neg P(x, y)]$
- **Translate** $P(x, y)$ is false for at least one x and one y

Go to Negation Qs

Using Logical Equivalences

Answers — 3

- There is at least one x such that $P(x, y)$ is true for all y.
- **Translate** $\exists x [\forall y [P(x, y)]]$
- **Negation** $\neg(\exists x [\forall y [P(x, y)]])$
- **Simplify** $\forall x [\exists y [\neg P(x, y)]]$
- **Translate** Given any x there is at least one y (possibly depending on x) such that $P(x, y)$ is false

Go to Negation Qs

Using Logical Equivalences

Answers — 4

- Given any x there is at least one y such that $P(x, y)$ is false.
- **Translate** $\forall x[\exists y[\neg P(x, y)]]$
- **Negation** $\neg(\forall x[\exists y[\neg P(x, y)]])$
- **Simplify** $\exists x[\forall y[P(x, y)]]$
- **Translate** There is at least one x such that for all y, $P(x, y)$ is true.

▸ Go to Negation Qs

Using Logical Equivalences

Answers — 5

- Given any x there is at least one y such that $P(x, y, z)$ is true for all z.
- **Translate** $\forall x[\exists y[\forall z[P(x, y, z)]]]$
- **Negation** $\neg(\forall x[\exists y[\forall z[P(x, y, z)]]])$
- **Simplify** $\exists x[\forall y[\exists z[\neg[P(x, y, z)]]]]$
- **Translate** There is at least one x such that for all y there is at least one z (possibly depending on y) such that $P(x, y, z)$ is false.

Go to Negation Qs

Answers — 6

- Given any x there is at precisely one y such that $P(x, y, z)$ is true for at least one z.
- **Translate** $\forall x[\exists! y[\exists z[P(x, y, z)]]]$ *Note* $\exists!$ *for exactly one*
- ***Eureka Step*** *Exactly one* means *At least one and not two or more*
- **Expand** $\forall x[\exists y[\exists z[P(x, y, z)]]$
  $\wedge$
  $\neg(\exists y_1, \exists y_2[y_1 \neq y_2 \wedge \exists z[P(x, y_1, z)] \wedge \exists z[P(x, y_2, z)]]])]$
- **Negation** $\neg(\forall x[\exists y[\exists z[P(x, y, z)]]$
  $\wedge$
  $\neg(\exists y_1, \exists y_2[y_1 \neq y_2 \wedge \exists z[P(x, y_1, z)] \wedge \exists z[P(x, y_2, z)]]])])$

Go to Negation Qs

Using Logical Equivalences

Answers — 6a

- *Negate* Given any x there is at precisely one y such that $P(x, y, z)$ is true for at least one z.
- **Negation** $\neg(\forall x[\exists y[\exists z[P(x, y, z)]]$
  $\wedge$
  $\neg(\exists y_1, \exists y_2[y_1 \neq y_2 \wedge \exists z[P(x, y_1, z)] \wedge \exists z[P(x, y_2, z)]]])])$
- **Simplify** $\exists x[\forall y[\forall z[\neg P(x, y, z)]]$
  $\vee$
  $(\exists y_1, \exists y_2[y_1 \neq y_2 \wedge \exists z[P(x, y_1, z)] \wedge \exists z[P(x, y_2, z)]]])]$
- **Translate** For at least one x there is either no y and z such that $P(x, y, z)$ is true or there are at least two y such that there exists a z (possible depending on the y) such that $P(x, y, z)$ is true.

Go to Negation Qs

Using Logical Equivalences

Answers — 7

- Given any x there is at least one y such that $P(x, y, z)$ is true for at most one z.
- **Translate** $\forall x[\exists y[$ for at most one $z[P(x, y, z)]]]$
- *Note lack of notation here*
- ***Eureka Step*** *At most one* means *none or exactly one* (we will have a lot of code here)
- **Expand** $\forall x[\exists y[\neg \exists z[P(x, y, z)]$

$$\vee$$

$$(\exists z[P(x, y, z)]$$

$$\wedge$$

$$\neg(\exists z_1, \exists z_2[z_1 \neq z_2 \wedge P(x, y, z_1) \wedge P(x, y, z_2)]))]]$$

Go to Negation Qs

Using Logical Equivalences

Answers — 7a

- *Negate* Given any x there is at least one y such that $P(x, y, z)$ is true for at most one z.
- **Negation** $\neg(\forall x[\exists y[\neg\exists z[P(x, y, z)]$
  $\lor$
  $(\exists z[P(x, y, z)]$
  $\land \neg(\exists z_1, \exists z_2[z_1 \neq z_2 \land P(x, y, z_1) \land P(x, y, z_2)]))]])$
- **Simplify** $\exists x[\forall y[\exists z[P(x, y, z)]$
  $\land$
  $(\forall z[\neg P(x, y, z)]$
  $\lor (\exists z_1, \exists z_2[z_1 \neq z_2 \land P(x, y, z_1) \land P(x, y, z_2)]))]]$

Go to Negation Qs

Using Logical Equivalences

Answers — 7b

- *Negate* Given any x there is at least one y such that $P(x, y, z)$ is true for at most one z.
- **Simplify** $\exists x[\forall y[\exists z[P(x, y, z)]$
  $\wedge$
  $(\forall z[\neg P(x, y, z)]$
  $\vee (\exists z_1, \exists z_2[z_1 \neq z_2 \wedge P(x, y, z_1) \wedge P(x, y, z_2)]))]]$
- **Simplify back up** $\exists x[\forall y[\exists z[P(x, y, z)]$
  $\wedge$
  $(\neg \exists z[P(x, y, z)]$
  $\vee (\exists z_1, \exists z_2[z_1 \neq z_2 \wedge P(x, y, z_1) \wedge P(x, y, z_2)]))]]$
- Now use the *Distributive Law*

Go to Negation Qs

Using Logical Equivalences

Answers — 7c

- *Negate* Given any x there is at least one y such that $P(x, y, z)$ is true for at most one z.
- **Simplify back up** $\exists x[\forall y[\exists z[P(x,y,z)]$
  $\land$
  $(\neg\exists z[P(x,y,z)]$
  $\lor (\exists z_1, \exists z_2[z_1 \neq z_2 \land P(x,y,z_1) \land P(x,y,z_2)]))]]$
- **Distributive Law** $\exists x[\forall y[$
  $(\exists z[P(x,y,z)] \land \neg\exists z[P(x,y,z)])$
  $\lor$
  $(\exists z[P(x,y,z)]$
  $\land (\exists z_1, \exists z_2[z_1 \neq z_2 \land P(x,y,z_1) \land P(x,y,z_2)]))]]$
- Now use the *Negation Law*

Go to Negation Qs

Using Logical Equivalences

Answers — 7d

- *Negate* Given any x there is at least one y such that $P(x, y, z)$ is true for at most one z.
- **Negation Law** $\exists x[\forall y[$
  $(False)$
  $\vee$
  $(\exists z[P(x, y, z)]$
  $\wedge (\exists z_1, \exists z_2[z_1 \neq z_2 \wedge P(x, y, z_1) \wedge P(x, y, z_2)])))]]$
- **Absorption Law** $\exists x[\forall y[\exists z[P(x, y, z)]$
  $\wedge (\exists z_1, \exists z_2[z_1 \neq z_2 \wedge P(x, y, z_1) \wedge P(x, y, z_2)])]]$
- **Translate** There exists at least one x such that for all y, $P(x, y, z)$ is true for more than one z

