Discrete Mathematics I

Computer Science Tripos, Part 1A
Paper 1

Natural Sciences Tripos, Part 1A,
Computer Science option

Politics, Psychology and Sociology, Part 1,
Introduction to Computer Science option

2012–13

Lecturer: Sam Staton

Computer Laboratory

University of Cambridge

`Time-stamp: November 14, 2012, 17:06`

Contents

Syllabus

Lecturer: Dr S. Staton

No. of lectures: 9

This course is a prerequisite for all theory courses as well as Probability, Discrete Mathematics II, Algorithms I, Security (Part IB and Part II), Artificial Intelligence (Part IB and Part II), Information Theory and Coding (Part II).

Aims

This course will develop the intuition for discrete mathematics reasoning involving numbers and sets.

Lectures

- **Logic.** Propositional and predicate logic and their relationship to informal reasoning, truth tables, validity.
- **Proof.** Proving propositional and predicate formulas in a structured way. Introduction and elimination rules.
- **Sets.** Basic set theory. Relations, graphs and orders.
- **Induction.** Proof by induction, including proofs about total functional programs over natural numbers and lists.

Objectives

On completing the course, students should be able to

- write a clear statement of a problem as a theorem in mathematical notation;
- prove and disprove assertions using a variety of techniques.

Recommended reading

Biggs, N.L. (1989). *Discrete mathematics.* Oxford University Press.
Bornat, R. (2005). *Proof and Disproof in Formal Logic.* Oxford University Press.
Cullinane, M.J. (2012). A transition to mathematics with proofs. Jones & Bartlett.
Devlin, K. (2003). *Sets, functions, and logic: an introduction to abstract mathematics.* Chapman and Hall/CRC Mathematics (3rd ed.).
Mattson, H.F. Jr (1993). *Discrete mathematics.* Wiley.
Nissanke, N. (1999). *Introductory logic and sets for computer scientists.* Addison-Wesley.
Pólya, G. (1980). *How to solve it.* Penguin.
(*) Rosen, K.H. (1999). *Discrete mathematics and its applications* (6th ed.). McGraw-Hill.
(*) Velleman, D. J. (1994). *How to prove it (a structured approach).* CUP.

For Supervisors (and Students too)

The main aim of the course is to enable students to confidently use the language of propositional and predicate logic, and set theory.

We first introduce the language of propositional logic, discussing the relationship to natural-language argument. We define the meaning of formulae with the truth semantics w.r.t. assumptions on the atomic propositions, and, equivalently, with truth tables. We also introduce equational reasoning, to make instantiation and reasoning-in-context explicit.

We then introduce quantifiers, again emphasising the intuitive reading of formulae and defining the truth semantics. We introduce the notions of free and bound variable (but not alpha equivalence).

We do not develop any metatheory, and we treat propositional assumptions, valuations of variables, and models of atomic predicate symbols all rather informally. There are no turnstiles, but we talk about valid formulae and (briefly) about satisfiable formulae.

We then introduce 'structured' proof. This is essentially natural deduction proof, laid out on the page in box-and-line style. The rationale here is to introduce a style of proof for which one can easily define what is (or is not) a legal proof, but where the proof text on the page is reasonably close to the normal mathematical 'informal but rigorous' practice that will be used in most of the rest of the Tripos. We emphasise how to prove and how to use each connective, and talk about the pragmatics of finding and writing proofs.

The set theory material introduces the basic notions of set, element, union, intersection, powerset, and product, relating to predicates (e.g. relating predicates and set comprehensions, and the properties of union to those of disjunction), with some more small example proofs. We then define some of the standard properties of relations (reflexive, symmetric, transitive, antisymmetric, acyclic, total) to characterise directed graphs, undirected graphs, equivalence relations, pre-orders, partial orders, and functions). These are illustrated with simple examples to introduce the concepts, but their properties and uses are not explored in any depth (for example, we do not define what it means to be an injection or surjection).

Finally, we recall inductive proof over the naturals, making the induction principle explicit in predicate logic, and over lists, talking about inductive proof of simple pure functional programs (taking examples from the previous SWEng II notes).

I'd suggest 3 supervisons. A possible schedule might be:

1. After the first 2–3 lectures
   Example Sheets 1 and 2, covering Propositional and Predicate Logic
2. After the next 3–4 lectures
   Example Sheets 3 and the first part of 4, covering Structured Proof and Sets
3. After all 9 lectures
   Example Sheet 4 (the remainder) and 5, covering Inductive Proof

These notes are based on notes written by Peter Sewell.

Learning Guide

Notes: These notes include all the slides, but by no means everything that'll be said in lectures.

Exercises: There are some exercises at the end of the notes. I suggest you do all of them. Most should be rather straightforward; they're aimed at strengthening your intuition about the concepts and helping you develop quick (but precise) manipulation skills, not to provide deep intellectual challenges. A few may need a bit more thought. Some are taken (or adapted) from Devlin, Rosen, or Velleman. More exercises and examples can be found in any of those.

Tripos questions: This version of the course was new in 2008.

Feedback: Please do complete the on-line feedback form at the end of the course, and let me know during it if you discover errors in the notes or if the pace is too fast or slow.

Errata: A list of any corrections to the notes will be on the course web page.

1 Introduction

Discrete Mathematics I

Computer Science Tripos, Part 1A

Natural Sciences Tripos, Part 1A, Computer Science

Politics, Psychology and Sociology Part 1, Introduction to Computer Science

Sam Staton

1A, 9 lectures

2011 – 2013

slide 1

Introduction

At the start of the Industrial Revolution, we built bridges and steam engines without enough applied maths, physics, materials science, etc.

Fig. 5. An Illustration of What Explosion Did to Stays and Braces

Fix: understanding based on continuous-mathematics models — calculus, matrices, complex analysis,...

slide 2

Introduction

Now, we build computer systems, and sometimes, sadly, ...

[Ariane 501]

But, computer systems are large and complex, and are largely *discrete*: we can't use approximate continuous models for correctness reasoning. So, need *applied discrete maths* — logic, set theory, graph theory, combinatorics, abstract algebra, ...

slide 3

Logic and Set Theory — Pure Mathematics

Origins with the Greeks, 500–350 BC, philosophy and geometry:

Aristotle, Euclid

Formal logic in the 1800s:

De Morgan, Boole, Venn, Peirce, Frege

Set theory, model theory, proof theory; late 1800s onwards:

Cantor, Russell, Hilbert, Zermelo, Frankel, Goedel, Gentzen, Tarski, Kripke, Martin-Lof, Girard

Focus then on the foundations of mathematics — but what was developed then turns out to be unreasonably effective in Computer Science.

slide 4

Logic and Set Theory — Applications in Computer Science

- modelling digital circuits (IA Digital Electronics, IB ECAD)
- proofs about particular algorithms (IA/IB Algorithms)
- proofs about what is (or is not!) computable and with what complexity (IB Computation Theory, Complexity Theory)
- foundations and proofs for programming languages (IA Regular Languages and Finite Automata, IB Prolog, IB/II Semantics of Programming Languages, II Types, II Topics in Concurrency)
- proofs about security and cryptography (IB/II Security)
- foundation of databases (IB Databases)
- automated reasoning and model-checking tools (IB Logic & Proof, II Hoare Logic, Temporal Logic and Model Checking)

slide 5

Outline

- Propositional Logic
- Predicate Logic
- Sets
- Inductive Proof

Focus on *using* this material, rather than on metatheoretic study.

More (and more metatheory) in Discrete Maths 2 and in Logic & Proof.

slide 6

Supervisons

Needs practice to become fluent.

Five example sheets. Many more suitable exercises in the books.

Up to your DoS and supervisor, but I'd suggest 3 supervisons. A possible schedule might be:

1. After the first 2–3 lectures
   Example Sheets 1 and 2, covering Propositional and Predicate Logic
2. After the next 3–4 lectures
   Example Sheets 3 and the first part of 4, covering Structured Proof and Sets
3. After all 9 lectures
   Example Sheet 4 (the remainder) and 5, covering Inductive Proof

slide 7

2 Propositional Logic

Propositional Logic

slide 8

In this section we cover propositional logic. We give a meaning to propositions using truth tables, and we consider equational reasoning on propositional logic. We also consider properties of propositions such as validity, tautology, and satisfiablity.

Students taking 50% Computer Science will have seen Boolean algebra in earlier courses, such as Digital Electronics. You should take note that mathematical logic is different in spirit from logic for electronics. For instance, xor and nand are not very important in mathematical logic, whereas implication is not so useful in electronics.

Propositional Logic

Starting point is informal natural-language argument:

Socrates is a man. All men are mortal. So Socrates is mortal.

If a person runs barefoot, then his feet hurt. Socrates' feet hurt. Therefore, Socrates ran barefoot

slide 9

It will either rain or snow tomorrow. It's too warm for snow. Therefore, it will rain.

Either the butler is guilty or the maid is guilty. Either the maid is guilty or the cook is guilty. Therefore, either the butler is guilty or the cook is guilty.

slide 10

It will either rain or snow tomorrow. It's too warm for snow. Therefore, it will rain.

Either the framger widget is misfiring or the wrompal mechanism is out of alignment. I've checked the alignment of the wrompal mechanism, and it's fine. Therefore, the framger widget is misfiring.

slide 11

Either the framger widget is misfiring or the wrompal mechanism is out of alignment. I've checked the alignment of the wrompal mechanism, and it's fine. Therefore, the framger widget is misfiring.

Either p *or* q. *Not* q. *Therefore,* p

slide 12

2.1 The Language of Propositional Logic

Atomic Propositions

$1 + 1 = 2$

$10 + 10 = 30$

Tom is a student

Is Tom a student? $\times$

Give Tom food! $\times$

$x + 7 = 10$ $\times$

$1 + 2 + ... + n = n(n + 1)/2$ $\times$

slide 13

Atomic Propositions

We'll use lowercase letters p, q, for *atomic propositions*.

slide 14

When you use logic to reason about particular things, you will want to have meaningful atomic propositions, like "Tom is a student" or "It is raining". For studying logic in general we use symbols like p and q.

Some people say "propositional variable" instead of "atomic proposition".

We do not fix atomic propositions to be true or false. Rather, we investigate how their truth and falsity affects the compound propositions that we build. Atomic propositions are *atomic* because, for the purposes of logic, they are indivisible and their truth does not depend on the truth of other things.

Building Propositions: Truth and Falsity

We'll write T for the constant true proposition, and F for the constant false proposition.

slide 15

Compound Propositions

We'll build more complex *compound propositions* out of the atomic propositions (p, q) and T and F.

We'll use capital letters (P, Q, etc.) to stand for arbitrary propositions. They might stand for atomic propositions or compound propositions.

slide 16

Building Compound Propositions: Conjunction

If P and Q are two propositions, $P \wedge Q$ is a proposition.

Pronounce $P \wedge Q$ as 'P and Q'. Sometimes written with $\&$ or .

Definition: $P \wedge Q$ is true if (and only if) P is true and Q is true

Examples:

Tom is a student $\wedge$ Tom has red hair

$(1 + 1 = 2) \wedge (7 \leq 10)$

$(1 + 1 = 2) \wedge (2 = 3)$

$((1 + 1 = 2) \wedge (7 \leq 10)) \wedge (5 \leq 5)$

$(\mathrm{p} \wedge \mathrm{q}) \wedge \mathrm{p}$

slide 17

Building Compound Propositions: Conjunction

We defined the meaning of $P \wedge Q$ by saying '$P \wedge Q$ is true if and only if P is true and Q is true'.

We could instead, equivalently, have defined it by enumerating all the cases, in a *truth table*:

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

According to this definition, is $((1+1=2) \wedge (7 \leq 10)) \wedge (5 \leq 5)$ true or false?

slide 18

Building Compound Propositions: Conjunction

We pronounce $P \wedge Q$ as 'P and Q', but not all uses of the English 'and' can be faithfully translated into $\wedge$.

Tom and Alice had a dance.

Grouping

Tom went to a lecture and had lunch.

Temporal ordering?

The Federal Reserve relaxed banking regulations, and the markets boomed.

Causality?

When we want to talk about time or causality in CS, we'll do so explicitly; they are not built into this logic.

slide 19

Building Compound Propositions: Conjunction

Basic properties:

The order doesn't matter: whatever P and Q are, $P \wedge Q$ means the same thing as $Q \wedge P$.

Check, according to the truth table definition, considering each of the 4 possible cases:

P	Q	$P \wedge Q$	$Q \wedge P$
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

In other words, $\wedge$ is *commutative*

slide 20

Building Compound Propositions: Conjunction

...and:

The grouping doesn't matter: whatever P, Q, and R are, $P \wedge (Q \wedge R)$ means the same thing as $(P \wedge Q) \wedge R$.

(Check, according to the truth table definition, considering each of the 8 possible cases).

In other words, $\wedge$ is *associative*

So we'll happily omit *some* parentheses, e.g. writing $P_1 \wedge P_2 \wedge P_3 \wedge P_4$ for $P_1 \wedge (P_2 \wedge (P_3 \wedge P_4))$.

slide 21

Building Compound Propositions: Disjunction

If P and Q are two propositions, $P \vee Q$ is a proposition.

Pronounce $P \vee Q$ as 'P or Q'. Sometimes written with $|$ or $+$

Definition: $P \vee Q$ is true if and only if P is true or Q is true

Equivalent truth-table definition:

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

slide 22

Building Compound Propositions: Disjunction

You can see from that truth table that $\vee$ is an *inclusive* or: $P \vee Q$ if *at least one* of P and Q.

$(2+2=4) \vee (3+3=6)$ is true

$(2+2=4) \vee (3+3=7)$ is true

The English 'or' is sometimes an *exclusive* or: P xor Q if *exactly* one of P and Q. 'Fluffy is either a rabbit or a cat.'

