Discrete Mathematics

Exercise Sheet 1

Alistair O'Brien

November 6, 2022

Before Attempting the Problems

The primary goal of this course is to introduce to you *formal reasoning* and various (discrete) mathematical structures such as *sets, relations, and functions* alongside some niche number and automata theory. It is vital that you master many of these topics for various other Tripos courses[1].

Please complete these exercises and submit your solutions 48 hours before our scheduled supervisions to `ajo41@cam.ac.uk` as a **PDF** attachment. Feel free to choose the format of your answers (handwritten, typed, LaTeX'd) – just ensure that they're (mostly) legible[2].

Attempt to complete at least 75% of the set exercises – if you get stuck simply make a note of it in your answer and move on to the next question. We will be able to discuss all the solutions during the supervisions.

These supervisions will focus on both examinable and *non-examinable material*, the latter being for pedagogical reasons. Each week I will provide a 'Material' section with notes/summaries of the lecture material. You are welcome to skip these sections, they are simply there for your benefit. However, some questions will (explicitly) refer to this material.

Some of the exercises are taken from Marcelo's exercise sheets which you can find here: https://www.cl.cam.ac.uk/teaching/2223/DiscMath/materials.html.

[1] In particular, Algorithms (IA), Semantics of Programming Languages (IB), Logic and Proof (IB), Computation Theory (IB), Complexity Theory (IB), Formal Models of Language (IB), Denotational Semantics (II), Types (II), Hoare Logic and Model Checking (II), and Category Theory (II)

[2] Don't worry too much about this though, I can't handwrite anything legible either

1 Material

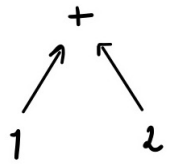
1.1 Propositional and Higher-Order Logic

Propositional logic deals with truth values and logical connectives *and, or, not*, etc[3]. Most of the concepts of propositional logic have counter-parts in other logics making it a useful starting point for learning logic.

Here are the most fundamental concepts of any logic:

Syntax refers to the *notation* for writing logical statements.

For example, syntactically 1 + 2 denotes an *expression* with the binary operator + and applied to the child expressions 1 and 2.



Syntax is often defined using a *grammar* – a set of *recursive rules* that describe how we can construct statements. For example, the grammar of *expressions* e (using BNF notation[4]) could be:

$$e ::= n \mid e + e \mid e \times e \mid e - e \mid e/e \mid (e)$$

We read this as follows: "an expression e could be a number n, or an addition $e_1 + e_2$, or a multiplication $e_1 \times e_2$, etc"

Semantics expresses the *meaning* of logical statements. For example, while 1 + 2 and 2 + 1 are syntactically distinct, they are semantically equivalent to 3. The semantics for logics are often boolean values (0 and 1).

Proof Theory concerns the methods of proving logical statements. This course doesn't discuss *formal proof theory* (this is taught in Logic and Proof, IB), but you should be able to write *formal proofs* (Section 1.2).

You may think of a logic as a *programming language* with syntax (how we write the program) and semantics (what the program means). Proofs are a means to reason about how these programs run (without running them).

For example, the statement: for integer n, n^2 is even if and only if n is even cannot be *semantically* proven since it requires checking the statement *for all* integers n (which there are infinitely many).

[3]Propositional logic is essentially the Boolean Algebra taught in Digitial Electronics
[4]Backus Naur-form

Propositional Logic

Syntax A *propositional symbol*, denoted $a, b, c, \ldots$, is a symbol that refers to some logical statement, such as "it is raining", analogous to representing a number by a letter (e.g. $n = 1$).

Propositional statements P are defined by the grammar:

$$\begin{aligned} P, Q ::= {} & a \mid \textbf{true} \mid \textbf{false} \\ & \mid \neg P && \text{(not)} \\ & \mid P \wedge Q && \text{(and)} \\ & \mid P \vee Q && \text{(or)} \\ & \mid P \implies Q && \text{(implies)} \\ & \mid P \iff Q && \text{(if and only if)} \end{aligned}$$

The logical connectives are listed in order of *precedence*; $\neg$ is the highest. Precedence is used to suppress needless parentheses, writing, for example:

$$((\neg a) \wedge b) \vee c \implies ((\neg a) \vee b) \qquad \text{as} \qquad \neg a \wedge b \vee c \implies \neg a \vee b$$

Similar to how $(1 + (2 \times (3 + 4)))$ is written as $1 + 2 \times (3 + 4)$.