Go to Negation Qs

Logical Equivalences

Comments on Exercises

- *Plain English* is never that plain
- Consider: *Fruit flies like a banana*
- A good notation should help clarify thought — see Whitehead quote
- Note how the ordering of clauses in English can lead to ambiguity — does a z depend on a previous y, for example — hence we need a precisely defined notation to determine scope of variables
- Using a *formal language* can help the manipulation but there is no free lunch
- You need a decent editor to check your syntax and bracket matching — software exists to help this — see Wikipedia Proof Assistant

Go to Negation Qs

Negation Exercises

Further Points

- The above exercises were just about the only instruction on Propositional and Predicate Logic I had as an undergraduate (in Physics and Maths, Sussex University)
- Below are copies of the original question sheet and my answers with markers comments.
- Notice that my mistakes mainly involved getting the order of the English clauses wrong — in English, it is harder to *see* the scope of names.
- I also confused the colloquial *at least one x or at least one y* for *at least one x and at least one y*

Logic Exercises

Further Points — Exercises (1)

1 Oct

ANALYSIS (Mathematics subject 4a)

DEFINITIONS

1. If P and Q are propositions (which may be true or false) we say that 'P implies Q' (in symbols P => Q) if the truth of P ensures the truth of Q.

Alternatively we have, a priori, the four possibilities:

(i) P and Q are both true

(ii) P and Q are both false

(iii) P is false and Q is true

(iv) P is true and Q is false

'P implies Q' means that the fourth alternative, but only the fourth, is excluded.

The point here is that if P is false 'P implies Q' provides no information whatever about (or imposes no restriction on) Q.

ii. If 'P => Q' and 'Q => P' we say that 'P is equivalent to Q' and write P <=> Q.

iii. The statement 'The proposition Q is false' is called the negation of the statement 'The proposition Q is true'.

Reference

Scott, D.B. and Tims, S.R. Mathematical Analysis: An introduction §0.2 p3 - 9, C.U.P 60/-

(also recommended for the Analysis (4a) course)

Logic Exercises

Further Points — Exercises (2)

Exercises (to be attempted in Registration week)

In each of the following questions two conditions are given. Decide in each case whether condition (a) is necessary or sufficient (or both) or neither, for (b), and then answer the same question with (a) and (b) interchanged. The usual notation for a triangle is employed in questions 1, 2, 3 and 8.

1. (a) The angle A is obtuse
   (b) a exceeds each of b and c.

2. (a) A exceeds $\pi/3$
   (b) a exceeds each of b and c.

3. (a) a^2 exceeds b^2+c^2
   (b) A is obtuse.

4. (a) $x^2-3x+2 = 0$
   (b) $x=1$.

5. (a) $\frac{y_1}{x_1} = \frac{y_2}{x_2} = \frac{y_3}{x_3}$ and $x_1x_2x_3 \neq 0$.
   (b) the three points (x_1,y_1), (x_2,y_2), (x_3,y_3) are collinear.

In each of the preceding five and the next three following questions does (a) imply (b) or conversely does (b) imply (a)?

6. (a) x exceeds 2
   (b) x exceeds 1.

7. (a) $x^4-5x^2+4 = 0$
   (b) $x=1$ or $x=2$.

8. (a) $a > b+c$
   (b) A is obtuse.

Negation Exercises

Further Points — Exercises (3)

- 2 -

In each of the following questions P(x,y, ...) denotes a statement involving objects x,y, ... Construct the negation of each of the following propositions.

9. P(x) is true for all x.

10. P(x,y) is true for all x and all y.

11. There is at least one x such that P(x,y) is true for all y.

12. Given any x there is at least one y such that P(x,y) is false.

•13. Given any x there is at least one y such that P(x,y,z) is true for all z.

•14. Given any x (there is precisely one y (such that P(x,y,z) is true for at least one z.))

•15. Given any x (there is at least one y (such that P(x,y,z) is true for at most one z.))

∃ x s.t given any y P(x y z) is ~~false~~ true for at least two z

(• more difficult question)

Negation Exercises

Further Points — Answers (1)

1

S.P. MOLYNEUX 20 Oct. 1968

ANALYSIS (MATHS 4A)

1. a) is sufficient for b)
   b) is necessary for a)
   a) implies b) b) does not imply a)
2. a) is necessary for b)
   b) is sufficient for a)
   b) implies a) a) does not imply b)
3. a) is both necessary and sufficient for b)
   b) " " " " " " a)
   a) is equivalent to b)
4. a) is necessary for b)
   b) is sufficient for a)
   b) implies a) a) does not imply b)
5. a) is sufficient for b)
   b) is necessary for a)
   a) implies b) b) does not imply a)
6. a) implies b)
7. b) implies a) (solution of a) is $x = \pm 1$ or ± 2)
8. a) implies b)

Negation Exercises

Further Points — Answers (2)

The negations are

9. $P(x)$ is false for at least one x.

10. $P(x, y)$ is false for at least one x ~~or~~ and one y.

11. There is at least one y such that $P(x, y)$ is false for all x.
See end.

12. ~~There is at least one y x such that $P(x,y)$ is true for all y~~
There is at least one y such that $P(x, y)$ is false for all x See end.

13. For any y there is at least one x such that $P(x, y, z)$ is false for at least one z

14. For at least one ~~value value of~~ x there is no y such that $P(x, y, z)$ is false for all z or for at least one x there is more than one y such that $P(x, y, z)$ is false for all z

(obtained by rewriting the original statement as: Given any x there is one y such that $P(x, y, z)$ is true for at least one z and the y referred to is ~~a partic~~ unique)

15. The original statement rewritten is:
Given any x there is at least one y such that $P(x, y, z)$ is true for no z or precisely one z.
$\therefore$ the negation is:
Given any y there is at least one x such that $P(x, y, z)$ is false for at least one z and (no z or more than one z)

Negation Exercises

Further Points — Answers (3)

2

S.P. MOLYNEUX — 3 OCT. 1968

ANALYSIS 4A

15 contd.

But in the above statement "at least one y" excludes the possibility of "no z"

∴ the negation is:

Given any y there is at least one x such that $P(x,y,z)$ is false for at least one z and more than one z

∴ The negation is:

Given any y there is at least one x such that $P(x,y,z)$ is false for more than one z

14. For at least one x there is more than one y †. such that $P(x,y,z)$ is ~~true~~ false for ~~more~~ any z

TRY AGAIN.

15. There exists an x such that for any y $P(x,y,z)$ is true for more than one z. ✓

Question 11.

"There is at least one x s.t. $P(x,y)$ is true for all y."

Negation: Given any x ∃ at least one y (possibly depending upon x) s.t. $P(x,y)$ is false.

Your negation implies that the <u>same</u> y serves for all x.