P	Q	$P \vee Q$	P xor Q
T	T	T	F
T	F	T	T
F	T	T	T
F	F	F	F

slide 23

Although xor is important in electronics, it does not play a primitive role in logic. If you feel that an English sentence 'P or Q' reads as (P xor Q), you should regard it more precisely as 'either P or Q but not both', which can be formalized using negation as $(P \vee Q) \wedge \neg(P \wedge Q)$.

Building Compound Propositions: Disjunction

Basic Properties

$\vee$ is also commutative and associative:

$P \vee Q$ and $Q \vee P$ have the same meaning

$P \vee (Q \vee R)$ and $(P \vee Q) \vee R$ have the same meaning

$\wedge$ distributes over $\vee$:

$P \wedge (Q \vee R)$ and $(P \wedge Q) \vee (P \wedge R)$ have the same meaning

'P and either Q or R' 'either (P and Q) or (P and R)'

and the other way round: $\vee$ distributes over $\wedge$

$P \vee (Q \wedge R)$ and $(P \vee Q) \wedge (P \vee R)$ have the same meaning

When we mix $\wedge$ and $\vee$, we take care with the parentheses!

slide 24

Building Compound Propositions: Negation

If P is some proposition, $\neg P$ is a proposition.

Pronounce $\neg P$ as 'not P'. Sometimes written as $\sim P$ or $\overline{P}$

Definition: $\neg P$ is true if and only if P is false

Equivalent truth-table definition:

P	$\neg P$
T	F
F	T

slide 25

Building Compound Propositions: Implication

If P and Q are two propositions, $P \Rightarrow Q$ is a proposition.

Pronounce $P \Rightarrow Q$ as 'P implies Q'. Sometimes written with $\rightarrow$

Definition: $P \Rightarrow Q$ is true if (and only if), whenever P is true, Q is true

Equivalent truth-table definition:

P	Q	$P \Rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

slide 26

Building Compound Propositions: Implication

That can be confusing. First, the logic is not talking about causation, but just about truth values.

$(1 + 1 = 2) \Rightarrow (3 < 4)$ is true

Second, $P \Rightarrow Q$ is vacuously true if P is false.

'If I'm a giant squid, then I live in the ocean'

For that to be true, either:

(a) I really am a giant squid, in which case I must live in the ocean, or

(b) I'm not a giant squid, in which case we don't care where I live.

$P \Rightarrow Q$ and $(P \wedge Q) \vee \neg P$ and $Q \vee \neg P$ all have the same meaning

slide 27

Building Compound Propositions: Implication

Basic properties:

$P \Rightarrow Q$ and $\neg Q \Rightarrow \neg P$ have the same meaning

$\Rightarrow$ is not commutative: $P \Rightarrow Q$ and $Q \Rightarrow P$ do not have the same meaning

$P \Rightarrow (Q \wedge R)$ and $(P \Rightarrow Q) \wedge (P \Rightarrow R)$ have the same meaning

$(P \wedge Q) \Rightarrow R$ and $(P \Rightarrow R) \wedge (Q \Rightarrow R)$ do not

$(P \wedge Q) \Rightarrow R$ and $P \Rightarrow (Q \Rightarrow R)$ do

slide 28

Building Compound Propositions: Bi-Implication

If P and Q are two propositions, $P \Leftrightarrow Q$ is a proposition.

Pronounce $P \Leftrightarrow Q$ as 'P if and only if Q'. Sometimes written with $P \leftrightarrow Q$ or $P = Q$.

Definition: $P \Leftrightarrow Q$ is true if (and only if) P is true whenever Q is true, and vice versa

Equivalent truth-table definition:

P	Q	$P \Leftrightarrow Q$
T	T	T
T	F	F
F	T	F
F	F	T

slide 29

The Language of Propositional Logic

Summarising, the propositions of propositional logic are the terms of the grammar

$$P, Q ::= \mathrm{p} \mid \mathrm{q} \mid \ldots \mid T \mid F \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \Rightarrow Q \mid P \Leftrightarrow Q$$

We use parentheses (P) as necessary to avoid ambiguity.

For any such proposition P, once the truth value of each atomic proposition p it mentions is fixed (true or false), we've defined whether P is true or false.

slide 30

Example Compound Truth Table

Given an arbitrary proposition P, we can calculate the meaning of P for all possible assumptions on its atomic propositions by enumerating the cases in a truth table.

For example, consider $P \stackrel{\text{def}}{=} ((\mathrm{p} \vee \neg\mathrm{q}) \Rightarrow (\mathrm{p} \wedge \mathrm{q}))$. It mentions two atomic propositions, p and q, so we have to consider 2^2 possibilities:

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

Notice that this calculation is *compositional* in the structure of P.

slide 31

The Binary Boolean Functions of one and two variables

$2^{(2^1)}$ functions of one variable

P	T	P	$\neg P$	F
T	T	T	F	F
F	T	F	T	F

$2^{(2^2)}$ functions of two variables

P	Q	T	$\vee$		P	$\Rightarrow$	Q	$\Leftrightarrow$	$\wedge$	nand	xor						F
T	T	T	T	T	T	T	T	T	T	F	F	F	F	F	F	F	F
T	F	T	T	T	T	F	F	F	F	T	T	T	T	F	F	F	F
F	T	T	T	F	F	T	T	F	F	T	T	F	F	T	T	F	F
F	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F

slide 32

All boolean functions can be defined in terms of connectives that we have introduced so far (see Ex Sheet 1, Q12).

2.2 Equational reasoning, validity and satisfiability

Equivalences

Identity:

$P \wedge T$ and P have the same meaning

$P \vee F$ and P have the same meaning

Complement:

$P \wedge \neg P$ and F have the same meaning

$P \vee \neg P$ and T have the same meaning

De Morgan:

$\neg(P \wedge Q)$ and $\neg P \vee \neg Q$ have the same meaning

$\neg(P \vee Q)$ and $\neg P \wedge \neg Q$ have the same meaning

Translating away $\Leftrightarrow$:

$P \Leftrightarrow Q$ and $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$ have the same meaning

slide 33

Equivalences

When we say 'P and Q have the same meaning', we really mean 'whatever assumption we make about the truth values of their atomic propositions, P and Q have the same truth value as each other'. In other words, 'P and Q have the same truth table'.

We write that as $P \equiv Q$

slide 34

Equational Reasoning

Equivalences are really useful because they can be used anywhere.

In more detail, this $P \equiv Q$ is a proper notion of equivalence. You can see from its definition that

- it's *reflexive*, i.e., for any proposition P, we have $P \equiv P$
- it's *symmetric*, i.e., if $P \equiv Q$ then $Q \equiv P$
- it's *transitive*, i.e., if $P \equiv Q$ and $Q \equiv R$ then $P \equiv R$

Moreover, if $P \equiv Q$ then we can replace a subformula P by Q in any context, without affecting the meaning of the whole thing. For example, if $P \equiv Q$ then $P \wedge \mathrm{r} \equiv Q \wedge \mathrm{r}$, $\mathrm{r} \wedge P \equiv \mathrm{r} \wedge Q$, $\neg P \equiv \neg Q$, etc.

slide 35

Equational Reasoning

Now we're in business: we can do equational reasoning, replacing equal subformulae by equal subformulae, just as you do in normal algebraic manipulation (where you'd use $2 + 2 = 4$ without thinking).

This complements direct verification using truth tables — sometimes that's more convenient, and sometimes this is. Later, we'll see a third option — structured proof.

slide 36

Some Collected Equivalences, for Reference

For any propositions P, Q, and R

Commutativity:
$P \wedge Q \equiv Q \wedge P$ (and-comm)
$P \vee Q \equiv Q \vee P$ (or-comm)

Unit:
$P \wedge F \equiv F$ (and-unit)
$P \vee T \equiv T$ (or-unit)

Associativity:
$P \wedge (Q \wedge R) \equiv (P \wedge Q) \wedge R$ (and-assoc)
$P \vee (Q \vee R) \equiv (P \vee Q) \vee R$ (or-assoc)

Complement:
$P \wedge \neg P \equiv F$ (and-comp)
$P \vee \neg P \equiv T$ (or-comp)

Distributivity:
$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$ (and-or-dist)
$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$ (or-and-dist)

De Morgan:
$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$ (and-DM)
$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$ (or-DM)

Identity:
$P \wedge T \equiv P$ (and-id)
$P \vee F \equiv P$ (or-id)

Defn:
$P \Rightarrow Q \equiv Q \vee \neg P$ (imp)
$P \Leftrightarrow Q \equiv (P \Rightarrow Q) \wedge (Q \Rightarrow P)$ (bi)

slide 37

Equational Reasoning — Example

Suppose we wanted to prove a 3-way De Morgan law

$\neg(P_1 \wedge P_2 \wedge P_3) \equiv \neg P_1 \vee \neg P_2 \vee \neg P_3$

We could do so either by truth tables, checking 2^3 cases, or by equational reasoning:

$$\begin{array}{rcll} \neg(P_1 \wedge P_2 \wedge P_3) & \equiv & \neg(P_1 \wedge (P_2 \wedge P_3)) & \text{choosing an } \wedge \text{ association} \\ & \equiv & \neg P_1 \vee \neg(P_2 \wedge P_3) & \text{by (and-DM)} \end{array}$$

(and-DM) is $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$. Instantiating the metavariables P and Q as

$$\begin{array}{rcl} P & \mapsto & P_1 \\ Q & \mapsto & P_2 \wedge P_3 \end{array}$$

we get exactly the $\neg(P_1 \wedge (P_2 \wedge P_3)) \equiv \neg P_1 \vee \neg(P_2 \wedge P_3)$ needed.

slide 38

$$\begin{array}{rcll} \neg(P_1 \wedge P_2 \wedge P_3) & \equiv & \neg(P_1 \wedge (P_2 \wedge P_3)) & \text{choosing an } \wedge \text{ association} \\ & \equiv & \neg P_1 \vee \neg(P_2 \wedge P_3) & \text{by (and-DM)} \\ & \equiv & \neg P_1 \vee (\neg P_2 \vee \neg P_3) & \text{by (and-DM)} \end{array}$$

(and-DM) is $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$. Instantiating the metavariables P and Q as

$$\begin{array}{rcl} P & \mapsto & P_2 \\ Q & \mapsto & P_3 \end{array}$$

we get $\neg(P_2 \wedge P_3) \equiv \neg P_2 \vee \neg P_3$. Using that in the context $\neg P_1 \vee \ldots$ gives us exactly the equality $\neg P_1 \vee \neg(P_2 \wedge P_3)) \equiv \neg P_1 \vee (\neg P_2 \vee \neg P_3)$.

$$\equiv \quad \neg P_1 \vee \neg P_2 \vee \neg P_3 \qquad \text{forgetting the } \vee \text{ association}$$

So by transitivity of $\equiv$, we have $\neg(P_1 \wedge P_2 \wedge P_3) \equiv \neg P_1 \vee \neg P_2 \vee \neg P_3$

slide 39

There I unpacked the steps in some detail, so you can see what's really going on. Later, we'd normally just give the brief justification on each line; we wouldn't write down the boxed reasoning (instantiation, context, transitivity) — but it should be clearly in your head when you're doing a proof.

If it's not clear, write it down — *use* the written proof as a tool for thinking.

Still later, you'll use equalities like this one as single steps in bigger proofs.

slide 40

Theorem. Equational reasoning is *sound*: however we instantiate the equations, and chain them together, if we deduce that $P \equiv Q$ then $P \equiv Q$.

Soundness is proved by combining the various facts established in this section so far, but we won't go into detail on the proof of soundness in this course.

Soundness is pragmatically important: if you've faithfully modelled some real-world situation in propositional logic, then you can do any amount of equational reasoning, and the result will be meaningful.

slide 41

Theorem. Equational reasoning *complete*: if $P \equiv Q$, then there is an equational proof.

Proving completeness is beyond the scope of DM1.

Completeness is pragmatically important: if $P \equiv Q$, and you systematically explore all possible candidate equational proofs, eventually you'll find one. But there are infinitely many candidates: at any point, there might be several you could try to apply, and sometimes there are infinitely many instantiations (consider $T \equiv P \vee \neg P$).

slide 42

...so naive proof search is not a decision procedure (but sometimes you can find short proofs).

In contrast, we had a terminating algorithm for checking truth tables (but that's exponential in the number of atomic propositions).

slide 43

Tautology, validity, and satisfiability

Say P is a *tautology*, or is *valid*, if it is always true — i.e., if, whatever assumption we make about the truth values of its atomic propositions, then P is true. In other words, P is a tautology if every row of its truth table is T.

There is a connection with equational reasoning: ($P \equiv Q$) exactly when ($P \Leftrightarrow Q$) is a tautology.

Say P is a *satisfiable* if, under *some* assumption about the truth values of its atomic propositions, P is true.

> $\mathrm{p} \vee \neg \mathrm{p}$ is a tautology (always true, no matter what assumptions are made about p)
>
> $\mathrm{p} \wedge \neg \mathrm{q}$ satisfiable (true under the assumption $\mathrm{p} \mapsto T$, $q \mapsto F$)
>
> $\mathrm{p} \wedge \neg \mathrm{p}$ unsatisfiable (not true under $\mathrm{p} \mapsto T$ or $\mathrm{p} \mapsto F$)

P is unsatisfiable if and only if $\neg P$ is valid.

slide 44

Object, Meta, Meta-Meta,...

We're taking care to distinguish the connectives of the object language that we're studying (propositional logic), and the informal mathematics and English that we're using to talk about it (our meta-language).

For now, we adopt a simple discipline: the former in symbols, the latter in words.

slide 45

Application: Combinational Circuits

Use T and F to represent high and low voltage values on a wire.

Logic gates (AND, OR, NAND, etc.) compute propositional functions of their inputs. Notation: $T, F, \wedge, \vee, \neg$ vs $0, 1, ., +, \overline{}$

SAT solvers: compute satisfiability of propositions with 10 000's of atomic propositions.

slide 46

3 Predicate Logic

Predicate Logic

slide 47

In this section we extend propositional logic with predicates and quantifiers.