Semantics Each statement P propositional logic has a logical meaning – either 0 or 1 – relative to the assigned logical meaning of the *propositional symbols* (e.g. assignment $a = 0, b = 1, c = 1$ satisfies the above example).

The meaning is defined by standard truth tables (like those from Digital Electronics)

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \implies Q$	$P \iff Q$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

NB: **true** and **false** are propositional statements that denote the values 0 and 1. The former is *syntactic*, whereas the latter belongs to the *semantics*.

Higher-Order Logic

Higher-order logic extends propositional logic to allow us to reason about elements in sets. We use *quantifiers*: $\forall$ (for all) and $\exists$ (there exists), where variables range over the given set of the quantifier (e.g. $\forall p \in \mathsf{People}.\ p$ is a philospher $\implies p$ is a scholar).

Typically x, y, z denote variables ranging over elements (e.g. integers), f, g, h denote functions, A, B, C denote sets, and R, S, T denote predicates (Lecture Notes, p24).

Syntax The syntax of e and logical statements P are defined by the grammars:

$$
\begin{aligned}
P, Q ::= {} & \textbf{true} \mid \textbf{false} \\
\mid {} & R(e) && \text{(Predicate)} \\
\mid {} & \neg P && \text{(not)} \\
\mid {} & P \wedge Q && \text{(and)} \\
\mid {} & P \vee Q && \text{(or)} \\
\mid {} & P \implies Q && \text{(implies)} \\
\mid {} & P \iff Q && \text{(if and only if)} \\
\mid {} & \forall x \in A.P && \text{(Universal Quantification)} \\
\mid {} & \exists x \in A.P && \text{(Existential Quantification)}
\end{aligned}
$$

$$
\begin{aligned}
e ::= {} & x && \text{(Variable Term)} \\
\mid {} & \ldots
\end{aligned}
$$

Expressions (or terms) are used to denote values, for example e^{1+2}, that occur in logical statements. **NB**: The definition for expressions isn't fully specified since (almost) anything can be an expression, thus the grammar would be rather large.

As with propositional logic, connectives are listed in order of precedence, with $\forall$ and $\exists$ binding 'more weakly' than any connective:

$$
\begin{aligned}
& \forall m, n \in A.R(m) \vee R(n) \implies R(m \times n) \\
& \quad \text{is} \qquad (\forall m, n \in A.((R(m) \vee R(n)) \implies R(m \times n)))
\end{aligned}
$$

Nested quantifiers such as $\forall x \in A.\forall y \in A.P$ are abbreviated to $\forall x, y \in A.P$.

Here are some logical statements with rough English translations:

– All supervisors are brilliant:

$$\forall s \in \text{Supervisor}.\ s \text{ is brilliant}$$

– Every student has a supervisor:

$$\forall s \in \text{Student}.\exists q \in \text{Supervisor}.\ q \text{ supervsises } s$$

– Every student's tutor is a member of the student's college:

$$\forall s \in \text{Student}, c \in \text{College}.\ s \text{ is a member of } c \implies tutor(s) \text{ is a member of } c$$

Semantics The informal semantics of higher-order logic follow those of propositional logic, with the following meaning for quantifiers:

– $\forall x \in A.\ P(x)$: For every element c in the set A, the property $P(c)$ holds.

For finite set $A = \{c_1, \ldots, c_n\}$ this is equivalent to $P(c_1) \wedge P(c_2) \wedge \cdots \wedge P(c_n)$.

– $\exists x \in A.P(x)$: There exists an element c in the set A, the property $P(c)$ holds.

The element c is often referred to as a *witness*.

If you're interested in the *formal semantics* of higher-order logic we can discuss this during supervisions.

1.2 Fitch-Style Proofs

Fitch-style proofs is a concise notational system for writing *formal proofs* (not essays). It is essentially a a numbered list of statements, such that each statement is either:

(i) an *assumption*,

(ii) a statement that follows from a *proof rule* and prior statements (referenced by line number).