Consider the example: There is at least one x s.t. $(x-y)^2 > 0$ for all y. (This is false because given any x ∃ a y $(=x)$ s.t. $(x-y)^2 = 0$ i.e. $(x-y)^2 \not> 0$.) But it is also false that ∃ an x s.t. $(x-y)^2 \not> 0$ for all y & we cannot have a statement & its negation simultaneously false.

Negation Exercises

Further Points — Answers (4)

Question 12.
"Given any x $\exists$ at least one y s.t. $P(x,y)$ is false"
Negation is "$\exists$ an x s.t. $P(x,y)$ is true for all y"
(You had it correct first time).

"There is at least one y s.t. $P(x,y)$ is false for all x"
Negation is: "Given any y $\exists$ an x s.t. $P(x,y)$ is true."

Question 13 (A) "Given any x $\exists$ at least one y s.t. $P(x,y,z)$ is true for all z."
Negation is (1) "$\exists$ an x s.t. for all y $\exists$ z (possibly depending upon y) s.t. $P(x,y,z)$ is false".

(2) "For any y $\exists$ at least one x s.t. $P(x,y,z)$ is false for at least one z"
Negation is (B) "$\exists$ a y s.t. $P(x,y,z)$ is true for all x and all ~~z~~ y" Since (A) & (B) are not the same (2) cannot be the negation of (A).

Predicate Logic

Interpretations

- An interpretation is an assignment of meaning to the symbols of a formal language
- An interpretation often (but not always) provides a way to determine the truth values of a sentence in a formal language.
- If an interpretation assigns the value True to a sentence or theory, the interpretation is called a *model* of that sentence or theory.
- The domain is the set of all the objects being discussed.
- An interpretation assigns an object in the domain to each of the constants in the logic, and an n-ary relation on the domain to each n-ary predicate
- See Definition 12 in the Unit 6, 7 Reader

Logic Exs
Interpretations Q 1

- Given the domain $\mathcal{D} = \{\text{Adam}, \text{Milton}, \text{Joan}\}$
- Consider $\forall X.((\text{banker}(X) \land \text{inHedgeFund}(X)) \Rightarrow \text{sellingShort}(X))$
- In which of the following interpretations is the formula True ?

(a)

- $\mathcal{I}(\text{banker}) = \{\text{Adam}, \text{Milton}, \text{Joan}\}$
- $\mathcal{I}(\text{inHedgeFund}) = \{\text{Milton}, \text{Joan}\}$
- $\mathcal{I}(\text{sellingShort}) = \{\text{Milton}, \text{Joan}\}$

(b)

- $\mathcal{I}(\text{banker}) = \{\text{Adam}, \text{Milton}, \text{Joan}\}$
- $\mathcal{I}(\text{inHedgeFund}) = \emptyset$ ($\emptyset$ denotes the *empty set*)
- $\mathcal{I}(\text{sellingShort}) = \emptyset$

(c)

- $\mathcal{I}(\text{banker}) = \{\text{Adam}\}$
- $\mathcal{I}(\text{inHedgeFund}) = \{\text{Adam}\}$
- $\mathcal{I}(\text{sellingShort}) = \{\text{Joan}\}$

(d)

- $\mathcal{I}(\text{banker}) = \{\text{Milton}\}$
- $\mathcal{I}(\text{inHedgeFund}) = \{\text{Adam}\}$
- $\mathcal{I}(\text{sellingShort}) = \{\text{Joan}\}$

Go to Interpretations Soln 1

Logic Exs

Interpretations Soln 1

(a) is True

(b) is True

(c) is False

(d) is True

- Give reasons for each of the above answers

Go to Interpretations Q 1

Logical Arguments

Validity and Justification

- There are two ways to model what counts as a logically good argument:
  - the **semantic** view
  - the **syntactic** view
- The notion of a valid argument in propositional logic is rooted in the semantic view.
- It is based on the semantic idea of interpretations: assignments of truth values to the propositional variables in the sentences under discussion.
- A *valid argument* is defined as one that preserves truth from the premises to the conclusions
- The syntactic view focuses on the syntactic form of arguments.
- Arguments which are correct according to this view are called *justified arguments*.

Logical Arguments

Proof Systems, Soundness, Completeness

- Semantic validity and syntactic justification are different ways of modelling the same intuitive property: whether an argument is logically good.
- A proof system is *sound* if any statement we can prove (justify) is also valid (true)
- A proof system is *adequate* if any valid (true) statement has a proof (justification)
- A proof system that is sound and adequate is said to be *complete*
- Propositional and predicate logic are *complete* — arguments that are valid are also justifiable and vice versa
- Unit 7 section 2.4 describes another logic where there are valid arguments that are not justifiable (provable)

Logical Arguments

Valid arguments

- Unit 6 defines valid arguments with the notation $\begin{array}{c} P_1 \\ \vdots \\ P_n \\ \hline C \end{array}$
- The argument is *valid* if and only if the value of C is *True* in each interpretation for which the value of each premise P_i is *True* for $1 \leq i \leq n$
- In some texts you see the notation $\{P_1, \ldots, P_n\} \vDash C$
- The expression denotes a *semantic sequent* or *semantic entailment*
- The $\vDash$ symbol is called the *double turnstile* and is often read as *entails* or *models*
- In LaTeX $\vDash$ and $\models$ are produced from `\vDash` and `\models` — see also the *turnstile* package
- In Unicode $\vDash$ is called *TRUE* and is U+22A8, HTML ⊨

Logical Arguments

Valid arguments — Tautology

- The argument $\{\} \vDash C$ is valid if and only if C is *True* in all interpretations
- That is, if and only if C is a tautology
- Beware different notations that mean the same thing
  - Alternate symbol for empty set: $\emptyset \vDash C$
  - Null symbol for empty set: $\vDash C$
  - Original M269 notation with null axiom above the line: $\overline{C}$

- Definition 7.1 An argument $\{P_1, P_2, \ldots, P_n\} \vdash C$ is a justified argument if and only if either the argument is an instance of an axiom or it can be derived by means of an inference rule from one or more other justified arguments.
- Axioms

$$\Gamma \cup \{\boldsymbol{A}\} \vdash \boldsymbol{A} \text{ (axiom schema)}$$

- This can be read as: any formula $\boldsymbol{A}$ can be derived from the assumption (premise) of $\{\boldsymbol{A}\}$ itself
- The $\vdash$ symbol is called the *turnstile* and is often read as *proves*, denoting *syntactic entailment*
- In LaTeX $\vdash$ is produced from \vdash
- In Unicode $\vdash$ is called *RIGHT TACK* and is U+22A2, HTML ⊢

Justified Arguments

Question 1

- Show that the argument $\{P \wedge Q, S, T\} \vdash P \wedge Q$ s justified, by showing that it is an instantiation of the axiom schema.