Predicate Logic

(or Predicate Calculus, or First-Order Logic)

Socrates is a man. All men are mortal. So Socrates is mortal.

Can we formalise in propositional logic?

Write p for *Socrates is a man*
Write q for *Socrates is mortal*
$\mathrm{p} \qquad \mathrm{p} \Rightarrow \mathrm{q} \qquad \mathrm{q}$

?

slide 48

Predicate Logic

Often, we want to talk about properties of things, not just atomic propositions.

All lions are fierce.
Some lions do not drink coffee.
Therefore, some fierce creatures do not drink coffee.

[Lewis Carroll, 1886]

Let x range over creatures. Write $\mathrm{L}(x)$ for 'x is a lion'. Write $\mathrm{C}(x)$ for 'x drinks coffee'. Write $\mathrm{F}(x)$ for 'x is fierce'.

$\forall\, x.\mathrm{L}(x) \Rightarrow \mathrm{F}(x)$
$\exists\, x.\mathrm{L}(x) \wedge \neg\mathrm{C}(x)$
$\exists\, x.\mathrm{F}(x) \wedge \neg\mathrm{C}(x)$

slide 49

3.1 The Language of Predicate Logic

Predicate Logic

So, we extend the language.

Variables x, y, etc., ranging over some specified domain.

Atomic predicates $\mathrm{A}(x)$, $\mathrm{B}(x)$, etc., like the earlier atomic propositions, but with truth values that depend on the values of the variables.

> Let $\mathrm{A}(x)$ denote $x + 7 = 10$, where x ranges over the natural numbers. $\mathrm{A}(x)$ is true if $x = 3$, otherwise false, so $\mathrm{A}(3) \wedge \neg\mathrm{A}(4)$
> Let $\mathrm{B}(n)$ denote $1 + 2 + ... + n = n(n+1)/2$, where n ranges over the naturals. $\mathrm{B}(n)$ is true for any value of n, so $\mathrm{B}(27)$.

Add these to the language of formulae:

$$P, Q ::= A(x) \mid T \mid F \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \Rightarrow Q \mid P \Leftrightarrow Q$$

where A ranges over atomic predicates A, B, etc.

slide 50

Predicate Logic — Universal Quantifiers

If P is a formula, then $\forall\ x.P$ is a formula

Pronounce $\forall\ x.P$ as 'for all x, P'.

Definition: $\forall\ x.P$ is true if (and only if) P is true for all values of x (taken from its specified domain).

Sometimes we write $P(x)$ for a formula that might mention x, so that we can write (e.g.) $P(27)$ for the formula with x instantiated to 27.

Then, if x is ranging over the naturals,
$\forall\ x.P(x)$ if and only if $P(0)$ and $P(1)$ and $P(2)$ and ...

Or, if x is ranging over $\{\mathrm{red}, \mathrm{green}, \mathrm{blue}\}$,then
$(\forall\ x.P(x)) \Leftrightarrow P(\mathrm{red}) \wedge P(\mathrm{green}) \wedge P(\mathrm{blue})$.

slide 51

Predicate Logic — Existential Quantifiers

If P is a formula, then $\exists\ x.P$ is a formula

Pronounce $\exists\ x.P$ as 'exists x such that P'.

Definition: $\exists\ x.P$ is true if (and only if) there is at least one value of x (taken from its specified domain) such that P is true.

So, if x is ranging over $\{\mathrm{red}, \mathrm{green}, \mathrm{blue}\}$, then $(\exists\ x.P(x))$ if and only if $P(\mathrm{red}) \vee P(\mathrm{green}) \vee P(\mathrm{blue})$.

Because the domain might be infinite, we don't give truth-table definitions for $\forall$ and $\exists$.

Note also that we don't allow infinitary formulae — I carefully *didn't* write
$(\forall\ x.P(x)) \Leftrightarrow P(0) \wedge P(1) \wedge P(2) \wedge ...$ $\times$

slide 52

The Language of Predicate Logic

Summarising, the formulae of predicate logic are the terms of the grammar

$$P, Q \quad ::= \quad A(x) \mid T \mid F \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \Rightarrow Q \mid \\ P \Leftrightarrow Q \mid \forall\, x.P \mid \exists\, x.P$$

Convention: the scope of a quantifier extends as far to the right as possible, so (e.g.) $\forall\, x.\mathrm{A}(x) \wedge \mathrm{B}(x)$ is $\forall x.(\mathrm{A}(x) \wedge \mathrm{B}(x))$, not $(\forall\, x.\mathrm{A}(x)) \wedge \mathrm{B}(x)$.

(other convention — no dot, always parenthesise: $\forall\, x(P)$)

slide 53

Predicate Logic — Extensions

n-ary atomic predicates $\mathrm{A}(x, y)$, $\mathrm{B}(x, y, z)$,...

(regard our old p, q, etc. as 0-ary atomic predicates)

Equality as a special binary predicate $(e = e')$ where e and e' are some mathematical expressions (that might mention variables such as x), and similarly for $<, >, \leq, \geq$ over numbers.

$(e \neq e')$ is shorthand for $\neg(e = e')$

$(e \leq e')$ is shorthand for $(e < e') \vee (e = e')$

slide 54

Predicate Logic — Examples

What do these mean? Are they true or false?

$\exists\, x.(x^2 + 2x + 1 = 0)$ where x ranges over the integers

$\forall\, x.(x < 0) \vee (x = 0) \vee (x \geq 0)$ where x ranges over the reals

$\forall\, x.(x \geq 0) \Rightarrow (2x > x)$ where x ranges over the reals

slide 55

Predicate Logic — Examples

Formalise:

If someone learns discrete mathematics, then they will find a good job. (*)

Let x range over all people.

Write $\mathrm{L}(x)$ to mean 'x learns discrete mathematics'
Write $\mathrm{J}(x)$ to mean 'x will find a good job'

Then $\forall\, x.\mathrm{L}(x) \Rightarrow \mathrm{J}(x)$ is a reasonable formalisation of (*).

Is it true? We'd need to know more...

slide 56

Predicate Logic — Nested Quantifers

What do these mean? Are they true?

$\forall\, x.\forall\, y.(x + y = y + x)$ where x, y range over the integers

$\forall\, x.\exists\, y.(x = y - 10)$ where x, y range over the integers

$\exists\, x.\forall\, y.(x \geq y)$ where x, y range over the integers

$\forall\, y.\exists\, x.(x \geq y)$ where x, y range over the integers

$\exists\, x.\exists\, y.(4x = 2y) \wedge (x + 1 = y)$ where x, y range over the integers

slide 57

Predicate Logic — Examples

Formalise:

Every real number except 0 has a multiplicative inverse

$\forall\ x.(\neg(x = 0)) \Rightarrow \exists\ y.(x\ y = 1)$ where x ranges over the reals

slide 58

Predicate Logic — Free and Bound Variables

A slightly odd (but well-formed) formula:

$\mathrm{A}(x) \wedge (\forall\ x.\mathrm{B}(x) \Rightarrow \exists\ x.\mathrm{C}(x, x))$

Really there are 3 different x's here, and it'd be clearer to write

$\mathrm{A}(x) \wedge (\forall\ x'.\mathrm{B}(x') \Rightarrow \exists\ x''.\mathrm{C}(x'', x''))$ or

$\mathrm{A}(x) \wedge (\forall\ y.\mathrm{B}(y) \Rightarrow \exists\ z.\mathrm{C}(z, z))$

Say an occurrence of x in a formula P is *free* if it is not inside any $(\forall\ x....)$ or $(\exists\ x....)$

All the other occurrences of x are *bound* by the closest enclosing $(\forall\ x....)$ or $(\exists\ x....)$

The *scope* of a quantifier in a formula ...$(\forall\ x.P)$... is all of P (except any subformulae of P of the form $\forall\ x....$ or $\exists\ x....$).

slide 59

Truth Semantics

Whether a formula P is true or false might depend on

1. an interpretation of the atomic predicate symbols used in P (generalising the 'assumptions on its atomic propositions' we had before)
2. the values of the free variables of P

Often 1 is fixed (as it is for $e = e'$)

slide 60

Predicate Logic — Basic Properties

De Morgan laws for quantifiers:

$(\neg\forall\ x.P) \equiv \exists\ x.\neg P$

$(\neg\exists\ x.P) \equiv \forall\ x.\neg P$

Distributing quantifiers over $\wedge$ and $\vee$:

$(\forall\ x.P \wedge Q) \equiv (\forall\ x.P) \wedge (\forall\ x.Q)$

$(\exists\ x.P \wedge Q) \not\equiv (\exists\ x.P) \wedge (\exists\ x.Q)$ $\times$ (left-to-right holds)

$(\forall\ x.P \vee Q) \not\equiv (\forall\ x.P) \vee (\forall\ x.Q)$ $\times$ (right-to-left holds)

$(\exists\ x.P \vee Q) \equiv (\exists\ x.P) \vee (\exists\ x.Q)$

slide 61

Predicate Logic — Examples

Formalise:

Everyone has exactly one best friend.

Let x, y, z range over all people.

Write $\mathrm{B}(x, y)$ to mean y is a best friend of x

Then $\forall\, x.\exists\, y.\mathrm{B}(x, y) \wedge \forall\, z.\mathrm{B}(x, z) \Rightarrow z = y$ is one reasonable formalisation.

Equivalently $\forall\, x.\exists\, y.\mathrm{B}(x, y) \wedge \forall\, z.(\neg(z = y)) \Rightarrow \neg\mathrm{B}(x, z)$.

Um. what about $y = x$?

slide 62

Application: Databases

slide 63

4 Proof

Proof

slide 64

In this section we introduce a structured approach to proof for predicate logic. Proofs are built according to the rules of structure proof, which comprise introduction and elimination rules for each logical connective and the rule of proof by contradiction.

There are some examples of structured proofs in these notes. I will give more examples in the lectures. You can practice using the exercises at the end of the notes, and you can also try writing structured proofs of some of the equivalences for propositional/predicate logic.

Proof

We've now got a rich enough language to express some non-trivial conjectures, e.g.

$\forall\, n.(n > 2) \Rightarrow \neg\exists\, x, y, z.x \neq 0 \wedge y \neq 0 \wedge z \neq 0 \wedge x^n + y^n = z^n$

(where n ranges over the naturals)

Is that true or false?

slide 65

Proof

$\forall\, n.(n > 2) \Rightarrow \neg\exists\, x, y.x \neq 0 \wedge y \neq 0 \wedge z \neq 0 \wedge x^n + y^n = z^n$

We have to be able to reason about this kind of thing, to *prove* that it's true (or to disprove it — to prove its negation...).

This course: 'informal' rigorous proof (normal mathematical practice). A proof is a rigorous argument to convince a very skeptical reader. It should be completely clear, and the individual steps small enough that there's no question about them.

(Later, study 'formal' proofs, as mathematical objects themselves...)

slide 66

Non-Proofs

There are *lots*.

'I have discovered a truly remarkable proof which this margin is too small to contain.'

'I'm your lecturer, and I say it's true'

'The world would be a sad place if this wasn't true'

'I can't imagine that it could be false'

slide 67

Statements

Theorem 1 [associativity of $+$] $\forall\ x, y, z. x + (y + z) = (x + y) + z$

Often leave top-level universal quantifiers implicit (but only in these top-level statements):

Theorem 2 $x + (y + z) = (x + y) + z$

Proposition — a little theorem

Lemma — a little theorem written down as part of a bigger proof

Corollary — an easy consequence of some theorem

any of those should come with a proof attached

Conjecture $x \mod 2 = 0 \vee x \mod 3 = 0 \vee x \mod 5 = 0$

slide 68

Structured Proof

The truth-table and equational reasoning from before is still sound, but we need more, to reason about the quantifiers. And truth tables aren't going to help there.

Going to focus instead on the structure of the formulae we're trying to prove (and of those we can use).

Practice on statements about numbers — not that we care about these results particularly, but just to get started.

slide 69

Theorem? The sum of two rationals is rational.

Clarify the logical form:

Theorem?
$\forall\ x. \forall\ y. (\mathrm{Rational}(x) \wedge \mathrm{Rational}(y)) \Rightarrow \mathrm{Rational}(x + y)$

and the definitions:

Say $\mathrm{Rational}(x)$ if $\exists\ n, m. (x = n/m)$

where x and y range over real numbers and n and m range over integers.

Sometimes this clarification is a major intellectual activity (and the subsequent proof might be easy); sometimes it's easy to state the problem (but the proof is very hard).

How *far* we have to clarify the definitions depends on the problem — here I didn't define the reals, integers, addition, or division.

slide 70

In the lectures we will carefully study a proof of this statement about sums of rational numbers. Most mathematicians would prove the statement by writing something like the following text.

Theorem $\forall\, x.\forall\, y.(\text{Rational}(x) \wedge \text{Rational}(y)) \Rightarrow \text{Rational}(x+y)$

Proof: Consider arbitrary real numbers x and y. Suppose that they are both rational. We must show that the sum $(x+y)$ is rational too. Since x and y are both rational, by definition, there are integers m_1, n_1, m_2 and n_2 such that $x = \frac{n_1}{m_1}$ and $y = \frac{n_2}{m_2}$. We can now use the laws of arithmetic:

$$x + y \;=\; \frac{n_1}{m_1} + \frac{n_2}{m_2} \;=\; \frac{n_1}{m_1}\cdot\frac{m_2}{m_2} + \frac{m_1}{m_1}\cdot\frac{n_2}{m_2} \;=\; \frac{n_1 m_2}{m_1 m_2} + \frac{m_1 n_2}{m_1 m_2} \;=\; \frac{n_1 m_2 + m_1 n_2}{m_1 m_2}.$$

Another basic fact of arithmetic is that $(n_1 m_2 + m_1 n_2)$ and $m_1 m_2$ are both integers, and so $(x+y)$ can be written as a fraction of integers. In other words, $(x+y)$ is rational. $\Box$

What makes this proof correct? Sometimes proofs that are written like this look convincing, but they turn out to be wrong. You need to learn to write correct proofs and to distinguish good arguments from bad ones. To this end, in this course, we will study proofs in the following more formal layout.