Introducing a new set of assumptions increases the level of indentation, beginning a new 'scope' – this conveys which assumptions are in scope for a given line in the proof. After the assumptions of the current scope have been introduced, a horizontal line is written.

For example, the following is a proof of $P \implies (Q \implies P)$:

1	P	
2	$\quad Q$	
3	$\quad P$	R, 1
4	$Q \implies P$	$\Rightarrow$I, 2–3
5	$P \implies (Q \implies P)$	$\Rightarrow$I, 1–4

This is read as:

1. The first subproof. We assume that P holds and wish to show that $Q \implies P$ holds.
2. The second subproof. We assume that Q holds and wish to show that P holds.
3. We have P from the assumptions in 1 using the *repitition* rule.
4. We conclude $Q \implies P$ using the $\implies$*-introduction* rule with assumptions starting at 2 and conclusion at 3.
5. We conclude $P \implies (Q \implies P)$ using the $\implies$*-introduction* rule with assumptions starting at 1 and conclusion at 4.

Proof Rules

Proof rules (or *proof patterns*) are a set of rules that can be systematically applied to prove statements.

Proof rules of logical connectives are split into either: *introduction* or *elimination* rules. For example, for implication connective $\implies$:

- Modus ponens rule (from Lectures): *if* $P \implies Q$ *and* P *holds, then* Q *holds* is an *elimination* rule since it **eliminates** the connective $\implies$ from the premises.
- Whereas, the proof rule: *assume* P*, if* Q *holds, then* $P \implies Q$ *holds* is an *introduction* rule since it **introduces** the connective $\implies$ in the conclusion.

A (non-exhaustive) list of proof rules for Fitch-Style proofs are as follows:

Repitition (R) The repetition rule allows us to repeat statements we've previously proved (or assumed).

m	P	
$\vdots$	$\cdots \mid \vdots$	
n	P	R, m

True Introduction ($\top$I) **true** always holds.

m	**true**	$\top$I

False Elimination ($\bot$E) If we have proven **false** is true – deducing a logical contradiction. We can then conclude any statement P holds. This rather non-sensical rule comes from the *Ex falso quodlibet* axiom (Google it).

m	**false**	
$\vdots$	$\vdots$	
n	P	$\bot$E, m

Negation Introduction ($\neg$I) If we assume P holds and conclude **false** (a logical contradiction), then we deduce that P cannot be true, hence $\neg P$ holds.

m	P	
$\vdots$	$\vdots$	
n	**false**	
$n+1$	$\neg P$	$\neg$I, m–n

Negation Elimination (¬E) If P and $\neg P$ holds, then we have a logical contradiction.

m	P	
$\vdots$	$\vdots$	
n	$\neg P$	
$\vdots$	$\vdots$	
p	**false**	$\neg$E, m, n

m	$\neg P$	
$\vdots$	$\vdots$	
n	P	
$\vdots$	$\vdots$	
p	**false**	$\neg$E, m, n

Conjunction Introduction ($\wedge$I) If P holds and Q holds, then we conclude that $P \wedge Q$ holds.

m	P	
$\vdots$	$\vdots$	
n	Q	
$\vdots$	$\vdots$	
p	$P \wedge Q$	$\wedge$I, m, n

m	Q	
$\vdots$	$\vdots$	
n	P	
$\vdots$	$\vdots$	
p	$P \wedge Q$	$\wedge$I, m, n

Conjunction Elimination ($\wedge$E) If $P \wedge Q$ holds, then P holds and Q holds.

m	$P \wedge Q$	
$\vdots$	$\vdots$	
n	P	$\wedge$E, m

m	$P \wedge Q$	
$\vdots$	$\vdots$	
n	Q	$\wedge$E, m

Proof by Contradiction (C) To prove P, assume P is false (that is, $\neg P$ is true) and deduce a contradiction.

m		$\neg P$	
$\vdots$		$\vdots$	
n		**false**	
$n+1$	P		C

Disjunction Introduction ($\vee$I) If we have proven either P or Q holds, then we have

proven $P \vee Q$.