Justified Arguments

Answer 1

- Suppose that, in the axiom schema $\Gamma \cup \{\boldsymbol{A}\} \vdash \boldsymbol{A}$, we instantiate Γ with $\{S, T\}$ and $\boldsymbol{A}$ with $P \land Q$
- Then we get the axiom $\{S, T\} \cup \{P \land Q\} \vdash P \land Q$
- Since the union $\{S, T\} \cup \{P \land Q\}$ is equal to $\{P \land Q, S, T\}$ the axiom can be written $\{P \land Q, S, T\} \vdash P \land Q$
- We use the following single line to record that the argument is justified because it is an instantiation of the axiom schema:

  1. $\{P \land Q, S, T\} \vdash P \land Q$ [Axiom]

Justified Arguments

Answer 1 — Discussion

- **Discussion** We could equally well have instantiated Γ with $\{S, T, P \wedge Q\}$ since $\{S, T, P \wedge Q\} \cup \{P \wedge Q\}$ is equal to $\{P \wedge Q, S, T\}$
- That is, a union does not produce duplicate elements.
- Notice that we begin the instantiation with a straightforward textual substitution, then simplify an expression involving sets and set operators.

Logic

Justified Arguments

- Section 2.3 of Unit 7 (not the Unit 6, 7 Reader) gives the inference rules for $\rightarrow$, $\wedge$, and $\vee$ — only dealing with *positive propositional logic* so not making use of negation — see List of logic systems
- Usually (Classical logic) have a functionally complete set of logical connectives — that is, every binary Boolean function can be expressed in terms the functions in the set

Justified Arguments

Inference Rules — Notation

- Inference rule notation:

$$\frac{\mathit{Argument}_1 \quad \ldots \quad \mathit{Argument}_n}{\mathit{Argument}}\ (\mathit{label})$$

Justified Arguments

Inference Rules — Conjunction

- $\dfrac{\Gamma \vdash \boldsymbol{A} \quad \Gamma \vdash \boldsymbol{B}}{\Gamma \vdash \boldsymbol{A} \land \boldsymbol{B}}$ ($\land$-introduction)
- $\dfrac{\Gamma \vdash \boldsymbol{A} \land \boldsymbol{B}}{\Gamma \vdash \boldsymbol{A}}$ ($\land$-elimination left)
- $\dfrac{\Gamma \vdash \boldsymbol{A} \land \boldsymbol{B}}{\Gamma \vdash \boldsymbol{B}}$ ($\land$-elimination right)

Inference Rules

Conjunction — Example

- Show that the argument $\{P, Q\} \vdash P \wedge Q$ is justified.
- **Answer**

1.	$\{P, Q\} \vdash P$	[Axiom]
2.	$\{P, Q\} \vdash Q$	[Axiom]
3.	$\{P, Q\} \vdash P \wedge Q$	[1,2, $\wedge$-I]

- **Discussion** Each line consists of a number, an argument, and a justification. The axiom schema is the justification for line 1 and line 2, while line 3 is justified by applying $\wedge$-introduction to lines 1 and 2
- The order is 1 then 2 rather than 2 then 1, corresponding to reading the first line of the rule from left to right.
- The lines above are called a *proof* of the argument $\{P, Q\} \vdash P \wedge Q$
- They are a step-by-step trace of how the argument in the final line is justified.

Justified Arguments

Question 2

- Give a proof of the argument $\{P, Q, R \vee S\} \vdash P \wedge Q$

Justified Arguments

Answer 2

1.	$\{P, Q, R \vee S\} \vdash P$	[Axiom]
2.	$\{P, Q, R \vee S\} \vdash Q$	[Axiom]
3.	$\{P, Q, R \vee S\} \vdash P \wedge Q$	[1,2,$\wedge$-I]

Justified Arguments

Answer 2 — Discussion

- **Discussion** There was no need need to write down an axiom deriving the premise $R \vee S$, because we only needed the premises P and Q in order to derive $P \wedge Q$
- It would not have been wrong to begin by deriving each of the three premises in turn, though, as in the following lines:

1.	$\{P, Q, R \vee S\} \vdash P$	[Axiom]
2.	$\{P, Q, R \vee S\} \vdash Q$	[Axiom]
3.	$\{P, Q, R \vee S\} \vdash R \vee S$	[Axiom]
4.	$\{P, Q, R \vee S\} \vdash P \wedge Q$	[1,2,$\wedge$-I]

- One possible strategy for constructing proofs is to begin by writing down an axiom for each premise, since this gives us a way of getting started: we can always remove any unnecessary lines later.
- Of course, this might involve revising the line numbers and references to line numbers. (there are packages in LaTeX that automate this)

Justified Arguments

Question 3

- Complete the following proof to justify $\{P \wedge Q\} \vdash Q \wedge P$

1.	$\{P \wedge Q\} \vdash P \wedge Q$	[Axiom]
2.	$\{P \wedge Q\} \vdash P$	[1, $\wedge$-E Left]
3.	$\{P \wedge Q\} \vdash Q$	[??]
4.	$\{P, Q\} \vdash Q \wedge P$	[??]

Justified Arguments

Answer 3

- Complete the following proof to justify $\{P \wedge Q\} \vdash Q \wedge P$

1.	$\{P \wedge Q\} \vdash P \wedge Q$	[Axiom]
2.	$\{P \wedge Q\} \vdash P$	[1, $\wedge$-E Left]
3.	$\{P \wedge Q\} \vdash Q$	[1, $\wedge$-E Right]
4.	$\{P, Q\} \vdash Q \wedge P$	[3,2,$\wedge$-I]

Justified Arguments

Inference Rules — Implication

- $\dfrac{\Gamma \cup \{\boldsymbol{A}\} \vdash \boldsymbol{B}}{\Gamma \vdash \boldsymbol{A} \rightarrow \boldsymbol{B}}$ ($\rightarrow$-introduction)
- The above should be read as: If there is a proof (justification, inference) for $\boldsymbol{B}$ under the set of premises, Γ, augmented with $\boldsymbol{A}$, then we have a proof (justification, inference) of $\boldsymbol{A} \rightarrow \boldsymbol{B}$, *under the unaugmented set of premises,* Γ.
  The unaugmented set of premises, Γ may have contained $\boldsymbol{A}$ already so we cannot assume

  $$(\Gamma \cup \{\boldsymbol{A}\}) - \{\boldsymbol{A}\} \text{ is equal to } \Gamma$$

- $\dfrac{\Gamma \vdash \boldsymbol{A} \quad \Gamma \vdash \boldsymbol{A} \rightarrow \boldsymbol{B}}{\Gamma \vdash \boldsymbol{B}}$ ($\rightarrow$-elimination)

Justified Arguments

Question 4

- Complete the following incomplete proof that the argument $\{P \wedge (P \to Q)\} \vdash Q$ is justified

1.	$\{P \wedge (P \to Q)\} \vdash P \wedge (P \to Q)$	[??]
2.	$\{P \wedge (P \to Q)\} \vdash P$	[1, $\wedge$-E Left]
3.	$\{P \wedge (P \to Q)\} \vdash P \to Q$	[1, ??]
4.	$\{P \wedge (P \to Q)\} \vdash Q$	[??]