Theorem $\forall\, x.\forall\, y.(\text{Rational}(x) \wedge \text{Rational}(y)) \Rightarrow \text{Rational}(x+y)$

Proof:

1. Consider an arbitrary real number x [*aim to prove:* $\forall\, y.(\text{Rat}(x) \wedge \text{Rat}(y)) \Rightarrow \text{Rat}(x+y)$]

2. Consider an arbitrary real number y [*aim to prove:* $(\text{Rat}(x) \wedge \text{Rat}(y)) \Rightarrow \text{Rat}(x+y)$]

3. Assume $\text{Rational}(x) \wedge \text{Rational}(y)$ [*aim to prove:* $\text{Rat}(x+y)$]

4. $\text{Rational}(x)$ from 3 by $\wedge$-elimination

5. $\text{Rational}(y)$ from 3 by $\wedge$-elimination

6. $\exists\, n, m.(x = n/m)$ from 4 by unfolding the definition of Rational

7. $\exists\, n, m.(y = n/m)$ from 5 by unfolding the definition of Rational

8. Consider actual integers n_1 and m_1 such that $x = n_1/m_1$
[*aim to prove:* $\exists\, n, m.(x + y = n/m)$ *by eliminating* $\exists$ *from 6*]

9. Consider actual integers n_2 and m_2 such that $y = n_2/m_2$
[*aim to prove:* $\exists\, n, m.(x + y = n/m)$ *by eliminating* $\exists$ *from 7*]

10. $x + y = (n_1/m_1) + (n_2/m_2)$ from 8 and 9, adding both sides

11. $\quad = \frac{n_1\ m_2}{m_1\ m_2} + \frac{m_1\ n_2}{m_1\ m_2}$ from 10, by arithmetic

12. $\quad = \frac{n_1\ m_2 + m_1\ n_2}{m_1\ m_2}$ from 11, by arithmetic

13. $\exists\, n, m.x + y = n/m$ from 10–12, $\exists$-introduction,
witness $n = n_1\ m_2 + m_1\ n_2$
$m = m_1\ m_2$

14. $\exists\, n, m.x + y = n/m$ from 7, 9–13, $\exists$-elimination

15. $\exists\, n, m.x + y = n/m$ from 6, 8–14, $\exists$-elimination

16. $\text{Rational}(x+y)$ from 15, folding the definition of Rational

17. $(\text{Rational}(x) \wedge \text{Rational}(y)) \Rightarrow \text{Rational}(x+y)$ by $\Rightarrow$-introduction, from 3–16

18. $\forall\, y.(\text{Rational}(x) \wedge \text{Rational}(y)) \Rightarrow \text{Rational}(x+y)$ by $\forall$-introduction, from 2–17

17. $\forall\, x.\forall\, y.(\text{Rational}(x) \wedge \text{Rational}(y)) \Rightarrow \text{Rational}(x+y)$ by $\forall$-introduction, from 1–16 $\Box$

What is a Proof (in this stylised form)?

A list of lines, each of which is either:

- a formula of predicate logic, with a justification (‘P, from ... by ...’)
- an assumption of some formula (‘Assume P’)
- an introduction of a arbitrary variable (‘Consider an arbitrary x (from the appropriate domain)’)
- an introduction of some actual witness variables and a formula (‘For some actual n, P’)

When we make an assumption, we open a box. We have to close it before we can discharge the assumption (by $\Rightarrow$-introduction at step 17).

slide 71

What is a Proof (in this stylised form)?

Lines are numbered

Introduced variables must be *fresh* (not free in any preceeding formula).

The justifications must not refer to later lines (no circular proofs, please!)

1. P by ... from 15 $\times$

...

15. Q by ... from 1

slide 72

What is a Proof (in this stylised form)?

The justifications must not refer to lines inside any earlier box

1. Assume P
...
15. U from ... by ...
...
27. Q from ... by ...

28. $P \Rightarrow Q$ by $\Rightarrow$-introduction, from 1–27

29. Assume R
...
1007. ... from 15 by ... $\times$

(earlier in an enclosing box is ok)

slide 73

What is a Justification (in this stylised form)?
Back to the Connectives — And

To use a conjunction: if we know $P \wedge Q$, then we can deduce P, or we can deduce Q (or both, as often as we like)

...
$m.$ $P \wedge Q$ from ...
...
$n.$ P from m by $\wedge$-elimination

or

...
$m.$ $P \wedge Q$ from ...
...
$n.$ Q from m by $\wedge$-elimination

slide 74

What is a Justification (in this stylised form)?
Back to the Connectives — And

To prove a conjunction: we can prove $P \wedge Q$ by proving P and proving Q.

...
$l.$ P from ...
...
$m.$ Q from ...
...
$n.$ $P \wedge Q$ from l and m by $\wedge$-introduction

(it doesn't matter in what order l and m are in)

slide 75

What is a Justification (in this stylised form)?
Back to the Connectives — Implication

To prove an implication: to prove $P \Rightarrow Q$, assume P, prove Q, and discharge the assumption.

...
$m.$ Assume P
...
$n.$ Q from ... by ...
$n+1.$ $P \Rightarrow Q$ from m–n, by $\Rightarrow$-introduction

slide 76

What is a Justification (in this stylised form)?
Back to the Connectives — Implication

To use an implication: if we know $P \Rightarrow Q$, and we know P, we can deduce Q

...
$l.$ $P \Rightarrow Q$ by ...
...
$m.$ P by ...
...
$n.$ Q from l and m by $\Rightarrow$-elimination

(also known as *modus ponens*)

slide 77

What is a Justification (in this stylised form)?
Back to the Connectives — Or

To prove a disjunction: to prove $P \vee Q$, we could prove P, or we could prove Q. (could even use $\neg Q$ or $\neg P$ resp.)

...
$m.$ P from ...
...
$n.$ $P \vee Q$ from m by $\vee$-introduction

or

...
$m.$ Q from ...
...
$n.$ $P \vee Q$ from m by $\vee$-introduction

slide 78

What is a Justification (in this stylised form)?

Back to the Connectives — Or

To use a disjunction: if we know $P \vee Q$, and by assuming P we can prove R, and by assuming Q we can prove R, then we can deduce R (a form of case analysis).

l. $P \vee Q$ from ... by ...

...

m_1. Assume P
...
m_2. R

...

n_1. Assume Q
...
n_2. R

...

o. R from l, m_1–m_2, n_1–n_2 by $\vee$-elimination

(it doesn't matter what order l, m_1–m_2, and n_1–n_2 are in)

slide 79

What is a Justification (in this stylised form)?

Back to the Connectives — Negation

To prove a negation: to prove $\neg P$, assume P, prove F, and discharge the assumption.

...

m. Assume P
...
n. F from ... by ...

$n + 1$. $\neg P$ from m–n, by $\neg$-introduction

That's a lot like $\Rightarrow$-introduction (not a surprise, as $\neg P \equiv (P \Rightarrow F)$).

slide 80

What is a Justification (in this stylised form)?

Back to the Connectives — Negation

To use a negation: if we know $\neg P$, and we know P, we can deduce F

...

l. P by ...

...

m. $\neg P$ by ...

...

n. F from l and m by $\neg$-elimination

slide 81

What is a Justification (in this stylised form)?

Back to the Connectives — Truth

To prove T: nothing to do

...

n. T-introduction.

There's no elimination rule for T.

slide 82

What is a Justification (in this stylised form)?
Falsity

If we can deduce F, then we can deduce any P

...
$m.$ F from ... by ...
...
$n.$ P from m, by F-elimination.

(hopefully this would be under some assumption(s)...)

There is no introduction rule for F.

slide 83

What is a Justification (in this stylised form)?
Contradiction

To prove P by contradiction: if, from assuming $\neg P$, we can prove F, then we can deduce P

...
$m.$ Assume $\neg P$
...
$n.$ F from ... by ...
$n+1.$ P from m–n, by contradiction

Note that in the other rules either a premise (for elimination rules) or the conclusion (for introduction rules) had some particular form, but here the conclusion is an arbitrary P.

slide 84

Example

Theorem $(P \wedge Q) \Rightarrow (P \vee Q)$

Proof:

1. Assume $P \wedge Q$
2. P from 1 by $\wedge$-elim
3. $P \vee Q$ from 2 by $\vee$-intro

4. $(P \wedge Q) \Rightarrow (P \vee Q)$ from 1–3 by $\Rightarrow$-intro

□

slide 85

Example

Theorem ? $(P \vee Q) \Rightarrow (P \wedge Q)$

Proof ?:

1. Assume $P \vee Q$
2. ...use $\vee$-elim somehow? prove by contradiction?

????

$n-2.$ P from ? by ?
$n-1.$ Q from ? by ?
$n.$ $(P \wedge Q)$ from $n-1, n-2$ by $\wedge$-intro

$n+1.$ $(P \vee Q) \Rightarrow (P \wedge Q)$ from 1–n by $\Rightarrow$-intro

Counterexample? Prove negation?

slide 86

What is a Justification (in this stylised form)?

Back to the Connectives — For all

To use a universally quantified formula: if we know $\forall x.P(x)$, then we can deduce $P(v)$ for any v (of the appropriate domain)

...

$m.$ $\forall\, x.P(x)$ from ...

...

$n.$ $P(v)$ from m by $\forall$-elimination

slide 87

What is a Justification (in this stylised form)?

Back to the Connectives — For all

To prove a universally quantified formula $\forall\, x.P(x)$, consider an arbitrary fresh variable x (ranging over the appropriate domain) and prove $P(x)$, then discharge the assumption.

...

$m.$ Consider an arbitrary x (from domain ...)
...
$n.$ $P(x)$ by ...

$n+1.$ $\forall\, x.P(x)$ from m–n by $\forall$-introduction

slide 88

What is a Justification (in this stylised form)?

Back to the Connectives — Exists

To prove an existentially quantified formula $\exists\, x.P(x)$, prove $P(v)$ for some witness v (from the appropriate domain).

...

$m.$ $P(v)$

...

$n.$ $\exists\, x.P(x)$ from m by $\exists$-introduction with witness $x = v$

slide 89

What is a Justification (in this stylised form)?

Back to the Connectives — Exists

To use an existentially quantified formula $\exists\, x.P(x)$, introduce a fresh variable (ranging over the appropriate domain) x_1, about which we know only $P(x_1)$. The elimination rule for existential quantifiers is reminiscent of the elimination rule for disjunction.

$l.$ $\exists\, x.P(x)$

...

$m.$ For some actual x_1, $P(x_1)$
...
$n.$ Q (where x_1 not free in Q)

...

$o.$ Q from l, m–n, by $\exists$-elimination

slide 90

Digression on $\exists$-instantiation: When you eliminate existential quantifiers, there are usually many reasonable places to close the box. Some logicians argue that it doesn't really matter where you actually close the box as long as it can be closed. This has lead some authors to describe '*Existential Instantiation*':

$m.$ $\exists\, x.P(x)$

...

$n.$ For some actual x_1, $P(x_1)$ from m by $\exists$-instantiation

Proper accounts of ∃-instantiation come with things to check about appearance of variables in completed proofs, which amount to checking that the boxes in ∃-elimination can be closed. For example, the conditions ensure that the statement $(\exists\, x.P(x)) \Rightarrow P(y)$ is not provable.

Example

Many theorems have a similar top-level structure, e.g.

$\forall\, x, y, z.(P \wedge Q \wedge R) \Rightarrow S$

1. Consider an arbitrary x, y, z.

   2. Assume $P \wedge Q \wedge R$.
   3. P from 2 by ∧-elimination
   4. Q from 2 by ∧-elimination
   5. R from 2 by ∧-elimination

   ...

   215. S by ...

216. $(P \wedge Q \wedge R) \Rightarrow S$ from 2–215 by ⇒-introduction

217. $\forall\, x, y, z.(P \wedge Q \wedge R) \Rightarrow S$ by ∀-introduction, from 1–216

slide 91

What is a Proof (in this stylised form)?

NB This particular stylised form is only one way to write down rigorous paper proofs. It's a good place to start, but its not always appropriate. Later, you'll sometimes take bigger steps, and won't draw the boxes.

But however they are written, they *have* to be written down clearly — a proof is a communication tool, to persuade. Each step needs a justification.

In questions, we'll say specifically "by structured proof", "by equational reasoning", "by truth tables", or, more generally "prove".

slide 92

This notation for 'natural deduction' proofs was first used by Jaśacowski in the 1920s and it was developed by Fitch in the 1950s. It is used in various books, including the book by Bornat. If you want, you can try building proofs using the Jape assistant, by following the links on the course materials web page: `www.cl.cam.ac.uk/teaching/current/DiscMathI/materials.html`. In 1B Logic & Proof you will see a different, tree-like notation for natural deduction proofs.

Soundness and Completeness?

Are these proof rules *sound*? (i.e., are all the provable formulae valid?)

Are these proof rules *complete*? (i.e., are all valid formulae provable?)