m	P	
$\vdots$	$\vdots$	
n	$P \vee Q$	$\vee$I, m

m	Q	
$\vdots$	$\vdots$	
n	$P \vee Q$	$\vee$I, m

Disjunction Elimination ($\vee$E) If we have $P \vee Q$ and wish to show R, we consider each of the following cases in turn:

(i) Assume P to establish R

(ii) Assume Q to establish R

m	$P \vee Q$		
$\vdots$	$\vdots$		
n	| P		
$\vdots$	| $\vdots$		
p	| R		
q	| Q		
$\vdots$	| $\vdots$		
r	| R		
$r+1$	R		$\vee$E, m, n–p, q–r

Implication Introduction ($\Rightarrow$I) To prove $P \implies Q$, assume P holds and deduce Q.

m	| P		
$\vdots$	| $\vdots$		
n	| Q		
$n+1$	$P \implies Q$		$\Rightarrow$I, m–n

Implication Elimination ($\Rightarrow$E) If we have proven P and $P \implies Q$, the statement Q follows. This is known as *Modus Ponens*.

$$
\begin{array}{l|ll}
m & P & \\
\vdots & \vdots & \\
n & P \implies Q & \\
\vdots & \vdots & \\
p & Q & \Rightarrow\text{E}, m, n
\end{array}
\qquad
\begin{array}{l|ll}
m & P \implies Q & \\
\vdots & \vdots & \\
n & P & \\
\vdots & \vdots & \\
p & Q & \Rightarrow\text{E}, m, n
\end{array}
$$

Forall Introduction ($\forall$I) To prove $\forall x \in A.P(x)$, we introduce an arbitrary 'fresh' variable $y \in A$ in between the vertical lines on line m. This variable may only occur in lines m-n. We then show that $P(y)$ holds.

If x is unique in the proof then the 'fresh' variable is no-longer required and you may use x in place of y.

$$
\begin{array}{l|l|ll}
m & y & \cdots & \\
\vdots & & \vdots & \\
n & & P(y) & \\
n+1 & \forall x \in A.P(x) & & \forall\text{I}, m\text{–}n
\end{array}
$$

Forall Elimination ($\forall$E) If we have $\forall x \in A.P(x)$, we can substitute x in $P(x)$ with any value $e \in A$, concluding $P(e)$ is true.

$$
\begin{array}{l|ll}
m & \forall x \in A.P(x) & \\
\vdots & \vdots & \\
n & P(e) & \forall\text{E}, m
\end{array}
$$

Exists Introduction ($\exists$I) To prove $\exists x \in A.P(x)$, find a *witness* e such that $P(e)$ holds.

$$
\begin{array}{l|ll}
m & P(e) & e \in A \\
\vdots & \vdots & \\
n & \exists x \in A.P(x) & \exists\text{I}, m
\end{array}
$$

Exists Elimination ($\exists$E) To use an existing statement of the form $\exists x \in A.P(x)$ to prove R. First we introduce a 'fresh' variable w to stand for some value $w \in A$ assuming

$P(w)$ holds, we then prove R.

m	$\exists x \in A.P(x)$		
$\vdots$	$\vdots$		
n	w	$P(w)$	
$\vdots$		$\vdots$	
p		R	
$p+1$	R		$\exists$E, m, n–p

Examples

A proof of one of *de Morgan's law* (in one direction) $\neg(P \vee Q) \implies \neg P \wedge \neg Q$ is as follows:

1		$\neg(P \vee Q)$		
2			P	
3			$P \vee Q$	$\vee$I, 2
4			**false**	$\neg$E, 1, 3
5		$\neg P$		$\neg$I, 2–4
6			Q	
7			$P \vee Q$	$\vee$I, 6
8			**false**	$\neg$E, 1, 7
9		$\neg Q$		$\neg$I, 6–8
10		$\neg P \wedge \neg Q$		$\wedge$I, 5, 9
11	$\neg(P \vee Q) \implies \neg P \wedge \neg Q$			$\Rightarrow$I, 1–10

The next two examples use quantifiers:

1		$\forall x \in A.P(x,x)$		
2		u	$P(u,u)$	$\forall$E, 1
3			$\exists z \in A.P(u,z)$	$\exists$I, 2
4		$\forall y \in A.\exists z \in A.P(y,z)$		$\forall$I, 2–3
5	$\forall x \in A.P(x,x) \implies \forall y \in A.\exists z \in A.P(y,z)$			$\Rightarrow$I, 1–4

1	$\forall x \in A.P(x) \implies Q(x)$		
2	$\quad \exists y \in A.P(y)$		
3	$\quad w \mid P(w)$		
4	$\quad\quad P(w) \implies Q(w)$		$\forall$E, 1
5	$\quad\quad Q(w)$		$\implies$E, 3, 4
6	$\quad\quad \exists y \in A.Q(y)$		$\exists$I, 5
7	$\quad \exists y \in A.Q(y)$		$\exists$E, 2, 3–6
8	$\exists y \in A.P(y) \implies \exists y \in A.Q(y)$		$\implies$I, 2–7

Recall that Proposition 8 from Lectures is defined as:

$$\forall m, n \in \mathbb{Z}.\ m, n \text{ is odd} \implies m \times n \text{ is odd}$$

The Fitch-Style proof is as follows:

1	mn	m is odd	
2		n is odd	
3		$\exists k \in \mathbb{Z}.\ m = 2k + 1$	Def. of odd, 1
4		$\exists l \in \mathbb{Z}.\ n = 2l + 1$	Def. of odd, 2
5		$k_0 \mid m = 2k_0 + 1$	
6		$\quad l_0 \mid n = 2l_0 + 1$	
7		$\quad\quad m \times n = (2k_0 + 1) \times (2l_0 + 1)$	Congruence, 5, 6
8		$\quad\quad m \times n = 4k_0l_0 + 2k_0 + 2l_0 + 1$	Dist, 7
9		$\quad\quad m \times n = 2(k_0l_0 + k_0 + l_0) + 1$	Factor, 8
10		$\quad\quad \exists p \in \mathbb{Z}.\ m \times n = 2p + 1$	$\exists$I, 9
11		$\quad\quad m \times n$ is odd	Def. of odd, 10
12		$\quad m \times n$ is odd	$\exists$E, 4, 6–11
13		$m \times n$ is odd	$\exists$E, 2, 5–13
14		m, n is odd $\implies m \times n$ is odd	$\implies$I, 1–13
15	$\forall m, n \in \mathbb{Z}.\ m, n$ is odd $\implies m \times n$ is odd		$\forall$I, 1–14

2 Exercises

1. In OCaml, we can represent logical statements P from propositional logic (Section 1.1) using the following recursive datatype:

```
type prop =
  | True (** true *)
  | False (** false *)
  | Symbol of string (** a,b,c,... *)
  | Not of prop (** ¬P *)
  | And of prop * prop (** P ∧ Q *)
  | Or of prop * prop (** P ∨ Q *)
  | Implies of prop * prop (** P ⟹ Q *)
  | Iff of prop * prop (** P ⟺ Q *)
```

   For example, `And(Symbol "a", Symbol "b")` denotes the statement $a \wedge b$.

   (i) An *assignment* `assn` is an *associative list*, mapping propositional symbols $a, b, c, \ldots$ to truth values. In OCaml, we represent assignments with the type:

```
type assignment = (string * bool) list
```

   Write a function `is_satisfiable` which returns `true` if statement `p` has the semantic meaning 1 under the assignment `assn`. Your function should have the type:

```
val is_satisfiable : assignment -> prop -> bool
```

   (ii) A statement P is said to be *valid* (or a *tautology*) if it is satisfiable for all assignments:

$$\forall\, \texttt{assn} \in \texttt{assignment}.\ \texttt{satisfies assn}\ P = \texttt{true}$$

   Write a function `is_valid` which returns `true` if P is valid. Your function should have the type:

```
val is_valid : prop -> bool
```

   [*hint*: Split the problem into generating the set of all possible assignments for a given statement P, and then checking all those assignments satisfy P]