Justified Arguments

Answer 4

- Complete the following incomplete proof that the argument $\{P \land (P \to Q)\} \vdash Q$ is justified

1.	$\{P \land (P \to Q)\} \vdash P \land (P \to Q)$	[Axiom]
2.	$\{P \land (P \to Q)\} \vdash P$	[1, $\land$-E left]
3.	$\{P \land (P \to Q)\} \vdash P \to Q$	[1, $\land$-E right]
4.	$\{P \land (P \to Q)\} \vdash Q$	[2,3, $\to$-E]

Justified Arguments

Question 5

- Complete the following incomplete proof that the argument $\{(P \wedge Q) \rightarrow R\} \vdash P \rightarrow (Q \rightarrow R)$ is justified

1.	$\{P, Q, (P \wedge Q) \rightarrow R\} \vdash P$	[Axiom]
2.	$\{P, Q, (P \wedge Q) \rightarrow R\} \vdash Q$	[??]
3.	$\{P, Q, (P \wedge Q) \rightarrow R\} \vdash (P \wedge Q) \rightarrow R$	[Axiom]
4.	$\{P, Q, (P \wedge Q) \rightarrow R\} \vdash P \wedge Q$	[??]
5.	$\{P, Q, (P \wedge Q) \rightarrow R\} \vdash R$	[4, 3, $\rightarrow$-E]
6.	$\{P, (P \wedge Q) \rightarrow R\} \vdash Q \rightarrow R$	[5, $\rightarrow$-I]
7.	$\{(P \wedge Q) \rightarrow R\} \vdash P \rightarrow (Q \rightarrow R)$	[6, ??]

Justified Arguments

Answer 5

- Complete the following incomplete proof that the argument $\{(P \wedge Q) \rightarrow R\} \vdash P \rightarrow (Q \rightarrow R)$ is justified

1.	$\{P, Q, (P \wedge Q) \rightarrow R\} \vdash P$	[Axiom]
2.	$\{P, Q, (P \wedge Q) \rightarrow R\} \vdash Q$	[Axiom]
3.	$\{P, Q, (P \wedge Q) \rightarrow R\} \vdash (P \wedge Q) \rightarrow R$	[Axiom]
4.	$\{P, Q, (P \wedge Q) \rightarrow R\} \vdash P \wedge Q$	[1, 2, $\wedge$-I]
5.	$\{P, Q, (P \wedge Q) \rightarrow R\} \vdash R$	[4, 3, $\rightarrow$-E]
6.	$\{P, (P \wedge Q) \rightarrow R\} \vdash Q \rightarrow R$	[5, $\rightarrow$-I]
7.	$\{(P \wedge Q) \rightarrow R\} \vdash P \rightarrow (Q \rightarrow R)$	[6, $\rightarrow$-I]

Justified Arguments

Inference Rules — Disjunction

- $\dfrac{\Gamma \vdash \boldsymbol{A}}{\Gamma \vdash \boldsymbol{A} \vee \boldsymbol{B}}$ ($\vee$-introduction left)
- $\dfrac{\Gamma \vdash \boldsymbol{B}}{\Gamma \vdash \boldsymbol{A} \vee \boldsymbol{B}}$ ($\vee$-introduction right)
- Disjunction elimination

$$\frac{\Gamma \vdash \boldsymbol{A} \vee \boldsymbol{B} \quad \Gamma \cup \{\boldsymbol{A}\} \vdash \boldsymbol{C} \quad \Gamma \cup \{\boldsymbol{B}\} \vdash \boldsymbol{C}}{\Gamma \vdash \boldsymbol{C}} \ (\vee\text{-elimination})$$

- The above should be read: if a set of premises Γ justifies the conclusion $\boldsymbol{A} \vee \boldsymbol{B}$ and Γ augmented with each of $\boldsymbol{A}$ or $\boldsymbol{B}$ separately justifies $\boldsymbol{C}$, then Γ justifies $\boldsymbol{C}$
- Disjunction elimination is a formal version of *proof by case analysis*

Inference Rules

Disjunction — Example 1

- Show that the argument $\{P\} \vdash P \vee Q$ is justified.
- **Answer**

1.	$\{P\} \vdash P$	[Axiom]
2.	$\{P\} \vdash P \vee Q$	[1, $\vee$-I left]

Justified Arguments

Question 6

- Show that the argument $\{Q\} \vdash P \vee Q$ is justified

Justified Arguments

Answer 6

1.	$\{Q\} \vdash Q$	[Axiom]
2.	$\{Q\} \vdash P \vee Q$	[1, $\vee$-I right]

Inference Rules

Disjunction — Example 2

- Show that the argument $\{P \vee Q\} \vdash Q \vee P$ is justified.
- **Answer**
  1. $\{P \vee Q\} \vdash P \vee Q$ [Axiom]
  2. $\{P \vee Q, P\} \vdash P$ [Axiom]
  3. $\{P \vee Q, P\} \vdash Q \vee P$ [2, $\vee$-I right]
  4. $\{P \vee Q, Q\} \vdash Q$ [Axiom]
  5. $\{P \vee Q, Q\} \vdash Q \vee P$ [4, $\vee$-I left]
  6. $\{P \vee Q\} \vdash Q \vee P$ [1, 3, 5, $\vee$-E]

Justified Arguments

Question 7

- Complete the following incomplete proof that the argument $\{Q \to R\} \vdash (P \lor Q) \to (P \lor R)$ is justified

1.	$\{Q \to R, P \lor Q, Q\} \vdash Q \to R$	[Axiom]
2.	$\{Q \to R, P \lor Q\} \vdash P \lor Q$	[??]
3.	$\{Q \to R, P \lor Q, P\} \vdash P$	[??]
4.	$\{Q \to R, P \lor Q, P\} \vdash P \lor R$	[?? $\lor$-I left]
5.	$\{Q \to R, P \lor Q, Q\} \vdash Q$	[Axiom]
6.	$\{Q \to R, P \lor Q, Q\} \vdash R$	[5, 1, $\to$-E]
7.	$\{Q \to R, P \lor Q, Q\} \vdash P \lor R$	[6, ??]
8.	$\{Q \to R, P \lor Q\} \vdash P \lor R$	[2, 4, 7, $\lor$-E]
9.	$\{Q \to R\} \vdash (P \lor Q) \to (P \lor R)$	[?? $\to$-I]

Justified Arguments

Answer 7

- Complete the following incomplete proof that the argument $\{Q \to R\} \vdash (P \lor Q) \to (P \lor R)$ is justified

1.	$\{Q \to R, P \lor Q, Q\} \vdash Q \to R$	[Axiom]
2.	$\{Q \to R, P \lor Q\} \vdash P \lor Q$	[Axiom]
3.	$\{Q \to R, P \lor Q, P\} \vdash P$	[Axiom]
4.	$\{Q \to R, P \lor Q, P\} \vdash P \lor R$	[3, $\lor$-I left]
5.	$\{Q \to R, P \lor Q, Q\} \vdash Q$	[Axiom]
6.	$\{Q \to R, P \lor Q, Q\} \vdash R$	[5, 1, $\to$-E]
7.	$\{Q \to R, P \lor Q, Q\} \vdash P \lor R$	[6, $\lor$-I right]
8.	$\{Q \to R, P \lor Q\} \vdash P \lor R$	[2, 4, 7, $\lor$-E]
9.	$\{Q \to R\} \vdash (P \lor Q) \to (P \lor R)$	[8, $\to$-I]