Think about *proof search*

slide 93

Aside: Writing Discrete Maths

By hand

In ASCII

```
P ::= T | F | p | A(x) | P /\ Q | P \/ Q
    | P=>Q | P<=>Q | !x.P | ?x.P
```

In LaTeX (but don't forget that typesetting is *not* real work)

slide 94

Pragmatics

Given some conjecture:

1. Ensure the statement is well-defined, and that you know the definitions of whatever it uses.
2. Understand intuitive what it's saying. Verbalize it.
3. Intuitively, why is it true? (or false?)
4. What are the hard (or easy) cases likely to be?
5. Choose a strategy — truth tables, equational reasoning, structured proof, induction, ...
6. Try it! (but be prepared to backtrack)
7. Expand definitions and make abbreviations as you need them.
8. Writing — to communicate, and to help you think.

slide 95

9. Choose variable names carefully; take care with parentheses
10. Use enough words and use enough symbols, but keep them properly nested. Don't use random squiggles ("$\Rightarrow$" or "$\because$") for meta-reasoning.
11. If it hasn't worked yet... either
    (a) you've make some local mistake (mis-instantiated, re-used a variable name, not expanded definitions enough, forgotten a useful assumption). Fix it and continue.
    (b) you've found that the conjecture is false. Construct a simple counterexample and check it.
    (c) you need to try a different strategy (different induction principle, strengthened induction hypothesis, proof by contradictions,...)
    (d) you didn't really understand intuitively what the conjecture is saying, or what the definitions it uses mean. Go back to them again.

slide 96

12. If it has worked: read through it, skeptically. Maybe *re-write* it.
13. Finally, give it to someone else, as skeptical and careful as you can find, to see if they believe it — to see if they believe that *what you've written down is a proof*, not that they believe that *the conjecture is true*.

slide 97

...more fallacies

slide 98

Summary of the rules of structured proof. *Slides 74–90*

	Introduction rules	***Elimination rules***
$\wedge$	... $l.$ P from ... ... $m.$ Q from ... ... $n.$ $P \wedge Q$ from l and m by $\wedge$-introduction (it doesn't matter in what order l and m are in)	... $m.$ $P \wedge Q$ from ... ... $n.$ P from m by $\wedge$-elimination or ... $m.$ $P \wedge Q$ from ... ... $n.$ Q from m by $\wedge$-elimination
$\vee$	... $m.$ P from ... ... $n.$ $P \vee Q$ from m by $\vee$-introduction or ... $m.$ Q from ... ... $n.$ $P \vee Q$ from m by $\vee$-introduction	$l.$ $P \vee Q$ from ... by ... ... m_1. Assume P ... m_2. R ... n_1. Assume Q ... n_2. R ... $o.$ R from l, m_1–m_2, n_1–n_2 by $\vee$-elimination (it doesn't matter what order l, m_1–m_2, and n_1–n_2 are in)
$\Rightarrow$	... $m.$ Assume P ... $n.$ Q from ... by ... $n+1.$ $P \Rightarrow Q$ from m–n, by $\Rightarrow$-introduction	... $l.$ $P \Rightarrow Q$ by ... ... $m.$ P by ... ... $n.$ Q from l and m by $\Rightarrow$-elimination
$\neg$	... $m.$ Assume P ... $n.$ F from ... by ... $n+1.$ $\neg P$ from m–n, by $\neg$-introduction	... $l.$ P by ... ... $m.$ $\neg P$ by ... ... $n.$ F from l and m by $\neg$-elimination
T	... $n.$ T	*No elimination rule for True.*
F	*No introduction rule for False.*	... $m.$ F from ... by ... ... $n.$ P from m, by F-elimination
$\forall$	... $m.$ Consider an arbitrary x (from domain ...) ... $n.$ $P(x)$ by ... $n+1.$ $\forall\, x.P(x)$ from m–n by $\forall$-introduction	... $m.$ $\forall\, x.P(x)$ from ... ... $n.$ $P(v)$ from m by $\forall$-elimination
$\exists$	... $m.$ $P(v)$ ... $n.$ $\exists\, x.P(x)$ from m by $\exists$-introduction with witness $x = v$	$l.$ $\exists\, x.P(x)$ ... $m.$ For some actual x_1, $P(x_1)$ ... $n.$ Q (where x_1 not free in Q) ... $o.$ Q from l, m–n, by $\exists$-elimination
	... $m.$ Assume $\neg P$ ... $n.$ F from ... by ... $n+1.$ P from m–n, by contradiction	*(Proof by contradiction)*

5 Set Theory

Set Theory

slide 99

In this section we will discuss sets. We will discuss how to describe sets and how to reason about sets. We will study relations and graphs by considering sets of pairs.

Set Theory

Now we've got some reasoning techniques, but not much to reason *about*.
Let's add *sets* to our language.

What is a set? An unordered collection of *elements*:

$\{0, 3, 7\} = \{3, 0, 7\}$

might be empty:

$\{\} = \emptyset = \varnothing$

might be infinite:

$$\begin{aligned} \mathbb{N} &= \{0, 1, 2, 3...\} \\ \mathbb{Z} &= \{..., -1, 0, 1, ...\} \\ \mathbb{R} &= \text{...all the real numbers} \end{aligned}$$

slide 100

Some more interesting sets

the set of nodes in a network (encode with $\mathbb{N}$?)

the set of paths between such nodes (encode ??)

the set of polynomial-time computable functions from naturals to naturals

the set of well-typed programs in some programming language (encode???)

the set of executions of such programs

the set of formulae of predicate logic

the set of valid proofs of such formulae

the set of all students in this room (?)

the set of all sets $\times$

slide 101

Basic relationships

membership $x \in A$

$3 \in \{1, 3, 5\}$
$2 \notin \{1, 3, 5\}$
(of course $(2 \notin \{1, 3, 5\})$ iff $\neg(2 \in \{3, 5, 1\})$)

equality between sets $A = B$ means $\forall\, x.x \in A \Leftrightarrow x \in B$

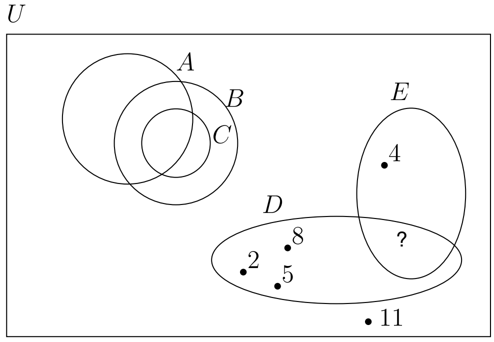
$\{1, 2\} = \{2, 1\} = \{2, 1, 2, 2\}$ $\qquad$ $\{\} \neq \{\{\}\}$

inclusion or *subset* $A \subseteq B$ means $\forall\, x.x \in A \Rightarrow x \in B$

Properties: $\subseteq$ is reflexive, transitive,
and antisymmetric ($(A \subseteq B \wedge B \subseteq A) \Rightarrow A = B$)
but not total: $\{1, 2\} \not\subseteq \{1, 3\} \not\subseteq \{1, 2\}$

slide 102

Venn Diagrams



slide 103

Bounded Quantifiers

Write

$\forall\, x \in A.P$ for $\forall\, x.x \in A \Rightarrow P$

$\exists\, x \in A.P$ for $\exists\, x.x \in A \wedge P$

where A is a subset of the domain that x ranges over.

Define Even to be the set of all even naturals

Then can write $\forall\, n \in \mathrm{Even}.\exists\, m \in \mathbb{N}.n = 3m$

slide 104

Building interesting subsets with set comprehension

$\mathrm{Even} \stackrel{\text{def}}{=} \{n \mid \exists\, m \in \mathbb{N}.n = 2m\}$

$\{x \mid x \in \mathbb{N} \wedge \neg\exists\, y, z \in \mathbb{N}.y > 1 \wedge z > 1 \wedge y\ z = x\}$

$\{x \mid x \in \mathbb{N} \wedge \forall\, y \in \mathbb{N}.y > x\}$

$\{2\ x \mid x \in \mathbb{N}\}$

slide 105

From sets to predicates, and back again

From sets to predicates: given a set A, can define a predicate
$P(x) \stackrel{\text{def}}{=} x \in A$

From predicates to sets: given $P(x)$ and some set U, can build a set
$A \stackrel{\text{def}}{=} \{x \mid x \in U \wedge P(x)\}$

(in some logics we'd really identify the two concepts – but not here)

Property of comprehensions: $x \in \{y \mid P(y)\} \Leftrightarrow P(x)$

slide 106

Building new sets from old ones: union, intersection, and difference

$A \cup B \stackrel{\text{def}}{=} \{x \mid x \in A \vee x \in B\}$

$A \cap B \stackrel{\text{def}}{=} \{x \mid x \in A \wedge x \in B\}$

$A - B \stackrel{\text{def}}{=} \{x \mid x \in A \wedge x \notin B\}$

A and B are *disjoint* when $A \cap B = \{\}$ (symm, not refl or tran)

slide 107

Building new sets from old ones: union, intersection, and difference

$\{1,2\} \cup \{2,3\} = \{1,2,3\}$

$\{1,2\} \cap \{2,3\} = \{2\}$

$\{1,2\} - \{2,3\} = \{1\}$

slide 108

Properties of union, intersection, and difference

Recall $\vee$ is associative: $P \vee (Q \vee R) \equiv (P \vee Q) \vee R$

Theorem $A \cup (B \cup C) = (A \cup B) \cup C$

Proof

$A \cup (B \cup C)$
1. $= \{x \mid x \in A \vee x \in (B \cup C)\}$ unfold defn of union
2. $= \{x \mid x \in A \vee x \in \{y \mid y \in B \vee y \in C\}\}$ unfold defn of union
3. $= \{x \mid x \in A \vee (x \in B \vee x \in C)\}$ comprehension property
4. $= \{x \mid (x \in A \vee x \in B) \vee x \in C\}$ by $\vee$ assoc
5. $= (A \cup B) \cup C$ by the comprehension property and folding defn of union twice $\square$

slide 109

Some Collected Set Equalities, for Reference

For any sets A, B, and C, all subsets of U

Commutativity:

$A \cap B = B \cap A$ ($\cap$-comm)

$A \cup B = B \cup A$ ($\cup$-comm)

Associativity:

$A \cap (B \cap C) = (A \cap B) \cap C$ ($\cap$-assoc)

$A \cup (B \cup C) = (A \cup B) \cup C$ ($\cup$-assoc)

Distributivity:

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ ($\cap$-$\cup$-dist)

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ ($\cup$-$\cap$-dist)

Identity:

$A \cap U = A$ ($\cap$-id)

$A \cup \{\} = A$ ($\cup$-id)

Unit:

$A \cap \{\} = \{\}$ ($\cap$-unit)

$A \cup U = U$ ($\cup$-unit)

Complement:

$A \cap (U - A) = \{\}$ ($\cap$-comp)

$A \cup (U - A) = U$ ($\cup$-comp)

De Morgan:

$U - (A \cap B) = (U - A) \cup (U - B)$ ($\cap$-DM)

$U - (A \cup B) = (U - A) \cap (U - B)$ ($\cup$-DM)

slide 110

Example Proof

Theorem $\{\} \subseteq A$

Proof

$\{\} \subseteq A$

1. $\equiv \forall x. x \in \{\} \Rightarrow x \in A$ unfolding defn of $\subseteq$
2. $\equiv \forall x. F \Rightarrow x \in A$ use defn of $\in$
3. $\equiv \forall x. T$ equational reasoning with $(F \Rightarrow P) \equiv T$
4. $\equiv T$ using defn of $\forall$ □

slide 111

Another Proof of the Same Theorem

Theorem $\{\} \subseteq A$

Another Proof (using the structured rules more explicitly)

1. Note that $\{\} \subseteq A$ means $\forall x. x \in \{\} \Rightarrow x \in A$ (unfolding defn of $\subseteq$)

We prove the r.h.s.:

2. Consider an arbitrary x
3. Assume $x \in \{\}$
4. F by defn of $\in$
5. $x \in A$ from 4, by F-elimination
6. $x \in \{\} \Rightarrow x \in A$ from 3–5, by $\Rightarrow$-introduction
7. $\forall x. x \in \{\} \Rightarrow x \in A$ from 2–6, by $\forall$-introduction □

slide 112

Building new sets from old ones: powerset

Write $\mathcal{P}(A)$ for the set of all subsets of a set A.

$\mathcal{P}\{\} = \{\{\}\}$

$\mathcal{P}\{7\} = \{\{\}, \{7\}\}$

$\mathcal{P}\{1, 2\} = \{\{\}, \{1\}, \{2\}, \{1, 2\}\}$

$A \in \mathcal{P}(A)$

(why ‘power’ set?)

slide 113

Building new sets from old ones: product

Write (a, b) (or sometimes $\langle a, b\rangle$) for an ordered pair of a and b

$A \times B \stackrel{\text{def}}{=} \{(a, b) \mid a \in A \wedge b \in B\}$

Similarly for triples $(a, b, c) \in A \times B \times C$ etc.

Pairing is *non-commutative*: $(a, b) \neq (b, a)$ unless $a = b$

Pairing is *non-associative* and distinct from 3-tupling etc:
$(a, (b, c)) \neq (a, b, c) \neq ((a, b), c)$ and
$A \times (B \times C) \neq A \times B \times C \neq (A \times B) \times C$

Why 'product'?
$\{1, 2\} \times \{\text{red}, \text{green}\} = \{(1, \text{red}), (2, \text{red}), (1, \text{green}), (2, \text{green})\}$

slide 114

We know $(a, b) = (b, a) \Rightarrow a = b$ for pairs

so why not lift the result to set product?

Theorem ? $(A \times B = B \times A) \Rightarrow A = B$

Proof?

The first components of the pairs in $A \times B$ are from A.

The first components of the pairs in $B \times A$ are from B.

If $A \times B = B \times A$ then these must be the same, so $A = B$.

slide 115

Theorem ? $(A \times B = B \times A) \Rightarrow A = B$

Proof?

1. Assume $A \times B = B \times A$

We prove $A = B$, i.e. $\forall\, x. x \in A \Leftrightarrow x \in B$

2. Consider an arbitrary x.

We first prove the $\Rightarrow$ implication.