2. Following from exercise 1, we may extend our example to a restricted form of higher-order logic (Section 1.1) where variables only range over finite sets of integers:

```
type pred = int list -> bool
type 'a quant = int list * (int -> 'a)

type prop =
    | True (** true *)
    | False (** false *)
    | Pred of pred * int list (** R(n_1,...,n_m) *)
    | Not of prop (** ¬P *)
```

```
| And of prop * prop (** P ∧ Q *)
| Or of prop * prop (** P ∨ Q *)
| Implies of prop * prop (** P ⟹ Q *)
| Iff of prop * prop (** P ⟺ Q *)
| Forall of prop quant (** ∀x ∈ A. P *)
| Exists of prop quant (** ∃x ∈ A. P *)
```

(i) For the finite set of integers $S \subseteq \mathbb{Z}$, consider the statement:

$$\forall x, y \in S. \exists z \in S.\ x + z = y - z$$

Give an OCaml representation, using the above datatype `prop`, for this statement.

(ii) Write a function `is_valid` which returns `true` if the statement `p` has the semantic meaning 1. Your function should now have the type:

```
val is_valid : prop -> bool
```

3. (i) [*optional*] Give a Fitch-style elimination and introduction proof rules for the $\iff$ logical connective.

(ii) For predicate $P(x)$ and statement Q (not mentioning x), show that the following equivalence holds:

$$((\exists x \in A.\ P(x)) \implies Q) \iff (\forall x \in A.\ P(x) \implies Q)$$

(iii) For predicate $P(x)$, show the following:

$$\neg\exists x \in A.\ P(x) \iff \forall x \in A.\ \neg P(x)$$

(iv) For predicates $P(x)$ and $Q(x)$, show the following equivalence:

$$(\forall x \in A.\ P(x)) \wedge (\forall x \in A.\ Q(x)) \iff \forall x \in A.\ P(x) \wedge Q(x)$$

(v) For predicates $P(x)$ and $Q(x)$, show the following:

$$(\forall x \in A.\ P(x)) \vee (\forall x \in A.\ Q(x)) \implies \forall x \in A.\ P(x) \vee Q(x)$$

Is the converse statement true? If so, provide a proof, otherwise provide a counterexample.

[*remark*: It may be worth doing these questions using Fitch-style proofs (Section 1.2)]

4. Characterise the integers $d, n \in \mathbb{Z}$ such that

(i) $0 \mid n$

(ii) $d \mid 0$

5. Let $m, n \in \mathbb{Z}$ be arbitrary integers and $k \in \mathbb{Z}^+$ be positive. Show that:

$$(k \times m) \mid (k \times n) \iff m \mid n$$

6. Prove or disprove that: For all natural numbers $n \in \mathbb{N}$, $2 \mid 2^n$.

7. Find a counterexample for the statement:

$$\forall k, m, n \in \mathbb{Z}^+.\ m \mid n \wedge n \mid k \implies (m \times n) \mid k$$

8. Show that for all integers $m, n \in \mathbb{Z}$:

$$m \mid n \vee n \mid m \implies m = n \vee m = -n$$

9. Prove or disprove:

$$\forall k, m, n \in \mathbb{Z}.\ k \mid (m \times n) \implies k \mid m \vee k \mid n$$

10. Let $P(n)$ be a predicate on natural numbers. Let us define the predicate $P^{\#}(n)$ as follows:

$$P^{\#}(n) \triangleq \forall k \in \mathbb{N}.\ 0 \leq k \leq n \implies P(k) \qquad (n \in \mathbb{N})$$

    (i) Show that:

$$\forall \ell \in \mathbb{N}.\ P^{\#}(\ell) \implies P(\ell)$$

    (ii) Exhibit a concrete statement $P(n)$ and a witness for $n \in \mathbb{N}$ for which the following statement does *not* hold:

$$P(n) \implies P^{\#}(n)$$

    (iii) Prove the following:
    - $P^{\#}(0) \iff P(0)$
    - $\forall n \in \mathbb{N}.\ (P^{\#}(n) \implies P^{\#}(n+1)) \iff (P^{\#}(n) \implies P(n+1))$
    - $(\forall n \in \mathbb{N}.\ P^{\#}(n)) \iff (\forall n \in \mathbb{N}.\ P(n))$