Proofs in Tree Form

- The syntax of proofs is recursive:
- A proof is either an axiom, or the result of applying a rule of inference to one, two or three proofs.
- We can therefore represent a proof by a tree diagram in which each node have one, two or three children
- For example, the proof of $\{P \wedge (P \to Q)\} \vdash Q$ in *Question 4* can be represented by the following diagram:

$$\dfrac{\dfrac{\{P \wedge (P \to Q)\} \vdash P \wedge (P \to Q)}{\{P \wedge (P \to Q)\} \vdash P}\ (\wedge\text{-E left}) \quad \dfrac{\{P \wedge (P \to Q)\} \vdash P \wedge (P \to Q)}{\{P \wedge (P \to Q)\} \vdash P \to Q}\ (\wedge\text{-E right})}{\{P \wedge (P \to Q)\} \vdash Q}\ (\to\text{-E})$$

Justified Arguments

Question 8

- Draw a diagram to represent the following proof:

1.	$\{P, R\} \vdash P$	[Axiom]
2.	$\{P, R\} \vdash R$	[Axiom]
2.	$\{P, R\} \vdash P \wedge R$	[1, 2, $\wedge$-I]

Justified Arguments

Answer 8

$$\dfrac{\{P, R\} \vdash P \quad \{P, R\} \vdash R}{\{P, R\} \vdash P \wedge R} \; (\wedge\text{-I})$$

Justified Arguments

Self-Assessment activity 7.4

- Is the following a justified argument ?
- $\{P \to R, Q \to R, P \vee Q\} \vdash R$
- First of all, prove
  - $\{P \to R, Q \to R, P \vee Q\} \vdash P \vee Q$
  - $\{P \to R, Q \to R, P \vee Q\} \cup \{P\} \vdash R$
  - $\{P \to R, Q \to R, P \vee Q\} \cup \{Q\} \vdash R$

Justified Arguments

Self-Assessment activity 7.4 — Tree layout

- Let $\Gamma = \{P \rightarrow R, Q \rightarrow R, P \vee Q\}$
- $$\frac{\Gamma \vdash P \vee Q \quad \Gamma \cup \{P\} \vdash R \quad \Gamma \cup \{Q\} \vdash R}{\Gamma \vdash R} \ (\vee\text{-elimination})$$
- $$\frac{\Gamma \cup \{P\} \vdash P \quad \Gamma \cup \{P\} \vdash P \rightarrow R}{\Gamma \cup \{P\} \vdash R} \ (\rightarrow\text{-elimination})$$
- $$\frac{\Gamma \cup \{Q\} \vdash Q \quad \Gamma \cup \{Q\} \vdash Q \rightarrow R}{\Gamma \cup \{Q\} \vdash R} \ (\rightarrow\text{-elimination})$$
- Complete tree layout
- $$\frac{\Gamma \vdash P \vee Q \quad \dfrac{\begin{array}{c}\Gamma \cup \{P\} \\ \vdash P\end{array} \quad \begin{array}{c}\Gamma \cup \{P\} \\ \vdash P \rightarrow R\end{array}}{\Gamma \cup \{P\} \vdash R}\ (\rightarrow\text{-E}) \quad \dfrac{\begin{array}{c}\Gamma \cup \{Q\} \\ \vdash Q\end{array} \quad \begin{array}{c}\Gamma \cup \{Q\} \\ \vdash Q \rightarrow R\end{array}}{\Gamma \cup \{Q\} \vdash R}\ (\rightarrow\text{-E})}{\Gamma \vdash R} \ (\vee\text{-E})$$

Justified Arguments

Self-assessment activity 7.4 — Linear Layout

1.	$\{P \to R, Q \to R, P \vee Q\} \vdash P \vee Q$	[Axiom]
2.	$\{P \to R, Q \to R, P \vee Q\} \cup \{P\} \vdash P$	[Axiom]
3.	$\{P \to R, Q \to R, P \vee Q\} \cup \{P\} \vdash P \to R$	[Axiom]
4.	$\{P \to R, Q \to R, P \vee Q\} \cup \{Q\} \vdash Q$	[Axiom]
5.	$\{P \to R, Q \to R, P \vee Q\} \cup \{Q\} \vdash Q \to R$	[Axiom]
6.	$\{P \to R, Q \to R, P \vee Q\} \cup \{P\} \vdash R$	[2, 3, $\to$-E]
7.	$\{P \to R, Q \to R, P \vee Q\} \cup \{Q\} \vdash R$	[4, 5, $\to$-E]
8.	$\{P \to R, Q \to R, P \vee Q\} \vdash R$	[1, 6, 7, $\vee$-E]

Logic Puzzles — Introduction

- The following puzzles are usually given as exercises in verbal reasoning — however you can use your knowledge of propositional logic to *calculate* the answers.
- The answers below (in the notes version) give references to the sources of the puzzles and solutions.

Calculating with Logic

Knights and Knaves

- There is a wide variety of puzzles about an island in which certain inhabitants called *knights* always tell the truth, and others called *knaves* always lie.
- It is assumed that every inhabitant of the island is either a knight or a knave.
- The following puzzles can be solved by verbal reasoning or by using truth tables

Calculating with Logic

Knights and Knaves

1. Three inhabitants of this island — A, B and C — are standing together in a garden. You pass by and ask A *Are you a knight or a knave ?* A answers but rather indistinctly so you cannot hear. You then ask B *What did A say ?* B replies *A said that he is a knave* At this point C says *Don't believe B; he is lying*
   What are B and C ?
2. Suppose instead of asking A what he is, you asked A *How many knights are among you ?* Again you cannot hear A's reply. So you ask B *What did A say ?* B replies *A said there is only one knight among us* Then C says *Don't believe B; he is lying*
   Now what are B and C ?

Calculating with Logic

Knights and Knaves

3. In this problem there are only two people A and B each of whom is either a knight or knave. A makes the following statement *At least one of us is a knave*
   What are A and B ?
4. Suppose A says *Either I am a knave or B is a knight*
   What are A and B ?
5. Suppose A says *Either I am a knave or else 2 + 2 = 5*
   What would you conclude ?
6. Again we have 3 people A B C each either a knave or a knight. A and B say the following:
   A: All of us are knaves
   B: Exactly one of us is a knight
   What are A B C ?

Calculating with Logic

Knights and Knaves

7. Again three inhabitants A B C each of whom is either a knight or knave. Two people are said to be of the same type if they are both knights or both knaves. A and B make the following statements:
   A: B is a knave
   B: A and C are of the same type
   What is C ?
8. Again three people A B C. A says *B and C are of the same type* Someone then asks C *Are A and B of the same type ?*
   What does C answer ?

Calculating with Logic

Knights and Knaves — Variant

- A variation on the above type of problems deals with three types of people: knights and knaves as before and normal people who sometimes lie and sometimes tell the truth.

9. We are given three people A B C one of whom is a knight, one a knave and one normal (but maybe not in that order). They make the following statements:
   A: I am normal
   B: That is true
   C: Exactly one of us is a knave
   What are A B C ?

Calculating with Logic

Knights and Knaves — Variant

10. Two people A and B each of whom is either a knight, or knave or normal make the following statements:
    A: B is a knight
    B: A is not a knight
    Prove that at least one of them is telling the truth but is not a knight.
11. This time A and B say the following:
    A: B is a knight
    B: A is a knave
    Prove that either one of them is telling the truth but is not a knight or one of them is lying but is not a knave.