3. Assume $x \in A$.
4. Consider an arbitrary $y \in B$.
5. $(x, y) \in A \times B$ by defn $\times$
6. $(x, y) \in B \times A$ by 1
7. $x \in B$ by defn $\times$
8. $x \in A \Rightarrow x \in B$ from 3–7 by $\Rightarrow$-introduction
9. The proof of the $\Leftarrow$ implication is symmetric
10. $\forall\, x. x \in A \Leftrightarrow x \in B$ from 2–9 by $\forall$-introduction

slide 116

Theorem
$(A \times B = B \times A) \wedge (\exists\, x. x \in A) \wedge (\exists\, y. y \in B) \Rightarrow A = B$

Proof

1. Assume $A \times B = B \times A \wedge (\exists\, x. x \in A) \wedge (\exists\, y. y \in B)$

1a. $A \times B = B \times A$ from 1 by $\wedge$-elimination

1b. $(\exists\, x. x \in A)$ from 1 by $\wedge$-elimination

1c. $(\exists\, y. y \in B)$ from 1 by $\wedge$-elimination

We prove $A = B$, i.e. $\forall\, x. x \in A \Leftrightarrow x \in B$

2. Consider an arbitrary x.

We first prove the $\Rightarrow$ implication.

3. Assume $x \in A$.
4. Consider some actual $y \in B$
5. $(x, y) \in A \times B$ by defn $\times$
6. $(x, y) \in B \times A$ by 1a
7. $x \in B$ by defn $\times$
8. $x \in B$ from 1c,4–7 by $\exists$-elimination
9. $x \in A \Rightarrow x \in B$ from 3–8 by $\Rightarrow$-introduction
10. The proof of the $\Leftarrow$ implication is symmetric
11. $\forall\, x. x \in A \Leftrightarrow x \in B$ from 2–10 by $\forall$-introduction $\square$

slide 117

Theorem

$(A \times B = B \times A) \wedge (\exists\, x. x \in A) \wedge (\exists\, y. y \in B) \Rightarrow A = B$

slide 118

Aside

Let $A \stackrel{\text{def}}{=} \{n \mid n = n+1\}$

Is $\forall\, x \in A. x = 7$ true?

Or $\forall\, x \in A. x = x + 1$? Or $\forall\, x \in A. 1 = 2$?

Is $\exists\, x \in A. 1 + 1 = 2$ true?

slide 119

5.1 Relations, Graphs, and Orders

Relations, Graphs, and Orders

slide 120

Using Products: Relations

Say a (binary) *relation* R between two sets A and B is a subset of all the (a, b) pairs (where $a \in A$ and $b \in B$)

$R \subseteq A \times B$ (or, or course, $R \in \mathcal{P}(A \times B)$)

Extremes: $\varnothing$ and $A \times B$ are both relations between A and B

$1_A \stackrel{\text{def}}{=} \{(a, a) \mid a \in A\}$ is the *identity relation* on A

$\varnothing \subseteq 1_A \subseteq A \times A$

Sometimes write infix: $a\ R\ b \stackrel{\text{def}}{=} (a, b) \in R$

slide 121

Relational Composition

Given $R \subseteq A \times B$ and $S \subseteq B \times C$, their *relational composition* is

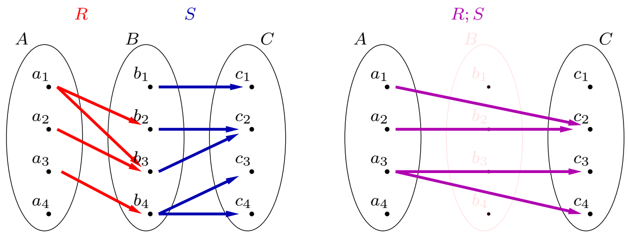
$R; S \stackrel{\text{def}}{=} \{(a, c) \mid \exists\, b. (a, b) \in R \wedge (b, c) \in S\}$

$R; S \subseteq A \times C$

Sometimes write that the other way round: $S \circ R \stackrel{\text{def}}{=} R; S$
(to match function composition)

slide 122

Relational Composition



$A \stackrel{\text{def}}{=} \{a_1, a_2, a_3, a_4\}$ $B \stackrel{\text{def}}{=} \{b_1, b_2, b_3, b_4\}$ $C \stackrel{\text{def}}{=} \{c_1, c_2, c_3, c_4\}$

$R \stackrel{\text{def}}{=} \{(a_1, b_2), (a_1, b_3), (a_2, b_3), (a_3, b_4)\}$

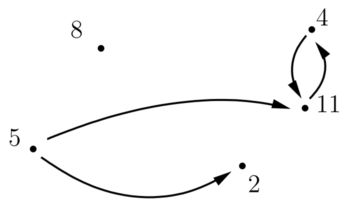
$S \stackrel{\text{def}}{=} \{(b_1, c_1), (b_2, c_2), (b_3, c_2), (b_4, c_3), (b_4, c_4)\}$

$R; S = \{(a_1, c_2), (a_2, c_2), (a_3, c_3), (a_3, c_4)\}$

slide 123

Relations as Directed Graphs

Relations from a set to itself

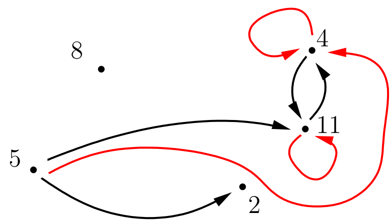


$G \subseteq \mathbb{N} \times \mathbb{N}$

$G = \{(5, 2), (5, 11), (4, 11), (11, 4)\}$

slide 124

Transitivity



$R \subseteq A \times A$

$R^+ \stackrel{\text{def}}{=} R \cup (R; R) \cup (R; R; R) \cup \ldots$

$G^+ = \{(5, 2), (5, 11), (4, 11), (11, 4)\} \cup \{(5, 4), (11, 11), (4, 4)\}$

R *is transitive* if $R = R^+$

slide 125

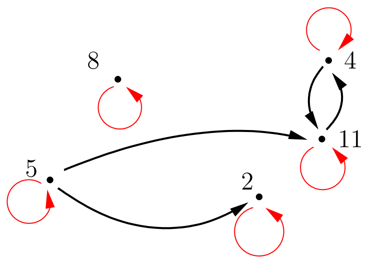
Directed Acyclic Graphs (DAGs)

$R \subseteq A \times A$ represents a *directed acyclic graph* if its transitive closure R^+ is *acyclic*, i.e.

$\neg \exists\, a \in A.(a, a) \in R^+$

slide 126

Reflexivity

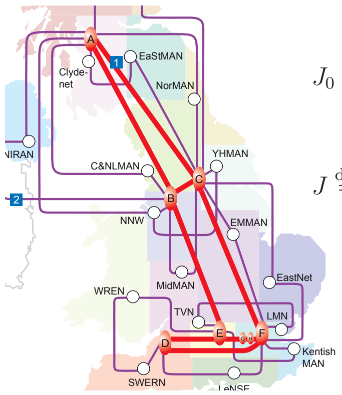


$A \overset{\text{def}}{=} \{2, 4, 5, 8, 11\}$

$G \cup 1_A =$
$\{(5,2), (5,11), (4,11), (11,4), (2,2), (4,4), (5,5), (8,8), (11,11)\}$

$R \subseteq A \times A$ *is reflexive* (over A) if $\forall\, a \in A.(a, a) \in R$

slide 127



This is an *undirected* graph

$J_0 \overset{\text{def}}{=} \{(A,B), (A,C), (B,C), (B,E), (C,F), (E,D), (D,F), (E,G), (G,H), (H,F)\}$

$J \overset{\text{def}}{=} J_0 \cup J_0^{-1}$

where the inverse of R is

$R^{-1} \overset{\text{def}}{=} \{(y,x) | (x,y) \in R\}$

so J is *symmetric*, i.e.

$J = J^{-1}$

slide 128

Equivalence Relations

$R \subseteq A \times A$ is an *equivalence relation* (over A) if:

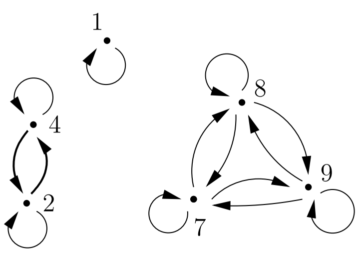
- R is reflexive, i.e. $\forall\, a \in A.(a, a) \in R$
- R is transitive,
  i.e. $\forall\, a_1, a_2, a_3 \in A.((a_1, a_2) \in R \wedge (a_2, a_3) \in R) \Rightarrow (a_1, a_3) \in R$
- R is symmetric, i.e. $\forall\, a_1, a_2 \in A.(a_1, a_2) \in R \Rightarrow (a_2, a_1) \in R$

e.g.$\{(m, n) \mid m \text{ mod } 3 = n \text{ mod } 3\}$ (over $\mathbb{N}$)

The *equivalence class* of $a \in A$ is all the things related to it, i.e.
$\{a' \mid (a, a') \in R\}$

slide 129

Equivalence Relations



An equivalence relation over $\{1, 2, 4, 7, 8, 9\}$

$\{(1,1), (2,2), (4,4), (2,4), (4,2), (7,7), (8,8), (9,9), (7,8), (8,7), (8,9), (9,8), (9,7), (7,9)\}$

with three equivalence classes: $\{1\}$, $\{2, 4\}$, and $\{7, 8, 9\}$

slide 130

Pre-Orders

Reflexive transitive relations are known as *pre-orders* .

Suppose $(\leq) \subseteq A \times A$ is a pre-order over A.
By the definition, $a \leq a$, and if $a_1 \leq a_2 \leq a_3$ then $a_1 \leq a_3$.
But we can have $a_1 \leq a_2 \leq a_1$ for $a_1 \neq a_2$.

(Note that we drew pairs (a_1, a_2) as $a_1 \longrightarrow a_2$, but write $(a_1, a_2) \in \leq$ or $a_1 \leq a_2$)

slide 131

Partial Orders

A *partial order* $\leq$ over A is a reflexive transitive relation (so a pre-order) that is also *antisymmetric*,

$\forall\, a_1, a_2 \in A.(a_1 \leq a_2 \wedge a_2 \leq a_1) \Rightarrow (a_1 = a_2)$

For example, here's part of the $\subseteq$ relation over sets:

$$\begin{tikzcd}
& \{1,2,3\} & \\
\{1,2\} \arrow[ur] & & \{1,3\} \arrow[ul] \\
& \{1\} \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

(when we draw a partial order, we usually omit the refl and tran edges — these are *Hasse diagrams*)

slide 132

Total Orders

A *total order* (or *linear order*) $\leq$ over A is a reflexive, transitive, antisymmetric relation (so a partial order) that is also *total*,

$\forall\, a_1, a_2 \in A.(a_1 \leq a_2 \vee a_2 \leq a_1)$

(in fact the reflexivity condition is redundant)

For example, here's a Hasse diagram of part of the usual $\leq$ relation over $\mathbb{N}$:

$$\begin{tikzcd}
2 \\
1 \arrow[u] \\
0 \arrow[u]
\end{tikzcd}$$

slide 133

Special Relations — Summary

A relation $R \subseteq A \times A$ is a directed graph. Properties:

- transitive $\forall\, a_1, a_2, a_3 \in A.(a_1\ R\ a_2 \wedge a_2\ R\ a_3) \Rightarrow a_1\ R\ a_3$
- reflexive $\forall\, a \in A.(a\ R\ a)$
- symmetric $\forall\, a_1, a_2 \in A.(a_1\ R\ a_2 \Rightarrow a_2\ R\ a_1)$
- acyclic $\forall\, a \in A.\neg(a\ R^+ a)$
- antisymmetric $\forall\, a_1, a_2 \in A.(a_1\ R\ a_2 \wedge a_2\ R\ a_1) \Rightarrow a_1 = a_2$
- total $\forall\, a_1, a_2 \in A.(a_1\ R\ a_2 \vee a_2\ R\ a_1)$

Combinations of properties: R is a ...

- directed acyclic graph if the transitive closure is acyclic
- undirected graph if symmetric
- equivalence relation if reflexive, transitive, and symmetric
- pre-order if reflexive and transitive,
- partial order if reflexive, transitive, and antisymmetric
- total order if reflexive, transitive, antisymmetric, and total

slide 134

Functions

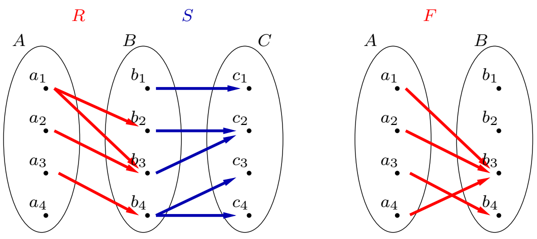
A *function* from A to B is just a relation which identifies exactly one element of B for each element of A.

$R \subseteq A \times B$ is defined to be *functional* when

$\forall\ a \in A.\exists\ b \in B.(a, b) \in R$ and

$\forall\ a \in A.\forall\ b, b' \in B.((a, b) \in R \wedge (a, b') \in R) \Rightarrow b = b'$

slide 135



Application — Relaxed Memory: One Intel/AMD Example

Initial shared memory values: $x = 0 \qquad y = 0$

Per-processor registers: $r_A \qquad r_B$

Processor A	Processor B
`store` $x := 1$	`store` $y := 1$
`load` $r_A := y$	`load` $r_B := x$

Processor A	Processor B
MOV [x]←$1	MOV [y]←$1
MOV EAX←[y]	MOV EBX←[x]

Final register values: $r_A =?$ $\qquad r_B =?$

slide 136

Application — Relaxed Memory: One Intel/AMD Example

Initial shared memory values: $x = 0 \qquad y = 0$

Per-processor registers: $r_A \qquad r_B$

Processor A	Processor B
`store` $x := 1$	`store` $y := 1$
`load` $r_A := y$	`load` $r_B := x$

Processor A	Processor B
MOV [x]←$1	MOV [y]←$1
MOV EAX←[y]	MOV EBX←[x]

slide 137

Final register values: $r_A =?$ $\qquad r_B =?$

Each processor can do its own store action before the store of the other processor.

Makes it hard to understand what your programs are doing!

Already a real problem for OS, compiler, and library authors.

Application — Relaxed Memory: part of the formalisation

$$
\begin{array}{l}
\text{preserved\_program\_order } \mathrm{E} = \\
\quad \{(e_1, e_2) \mid (e_1, e_2) \in (\text{po\_strict } \mathrm{E}) \wedge \\
\quad\quad ((\exists p\ r.(\text{loc } e_1 = \text{loc } e_2) \wedge \\
\quad\quad\quad\quad (\text{loc } e_1 = \mathsf{Some}\ (\mathsf{Location\_reg}\ p\ r))) \vee \\
\quad\quad (\text{mem\_load } e_1 \wedge \text{mem\_load } e_2) \vee \\
\quad\quad (\text{mem\_store } e_1 \wedge \text{mem\_store } e_2) \vee \\
\quad\quad (\text{mem\_load } e_1 \wedge \text{mem\_store } e_2) \vee \\
\quad\quad (\text{mem\_store } e_1 \wedge \text{mem\_load } e_2 \wedge (\text{loc } e_1 = \text{loc } e_2)) \vee \\
\quad\quad ((\text{mem\_load } e_1 \vee \text{mem\_store } e_1) \wedge \text{locked } \mathrm{E}\ e_2) \vee \\
\quad\quad (\text{locked } \mathrm{E}\ e_1 \wedge (\text{mem\_load } e_2 \vee \text{mem\_store } e_2)))\}
\end{array}
$$

slide 138

6 Induction

Induction

slide 139

In this section we will discuss different forms of proof by induction, both for natural numbers and for lists.

Example

Theorem $\sum_{i=1}^{n} i = n * (n+1)/2$

Proof By induction on n.

Base case (0): $\sum_{i=1}^{0} i = 0 = 0 * 1/2$

Inductive case ($n+1$): Assume $\sum_{i=1}^{n} i = n * (n+1)/2$ as the inductive hypothesis, then we have to prove
$\sum_{i=1}^{n+1} i = (n+1) * ((n+1)+1)/2.$
But $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = n * (n+1)/2 + (n+1) =$
$(n+1) * (n+1+1)/2$ $\square$

slide 140

What's really going on?

Using a fact about $\mathbb{N}$, the *induction principle*

$(P(0) \wedge (\forall\ n.P(n) \Rightarrow P(n+1))) \Rightarrow \forall\ n.P(n)$

(really a schema — that's true for any predicate P)

We think of an induction hypothesis, here taking

$P(n) \stackrel{\text{def}}{=} \sum_{i=1}^{n} i = n * (n+1)/2$

and instantiate the schema with it:

$$\begin{array}{l} (\ \ (\sum_{i=1}^{0} i = 0 * (0+1)/2) \wedge \\ \quad (\forall\ n.(\sum_{i=1}^{n} i = n * (n+1)/2) \\ \qquad\quad \Rightarrow \\ \qquad\quad (\sum_{i=1}^{n+1} i = (n+1) * ((n+1)+1)/2))) \\ \Rightarrow \\ \forall\ n. \sum_{i=1}^{n} i = n * (n+1)/2 \end{array}$$

slide 141

$$\begin{array}{l} (\ \ (\sum_{i=1}^{0} i = 0 * (0+1)/2) \wedge \\ \quad (\forall\ n.(\sum_{i=1}^{n} i = n * (n+1)/2) \\ \qquad\quad \Rightarrow \\ \qquad\quad (\sum_{i=1}^{n+1} i = (n+1) * ((n+1)+1)/2))) \\ \Rightarrow \\ \forall\ n. \sum_{i=1}^{n} i = n * (n+1)/2 \end{array}$$

Then we prove the antecedents of the top-level implication (with our normal proof techniques), and use modus ponens to conclude the consequent.

slide 142

Induction on lists

An ML function to append two lists:

```
fun app ([], ys)     = ys
  | app (x::xs, ys) = x :: app(xs,ys)
```

This is *terminating* and *pure* (no mutable state, no IO, no exceptions). So we can regard it as a mathematical function $\texttt{app}$.

It operates on *lists*. Suppose they are lists of elements of a set A.

Is $\texttt{app}$ associative?

slide 143

Induction on lists

Theorem

$\forall\ xs, ys, zs.\texttt{app}(\texttt{app}(xs, ys), zs) = \texttt{app}(xs, \texttt{app}(ys, zs))$

Proof We use the *induction schema for lists*

$(P([]) \wedge (\forall\ xs.P(xs) \Rightarrow \forall\ x.P(x :: xs))) \Rightarrow \forall\ xs.P(xs)$

with the induction hypothesis

$P(xs) \stackrel{\text{def}}{=}\ \forall\ ys, zs.\texttt{app}(\texttt{app}(xs, ys), zs) = \texttt{app}(xs, \texttt{app}(ys, zs))$

Base case: we have to prove $P([])$,
i.e. $\forall\ ys, zs.\texttt{app}(\texttt{app}([], ys), zs) = \texttt{app}([], \texttt{app}(ys, zs))$

a. Consider arbitrary ys and zs.
b. $\texttt{app}(\texttt{app}([], ys), zs) = \texttt{app}(ys, zs)$ by the first clause of the defn of $\texttt{app}$
c. ... $= \texttt{app}([], \texttt{app}(ys, zs))$ by the first clause of the defn of $\texttt{app}$ (backwards)

slide 144

Inductive step: we have to prove $(\forall\ xs.P(xs) \Rightarrow \forall\ x.P(x :: xs)))$

1. Consider an arbitrary xs.
2. Assume $P(xs)$
3. $\forall\ ys, zs.\texttt{app}(\texttt{app}(xs, ys), zs) = \texttt{app}(xs, \texttt{app}(ys, zs))$ from 2, unfolding defn of P
4. Consider an arbitrary x
   (now we have to prove $P(x :: xs)$, i.e.
   $\forall\ ys, zs.\texttt{app}(\texttt{app}(x :: xs, ys), zs) = \texttt{app}(x :: xs, \texttt{app}(ys, zs)))$
5. Consider arbitrary ys and zs
6. $\texttt{app}(\texttt{app}(x :: xs, ys), zs) = \texttt{app}(x :: \texttt{app}(xs, ys), zs)$ by the second clause of $\texttt{app}$
7. ... $= x :: \texttt{app}(\texttt{app}(xs, ys), zs)$ by the second clause of $\texttt{app}$
8. ... $= x :: \texttt{app}(xs, \texttt{app}(ys, zs))$ instantiating 3 with $ys = ys$,$zs = zs$ under $x ::$ ...
9. ... $= \texttt{app}(x :: xs, \texttt{app}(ys, zs))$ by the second clause of $\texttt{app}$ (backwards)
10. $P(x :: xs)$ from 5–9, by $\forall$-introduction and folding the defn of P
11. $\forall\ x.P(x :: xs)$ from 4–10 by $\forall$-introduction
12. $P(xs) \Rightarrow \forall\ x.P(x :: xs)$ from 2–11 by $\Rightarrow$-introduction
13. $\forall\ xs.P(xs) \Rightarrow \forall\ x.P(x :: xs)$ from 1–12 by $\forall$-introduction

Now from the induction scheme, (c), and (13), we have $\forall xs.P(xs)$, which (unfolding the defn of P) is exactly the theorem statement.

slide 145

Simpler proof structure: first rearrange the quantifiers

$\forall\ xs, ys, zs.\texttt{app}(\texttt{app}(xs, ys), zs) = \texttt{app}(xs, \texttt{app}(ys, zs))$
iff
$\forall\ ys, zs.\forall\ xs.\texttt{app}(\texttt{app}(xs, ys), zs) = \texttt{app}(xs, \texttt{app}(ys, zs))$

Then consider arbitrary ys and zs, and inside that do induction on lists, with induction hypothesis

$P(xs) \stackrel{\text{def}}{=} \texttt{app}(\texttt{app}(xs, ys), zs) = \texttt{app}(xs, \texttt{app}(ys, zs))$

(instead of $P(xs) \stackrel{\text{def}}{=} \forall\ ys, zs.\texttt{app}(\texttt{app}(xs, ys), zs) = \texttt{app}(xs, \texttt{app}(ys, zs)))$)

OK, as we don't need to instantiate P at different ys and zs

slide 146

Generalizing an Induction Hypothesis

ML functions for the length of a list:

```
fun nlength []       = 0
  | nlength (x::xs) = 1 + nlength xs
fun addlen (k,[])     = k
  | addlen (k,x::xs) = addlen(k+1,xs)
```

(compiler optimization?) Both are terminating and pure.

Theorem ? $\texttt{addlen}(0, \ell) = \texttt{nlength}(\ell)$

Induction on ℓ — but which induction hypothesis?

$P''(\ell) \stackrel{\text{def}}{=} \texttt{addlen}(0, \ell) = \texttt{nlength}(\ell)$ too weak

$P'(\ell) \stackrel{\text{def}}{=} \texttt{addlen}(k, \ell) = k + \texttt{nlength}(\ell)$ too rigid: need to vary k

$P(\ell) \stackrel{\text{def}}{=} \forall\ k.\texttt{addlen}(k, \ell) = k + \texttt{nlength}(\ell)$ just right

slide 147

Base case: we need to show $P([])$, i.e. $\forall\, k.\mathtt{addlen}(k, []) = k + \mathtt{nlength}([])$
1. Consider an arbitrary k.
2. $\mathtt{addlen}(k, []) = k = k + 0 = k + \mathtt{nlength}(0)$ by the defn addlen and nlength

Inductive step: we need to show $(\forall\, \ell.P(\ell) \Rightarrow \forall\, x.P(x :: \ell)))$
3. Consider an arbitrary ℓ
4. Assume the induction hypothesis $P(\ell)$, i.e. $\forall\, k.\mathtt{addlen}(k, \ell) = k + \mathtt{nlength}(\ell)$
5. Consider an arbitrary x
(now we have to show $P(x :: \ell)$, i.e. $\forall\, k.\mathtt{addlen}(k, x :: \ell) = k + \mathtt{nlength}(x :: \ell)$)
6. Consider an arbitrary k
7. $\mathtt{addlen}(k, x :: \ell) = \mathtt{addlen}(k + 1, \ell)$ by defn addlen
8. ... $= (k + 1) + \mathtt{nlength}(\ell)$ by 4, instantiating k with $k + 1$
9. ... $= k + \mathtt{nlength}(x :: \ell)$ by defn nlength
11. $\forall\, k.\mathtt{addlen}(k, x :: \ell) = k + \mathtt{nlength}(x :: \ell)$ from 6–9 by $\forall$-introduction
12. $P(x :: \ell)$ from 11 by folding defn P
13. $\forall\, x.P(x :: \ell)$ from 5–12 by $\forall$-introduction
14. $P(\ell) \Rightarrow \forall\, x.P(x :: \ell)$ from 4–13 by $\Rightarrow$-introduction
15. $\forall\, \ell.P(\ell) \Rightarrow \forall\, x.P(x :: \ell)$ from 3–14 by $\forall$-introduction
The theorem follows by instantiating P with $k = 0$ $\square$

slide 148

...rewriting that semi-structured proof more idiomatically:

Theorem $\mathtt{addlen}(0, \ell) = \mathtt{nlength}(\ell)$
Proof Induction on ℓ, with I.H. $P(\ell) \stackrel{\text{def}}{=} \forall\, k.\mathtt{addlen}(k, \ell) = k + \mathtt{nlength}(\ell)$
in induction schema $(P([]) \wedge (\forall\, xs.P(xs) \Rightarrow \forall\, x.P(x :: xs))) \Rightarrow \forall\, xs.P(xs)$

Base case: we need to show $P([])$
Consider an arbitrary k, then $\mathtt{addlen}(k, []) = k = k + 0 = k + \mathtt{nlength}(0)$ by defn addlen and nlength

Inductive step: consider an arbitrary ℓ, assume $P(\ell)$, and consider an arbitrary x. We have to show $P(x :: \ell)$.

Consider an arbitrary k.
$\mathtt{addlen}(k, x :: \ell) = \mathtt{addlen}(k + 1, \ell)$ by defn addlen
... $= (k + 1) + \mathtt{nlength}(\ell)$ by $P(\ell)$, instantiating k with $k + 1$
... $= k + \mathtt{nlength}(x :: \ell)$ by defn nlength

slide 149

7 Conclusion

Conclusion

We've introduced a good part of the language of discrete mathematics

(vocabulary, grammar, pragmatics...)

Fluency comes with use; you'll see that this is a remarkably flexible tool for formulating and analysing computational problems.

slide 150

The End

slide 151

A Exercises

Exercise Sheet 1: Propositional Logic

1. Let p stand for the proposition "I bought a lottery ticket" and q for "I won the jackpot". Express the following as natural English sentences:
   (a) $\neg p$
   (b) $p \vee q$
   (c) $p \wedge q$
   (d) $p \Rightarrow q$
   (e) $\neg p \Rightarrow \neg q$
   (f) $\neg p \vee (p \wedge q)$
2. Formalise the following in terms of atomic propositions r, b, and w, first making clear how they correspond to the English text.
   (a) Berries are ripe along the path, but rabbits have not been seen in the area.
   (b) Rabbits have not been seen in the area, and walking on the path is safe, but berries are ripe along the path.
   (c) If berries are ripe along the path, then walking is safe if and only if rabbits have not been seen in the area.
   (d) It is not safe to walk along the path, but rabbits have not been seen in the area and the berries along the path are ripe.
   (e) For walking on the path to be safe, it is necessary but not sufficient that berries not be ripe along the path and for rabbits not to have been seen in the area.
   (f) Walking is not safe on the path whenever rabbits have been seen in the area and berries are ripe along the path.
3. Formalise these statements and determine (with truth tables or otherwise) whether they are consistent (i.e. if there are some assumptions on the atomic propositions that make it true): "The system is in a multiuser state if and only if it is operating normally. If the system is operating normally, the kernel is functioning. Either the kernel is not functioning or the system is in interrupt mode. If the system is not in multiuser state, then it is in interrupt mode. The system is not in interrupt mode."
4. When is a propositional formula P *valid*? When is P *satisfiable*?
5. For each of the following propositions, construct a truth table and state whether the proposition is valid or satisfiable. (For brevity, you can just write one truth table with many columns.)
   (a) $p \wedge \neg p$
   (b) $p \vee \neg p$
   (c) $(p \vee \neg q) \Rightarrow q$
   (d) $(p \vee q) \Rightarrow (p \wedge q)$
   (e) $(p \Rightarrow q) \Leftrightarrow (\neg q \Rightarrow \neg p)$
   (f) $(p \Rightarrow q) \Rightarrow (q \Rightarrow p)$
6. For each of the following propositions, construct a truth table and state whether the proposition is valid or satisfiable.