Calculating with Logic

Harder Logic Puzzles

- Here are several logic puzzles which involve liars, truth-tellers and those who speak the truth or lie at random.
- The later puzzles are actually extensions of the first (so if you have really solved the first, the rest might be easier).

Calculating with Logic

Harder Logic Puzzles

1.

- A tourist is enjoying an afternoon refreshment in a local pub in England when the bartender says to him: "Do you see those three men over there ? One is Mr. X, who always tells the truth, another is Mr. Y, who always lies, and the third is Mr. Z, who sometimes tells the truth and sometimes lies (that is, Mr. Z answers *yes* or *no* at random without regard for the question). You may ask them three yes/no questions, always indicating which man should answer. If, after asking these three questions, you correctly identify Mr. X, Mr. Y, and Mr. Z, they will buy you a drink."
- What yes/no questions should the thirsty tourist ask ?
- Hint: Use the first question to find some person of the three who is not Mr. Z. Ask him the other two questions.

Calculating with Logic

Harder Logic Puzzles

2.

- In a certain country, there are three kinds of people: workers (who always tell the truth), capitalists (who never tell the truth), and students (who sometimes tell the truth and sometimes lie).
- At a fork in the road, one branch leads to the capital. A worker, a capitalist, and a student are standing at the side of the road but are not identifiable in any obvious way.
- By asking two yes or no questions, find out which fork leads to the capital. (Each question may be addressed to any of the three.)

Calculating with Logic

Harder Logic Puzzles

3.

- Three gods A, B, and C are called, in some order, True, False, and Random. True always speaks truly, False always speaks falsely, but whether Random speaks truly or falsely is a completely *random* matter.
- Your task is to determine the identities of A, B, and C by asking three questions; each question must be put to exactly one god.
- The gods understand English, but will answer all questions in their own language, in which the words for “yes” and “no” are “da” and “ja”, in some order. *You do not know which word means which*.

Calculating with Logic

Knights and Knaves — Truth Table Answers

- We will use the following notation
  - A stands for *A is a Knight*
  - **not** A stands for *A is a Knave*
  - SA stands for *The statement by A is **True***
  - **not** SA stands for *The statement by A is **False***
- Hence, in this world of truth tellers and liars we know:
  - (A ⇒ SA) **and** (**not** A ⇒ **not** SA)
- The above is equivalent to:
  - A ⟺ SA
- This gives us a way of solving the puzzles using truth tables

Question 1

Problem 26 in Smullyan (1981)

- We have the following from the statements of B and C:
  - $B \iff (A \iff \textbf{not}\, A)$
  - $C \iff \textbf{not}\, B$
- We now construct a truth table for the conjunction of the two propositions and see which entries are **True**.

B	C	$B \iff (A \iff$	$\textbf{not}\ A)$	**and**	$C \iff \textbf{not}\ B$
True	**True**	**False**	**False**	**False**	**False**
True	**False**	**False**	**False**	**False**	**True**
False	**True**	**True**	**False**	**True**	**True**
False	**False**	**True**	**False**	**False**	**False**

- The **True** tells us that B is a knave and C is a knight — it is the only entry in the truth table for the proposition which is **True**.

Question 2

Problem 27 in Smullyan (1981)

- The answer is the same as that of the preceding problem, though the reasoning is a bit different.
- We have the following from the statements of B and C:
  - B $\Longleftrightarrow$ (A $\Longleftrightarrow$ 1 knight)
  - C $\Longleftrightarrow$ **not** B

A	B	C	B $\Longleftrightarrow$ (A $\Longleftrightarrow$ 1 knight)		**and**	C $\Longleftrightarrow$ **not** B
True	**True**	**True**	**False**	**False**	**False**	**False**
True	**True**	**False**	**False**	**False**	**False**	**True**
True	**False**	**True**	**True**	**False**	**True**	**True**
True	**False**	**False**	**False**	**True**	**False**	**False**
False	**True**	**True**	**True**	**True**	**False**	**False**
False	**True**	**False**	**False**	**False**	**False**	**True**
False	**False**	**True**	**True**	**False**	**True**	**True**
False	**False**	**False**	**False**	**True**	**False**	**False**

Question 3

Problem 28 in Smullyan (1981)

- We have the following from the statement of A:
  - $A \iff 1$ or more knaves

A	B	$A \iff$	1 or more knaves
True	**True**	**False**	**False**
True	**False**	**True**	**True**
False	**True**	**False**	**True**
False	**False**	**False**	**True**

- So A is a knight and B is a knave.

Question 4

Problem 29 in Smullyan (1981)

- We have the following from the statement of A:
  - $A \iff (\textbf{not}\ A\ \textbf{or}\ B)$

A	B	$A \iff$	(**not** A **or** B)
True	**True**	**True**	**True**
True	**False**	**False**	**False**
False	**True**	**False**	**True**
False	**False**	**False**	**True**

- So we have A and B are both knights.

Question 5

Problem 30 in Smullyan (1981)

- We have the following from the statement of A:
  - $A \iff (\textbf{not}\, A\, \textbf{or}\, 2 + 2 = 5)$

A	$\iff$	(**not** A **or** $2 + 2 = 5$)
True	**False**	**False**
False	**False**	**True**

- So here there is no solution for any possible assignment of truth values — we call this a contradiction.

Question 6

Problem 31 in Smullyan (1981)

- We have the following statements:
  - SA: All Knaves
  - SB: Exactly 1 knave

A	B	C	(A $\iff$ SA)	and	(B $\iff$ SB)
True	**True**	**True**	**False**	**False**	**False**
True	**True**	**False**	**False**	**False**	**True**
True	**False**	**True**	**False**	**False**	**False**
True	**False**	**False**	**False**	**False**	**True**
False	**True**	**True**	**True**	**True**	**True**
False	**True**	**False**	**True**	**False**	**False**
False	**False**	**True**	**True**	**False**	**False**
False	**False**	**False**	**False**	**False**	**True**

- So we have A knave and B and C knights.

Question 7

Problem 34 in Smullyan (1981)

- We have the following statements:
  - SA: **not** B
  - SB: A & C are the same

A	B	C	(A $\iff$ SA)	**and**	(B $\iff$ SB)
True	**True**	**True**	**False**	**False**	**True**
True	**True**	**False**	**False**	**False**	**False**
True	**False**	**True**	**True**	**False**	**False**
True	**False**	**False**	**True**	**True**	**True**
False	**True**	**True**	**False**	**False**	**False**
False	**True**	**False**	**True**	**True**	**True**
False	**False**	**True**	**False**	**False**	**True**
False	**False**	**False**	**True**	**False**	**False**

- So C must be a knave.

Question 8

Problem 35 in Smullyan (1981)

- We have the following statement:
  - SA: B & C are same
- We now construct a truth table including the response of C to the question *Is it **True** that A and B are the same ?*

A	B	C	(A $\Longleftrightarrow$ SA)	**and**	(C $\Longleftrightarrow$ A = B)	
True	**True**	**True**	**True**	**True**	**True**	yes
True	**True**	**False**	**False**	**False**	**False**	no
True	**False**	**True**	**False**	**False**	**False**	no
True	**False**	**False**	**True**	**True**	**True**	yes
False	**True**	**True**	**False**	**False**	**False**	no
False	**True**	**False**	**True**	**True**	**True**	yes
False	**False**	**True**	**True**	**True**	**True**	yes
False	**False**	**False**	**False**	**False**	**False**	no

- Thus in both cases C answers *yes*

Harder Logic Puzzles
Solutions

- The approach to finding a solution is based on the answer to exercise 1.22 in Backhouse (1986)
- The source of the problems is as follows:
  - Q 1 is Problem 2-7(b) in Manna (1974)
  - Q 2 is exercise 1.46(b) in Mendelson (2009)
  - Q 3 is from chapter 29 of Boolos (1998)
    This problem was originally in an article by George Boolos in *The Harvard Review of Philosophy* 6 (1996): 62–65

Harder Logic Puzzles

Q 1 (Problem 2-7(b) in Manna (1974))

- A tourist is enjoying an afternoon refreshment in a local pub in England when the bartender says to him: “Do you see those three men over there ? One is Mr. X, who always tells the truth, another is Mr. Y, who always lies, and the third is Mr. Z, who sometimes tells the truth and sometimes lies (that is, Mr. Z answers *yes* or *no* at random without regard for the question). You may ask them three yes/no questions, always indicating which man should answer. If, after asking these three questions, you correctly identify Mr. X, Mr. Y, and Mr. Z, they will buy you a drink.”
- What yes/no questions should the thirsty tourist ask ?
- Hint: Use the first question to find some person of the three who is not Mr. Z. Ask him the other two questions.

Harder Logic Puzzles

Q 1 Solution (a)

- We can label the people (say by distance from us) as A, B and C.
- With no prior knowledge we may as well ask the first question to A.
- A could be a knight, a knave or a normal (that's what we call people who lie or tell the truth at random).
- The hint tells us that if should use the first question to identify someone who is not normal.
- Once we have done that the rest is easy: ask a knight or a knave if $2 + 2 = 5$ and you immediately know what they are and can then use them to tell you who the rest are with one question.

Harder Logic Puzzles

Q 1 Solution (b)

- The *Eureka* step is to realise that you can *calculate* the first question by working out what properties it must have and then rearranging a description of the properties as propositions into the form:
- $Q \iff$ some proposition not involving Q
- where Q stands for a question of the form “Is it **True** that . . . ” where the question is trying to identify whether B is normal or not.

Harder Logic Puzzles

Q 1 Solution (c)

- In this case our first question (to A) should satisfy the following:
  1. If A is a knight and A says Q is **True** then B is normal.
  2. If A is a knave and A says Q is **True** then B is normal.
  3. If A is a knight and A says Q is **False** then B is **not** normal.
  4. If A is a knave and A says Q is **False** then B is **not** normal.

Harder Logic Puzzles

Q 1 Solution (d)

- We can represent the above statements as a compound proposition.
- We use “BN” to represent “B is normal”; “Q” stands for “Q is **True**”.
- Remember that is a knave says “Q is **True**” that “**not** Q” is really the case (and vice-versa).

(A **and** Q) $\Rightarrow$ BN

and

(**not** A **and not** Q) $\Rightarrow$ BN

and

(A **and not** Q) $\Rightarrow$ **not** BN

and

(**not** A **and** Q) $\Rightarrow$ **not** BN

Harder Logic Puzzles

Q 1 Solution (e)

- We now use the following identity (use a truth table to prove the identity):
- $(p\ \mathbf{and}\ q) \Rightarrow r \equiv q \Rightarrow (p \Rightarrow r)$
- This gives us:

$\mathrm{Q} \Rightarrow (\mathrm{A} \Rightarrow \mathrm{BN})$

and

$\mathbf{not}\,\mathrm{Q} \Rightarrow (\mathbf{not}\,\mathrm{A} \Rightarrow \mathrm{BN})$

and

$\mathbf{not}\,\mathrm{Q} \Rightarrow (\mathrm{A} \Rightarrow \mathbf{not}\,\mathrm{BN})$

and

$\mathrm{Q} \Rightarrow (\mathbf{not}\,\mathrm{A} \Rightarrow \mathbf{not}\,\mathrm{BN})$

Harder Logic Puzzles

Q 1 Solution (f)

- We now use the following identity (again prove that this is an identity):
- $(p \Rightarrow q)$ **and** $(p \Rightarrow r) \equiv p \Rightarrow (q$ **and** $r)$
- This gives us:

$$\mathrm{Q} \Rightarrow ((\mathrm{A} \Rightarrow \mathrm{BN}) \textbf{ and } (\textbf{not } \mathrm{A} \Rightarrow \textbf{not } \mathrm{BN}))$$

and

$$\textbf{not } \mathrm{Q} \Rightarrow ((\textbf{not } \mathrm{A} \Rightarrow \mathrm{BN}) \textbf{ and } (\mathrm{A} \Rightarrow \textbf{not } \mathrm{BN}))$$

Harder Logic Puzzles

Q 1 Solution (g)

- We now use the following identity (again prove that this is an identity):
- $(\textbf{not}\ p \Rightarrow \textbf{not}\ q) \equiv p \Rightarrow q$
- This gives us:

$$Q \Rightarrow ((A \Rightarrow BN)\ \textbf{and}\ (BN \Rightarrow A))$$

and

$$\textbf{not}\ Q \Rightarrow ((\textbf{not}\ A \Rightarrow BN)\ \textbf{and}\ (BN \Rightarrow \textbf{not}\ A))$$

- Use the definition of $\iff$

$$Q \Rightarrow (A \iff BN)$$

and

$$\textbf{not}\ Q \Rightarrow (\textbf{not}\ A \iff BN)$$

Harder Logic Puzzles

Q 1 Solution (h)

- We finally use the definition of $\iff$ and the identity:
- $\textbf{not}\, p \iff q \equiv \textbf{not}(p \iff q)$
- This gives us:

  $$\mathrm{Q} \iff (\mathrm{A} \iff \mathrm{BN})$$

- So in English our first question (to A) would be:
  - *Is it true that the statement that you are a truth teller is equivalent to the statement that B is normal ?*
- This gives a general approach to similar puzzles

Logic

Programming Language Theory & Proof Theory

- Curry-Howard isomorphism is the direct relationship between computer programs and mathematical proofs
- A proof is a program
- The formula it proves is the type for the program
- A logic corresponds to a programming language
- For example, at the level of formulas and types:
- Implication ↔ function type
- Conjunction (AND) ↔ product type
- Disjunction (OR) ↔ sum type
- Haskell/The Curry-Howard isomorphism — article on CH and the functional programming language Haskell
- Curry-Howard isomorphism — overview article

Future Work

Topics & Events

- Wednesday 28 April 2021 iCMA46 due
- Sunday, 2 May 2021 online tutorial Unit 7 Computability, Complexity
- Sunday, 16 May 2021 online tutorial exam revision
- Saturday, 22 May 2021 online tutorial exam revision
- Tuesday 25 May 2021 iCMA47 due
- Tuesday 8 June 2021 Exam
- Please email me with any requests for particular topics