(a) $\mathrm{p} \Rightarrow (\neg \mathrm{q} \vee \mathrm{r})$

(b) $\neg \mathrm{p} \Rightarrow (\mathrm{q} \Rightarrow \mathrm{r})$

(c) $(\mathrm{p} \Rightarrow \mathrm{q}) \vee (\neg \mathrm{p} \Rightarrow \mathrm{r})$

(d) $(\mathrm{p} \Rightarrow \mathrm{q}) \wedge (\neg \mathrm{p} \Rightarrow \mathrm{r})$

(e) $(\mathrm{p} \Leftrightarrow \mathrm{q}) \vee (\neg \mathrm{q} \Leftrightarrow \mathrm{r})$

(f) $(\neg \mathrm{p} \Leftrightarrow \neg \mathrm{q}) \Leftrightarrow (\mathrm{q} \Leftrightarrow \mathrm{r})$

7. Formalise the following and, by writing truth tables for the premises and conclusion, determine whether the arguments are valid.

   (a) Either John isn't stupid and he is lazy, or he's stupid.
   John is stupid.
   Therefore, John isn't lazy.

   (b) The butler and the cook are not both innocent
   Either the butler is lying or the cook is innocent
   Therefore, the butler is either lying or guilty

8. Use truth tables to determine which of the following are equivalent to each other:

   (a) P

   (b) $\neg P$

   (c) $P \Rightarrow F$

   (d) $P \Rightarrow T$

   (e) $F \Rightarrow P$

   (f) $T \Rightarrow P$

   (g) $\neg\neg P$

9. Use truth tables to determine which of the following are equivalent to each other:

   (a) $(P \wedge Q) \vee (\neg P \wedge \neg Q)$

   (b) $\neg P \vee Q$

   (c) $(P \vee \neg Q) \wedge (Q \vee \neg P)$

   (d) $\neg(P \vee Q)$

   (e) $(Q \wedge P) \vee \neg P$

10. Imagine that a logician puts four cards on the table in front of you. Each card has a number on one side and a letter on the other. On the uppermost faces, you can see E, K, 4, and 7. He claims that if a card has a vowel on one side, then it has an even number on the other. How many cards do you have to turn over to check this?

11. Give a truth-table definition of the ternary boolean operation **if** P **then** Q **else** R.

12. Given the truth table for an arbitrary n-ary function $f(\mathrm{p}_1, .., \mathrm{p}_n)$ (from n atomic propositions $\mathrm{p}_1, .., \mathrm{p}_n$ to $\{T, F\}$), describe how one can build a proposition, using only $\mathrm{p}_1, .., \mathrm{p}_n$ and the connectives $\wedge$, $\vee$, and $\neg$, that has the same truth table as f. (Hint: first consider each *line* of the truth table separately, and then how to combine them.)

13. Show, by equational reasoning from the axioms in the notes, that $\neg(P \wedge (Q \vee R \vee S))$ iff $\neg P \vee (\neg Q \wedge \neg R \wedge \neg S)$

Exercise Sheet 2: Predicate Logic

1. Formalise the following statements in predicate logic, making clear what your atomic predicate symbols stand for and what the domains of any variables are.
   (a) Anyone who has forgiven at least one person is a saint.
   (b) Nobody in the calculus class is smarter than everybody in the discrete maths class.
   (c) Anyone who has bought a Rolls Royce with cash must have a rich uncle.
   (d) If anyone in the college has the measles, then everyone who has a friend in the college will have to be quarantined.
   (e) Everyone likes Mary, except Mary herself.
   (f) Jane saw a bear, and Roger saw one too.
   (g) Jane saw a bear, and Roger saw it too.
   (h) If anyone can do it, Jones can.
   (i) If Jones can do it, anyone can.
2. Translate the following into idiomatic English.
   (a) $\forall\, x.(\mathrm{H}(x) \wedge \forall\, y.\neg\mathrm{M}(x, y)) \Rightarrow \mathrm{U}(x)$ where $\mathrm{H}(x)$ means x is a man, $\mathrm{M}(x, y)$ means x is married to y, $\mathrm{U}(x)$ means x is unhappy, and x and y range over people.
   (b) $\exists\, z.\mathrm{P}(z, x) \wedge \mathrm{S}(z, y) \wedge \mathrm{W}(y)$ where $\mathrm{P}(z, x)$ means z is a parent of x, $\mathrm{S}(z, y)$ means z and y are siblings, $\mathrm{W}(y)$ means y is a woman, and x, y, and z range over people.
3. State whether the following are true or false, where x, y and z range over the integers.
   (a) $\forall\, x.\exists\, y.(2x - y = 0)$
   (b) $\exists\, y.\forall\, x.(2x - y = 0)$
   (c) $\forall\, x.\exists\, y.(x - 2y = 0)$
   (d) $\forall\, x.x < 10 \Rightarrow \forall\, y.(y < x \Rightarrow y < 9)$
   (e) $\exists\, y.\exists\, z.y + z = 100$
   (f) $\forall\, x.\exists\, y.(y > x \wedge \exists\, z.y + z = 100)$
4. What changes above if x, y and z range over the reals?
5. Formalise the following (over the real numbers):
   (a) Negative numbers don't have square roots
   (b) Every positive number has exactly two square roots

Exercise Sheet 3: Structured Proof

1. Give structured proofs of
   (a) $(P \Rightarrow Q) \Rightarrow ((Q \Rightarrow R) \Rightarrow (P \Rightarrow R))$
   (b) $(P \Rightarrow Q) \Rightarrow ((R \Rightarrow \neg Q) \Rightarrow (P \Rightarrow \neg R))$
   (c) $(P \Rightarrow (Q \Rightarrow R)) \Rightarrow (\neg R \Rightarrow (P \Rightarrow \neg Q))$

   (For more practice with structured proofs, try proving some of the standard logical equivalences.)
2. Consider the following non-Theorem. What's wrong with the claimed proof?

   Non-Theorem Suppose x and y are reals, and $x + y = 10$. Then $x \neq 3$ and $y \neq 8$.

   Proof Suppose the conclusion of the Theorem is false. Then $x = 3$ and $y = 8$. But then $x + y = 11$, which contradicts the assumption that $x + y = 10$. Hence the conclusion must be true.
3. Give a structured proof of $((\forall\, x.\mathrm{L}(x) \Rightarrow \mathrm{F}(x)) \wedge (\exists\, x.\mathrm{L}(x) \wedge \neg \mathrm{C}(x))) \Rightarrow \exists\, x.\mathrm{F}(x) \wedge \neg \mathrm{C}(x)$
4. Give a structured proof of $(\exists\, x.(P(x) \Rightarrow Q(x))) \Rightarrow ((\forall\, x.P(x)) \Rightarrow \exists\, x.Q(x))$
5. Prove that, for any $n \in \mathbb{N}$, n is even iff n^3 is even (hint: first define what 'even' means).
6. Prove that the following are equivalent:
   (a) $\exists\, x.P(x) \wedge \forall\, y.(P(y) \Rightarrow y = x)$
   (b) $\exists\, x.\forall\, y.P(y) \Leftrightarrow y = x$

Exercise Sheet 4: Sets

1. Consider the set $A \stackrel{\text{def}}{=} \{\{\}, \{\{\}\}, \{\{\{\}\}\}\}$. If $x \in A$, how many elements might x have?
2. Prove that if $A \subseteq B$ then $A \cup B = B$
3. Prove that if $A \subseteq A'$ and $B \subseteq B'$ then $A \times B \subseteq A' \times B'$
4. What can you say about sets A and B if you know that
   (a) $A \cup B = A$
   (b) $A \cap B = A$
   (c) $A - B = A$
   (d) $A \cap B = B \cap A$
   (e) $A - B = B - A$
5. Draw the Hasse diagram for the subset relation $\subseteq$ among the sets $A \stackrel{\text{def}}{=} \{2, 4, 6\}$, $B \stackrel{\text{def}}{=} \{2, 6\}$, $C \stackrel{\text{def}}{=} \{4, 6\}$, and $D \stackrel{\text{def}}{=} \{4, 6, 8\}$.
6. Is $\mathcal{P}(A \cap B) = \mathcal{P}(A) \cap \mathcal{P}(B)$ true for all sets A and B? Either prove it or give a counterexample.
7. Is $\mathcal{P}(A \cup B) = \mathcal{P}(A) \cup \mathcal{P}(B)$ true for all sets A and B? Either prove it or give a counterexample.
8. Draw pictures illustrating the following subsets of $\mathbb{R}^2$.
   (a) $\{(x, y) \mid y = x^2 - x - 2\}$
   (b) $\{(x, y) \mid y < x\}$

(c) $\{(x, y) \mid (y > 0 \wedge y = x)\} \cup \{(2, y) \mid y > 1\} \cup \{(0, 0)\}$

9. Let S be a set of students, R a set of college rooms, P a set of professors, and C a set of courses. Let $L \subseteq S \times R$ be the relation containing (s, r) if student s lives in room r. Let $E \subseteq S \times C$ be the relation containing (s, c) if student s is enrolled for course c. Let $T \subseteq C \times P$ be the relation containing (c, p) if course c is lectured by professor p. Describe the following relations.

   (a) E^{-1}

   (b) $L^{-1}; E$

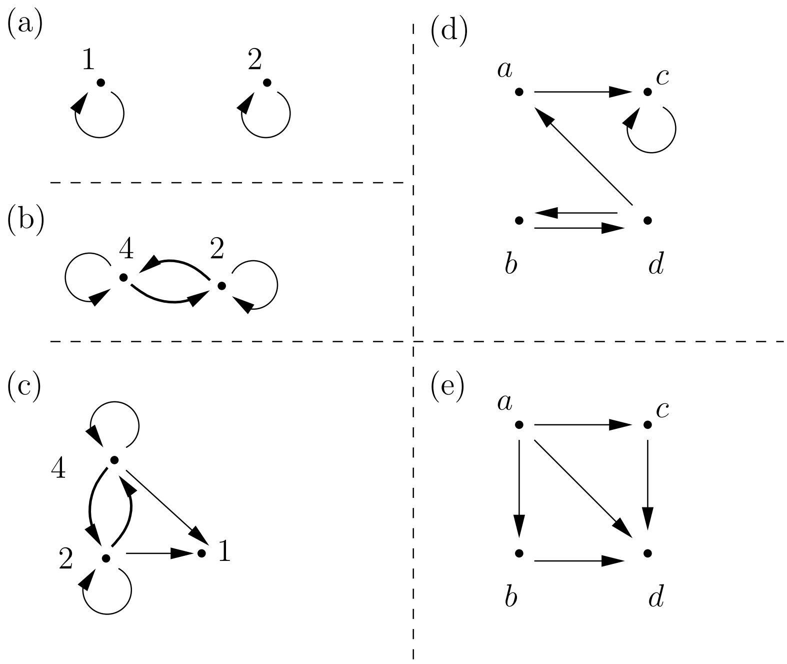
   (c) $E; E^{-1}$

   (d) $(L^{-1}; E); T$

   (e) $L^{-1}; (E; T)$

   (f) $(L^{-1}; L)^{+}$

10. For each of the following 5 relations, list its ordered pairs. Give a table showing for each whether it is reflexive, symmetric, transitive, acyclic, antisymmetric, and/or total.



11. Give a table showing, for each of the following relations over $\mathbb{N}$, whether it is reflexive, symmetric, transitive, or functional.

   (a) $n\ R\ m \stackrel{\text{def}}{=} n = 2m$

   (b) $n\ R\ m \stackrel{\text{def}}{=} 2n = m$

   (c) $n\ R\ m \stackrel{\text{def}}{=} \exists\, k.k \geq 2 \wedge k \text{ divides } n \wedge k \text{ divides } m$

12. (a) If R and S are directed acyclic graphs over a set A, is $R; S$? Either prove it or give a counterexample.

   (b) If R and S are directed acyclic graphs over a set A, is $R \cup S$? Either prove it or give a counterexample.

   (c) If R and S are directed acyclic graphs over a set A, is $R \cap S$? Either prove it or give a counterexample.

(d) If R is a relation over a set A, can it be both symmetric and antisymmetric? Either give an example or prove it cannot.

Exercise Sheet 5: Inductive Proof

In all of the following, please state your induction hypothesis explicitly as a predicate.

1. Prove that, for all natural numbers n, $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$.
2. Prove that, for all natural numbers x, m, and n, $x^{m+n} = x^m\, x^n$.
3. Prove that for all $n \geq 3$, if n distinct points on a circle are joined by consecutive order by straight lines, then the interior angles of the resulting polygon add up to $180(n-2)$ degrees.
4. Prove that, for any positive integer n, a $2^n \times 2^n$ square grid with any one square removed can be tiled with L-shaped pieces consisting of 3 squares.
5. Consider the following pair of ML function declarations:

   ```
   fun takew p [] = []
     | takew p (x::xs) = if p x then x ::  takew p xs else []
   fun dropw p [] = []
     | dropw p (x::xs) = if p x then dropw p xs else x::xs
   ```

   Prove `(takew p xs) @ (dropw p xs) = xs` using induction. (Assume that function `p` always terminates.) [Software Engineering II, 2001, p.2, q.9b]
6. Consider the following two ML functions:

   ```
   fun sumfiv [] = 0
     | sumfiv (x::xs) = 5*x + sumfiv xs
   fun summing z [] = z
     | summing z (x::xs) = summing (z + x) xs
   ```

   Prove that `sumfiv xs` is equal to `5 * summing 0 xs`. [Software Engineering II, 1999, p.2, q.9c]