CO519 - Theory of Computing - Logic

Part A : Propositional logic and its natural deduction proofs

Dominic Orchard
School of Computing, University of Kent

Last updated on September 12, 2018

These are the accompanying notes for CO519 (the logic part). They provide a counterpart to the lectures, but do not replace them; the lectures will provide content, detail, and discussion not given in these notes.

These notes also provide a counterpart to the course textbook, *Logic in Computer Science* (by Huth and Ryan) which I recommend. We will only cover material from the first two chapters due to the short length of this part of the course. I recommend reading these chapters, but the assessable material for this course is covered in lectures and these notes (excluding the appendix); the textbook contains extra detail which is not assessed, but worth learning anyway.

If you spot any errors or have suggested edits, the notes are written in LaTeX and are available on GitHub at `http://github.com/dorchard/co519-logic`. Please fork and submit a pull request with any suggested changes.

1 Natural deduction for propositional logic

Truth tables are a handy way to give meaning to logical operators and formulas, in terms of truth or falsehood. But they are difficult to use for reasoning about anything but small formulas since the number of rows is 2^n where n is the number of contingent formulas. For example, the truth table for $P \vee (Q \wedge R) \rightarrow (P \vee Q) \wedge (P \vee R)$ has $2^3 = 8$ rows as there are three contingent formulas: P, Q, and R whose truth or falsehood determines the truth or falsehood of the overall formula. Calculating even this modest truth table takes some significant calculation to enumerate all possibilities.

Instead, logicians have constructed formal languages (calculi) for building complex chains of reasoning (proofs) in a more compact form. In this course, we use the *natural deduction* calculus due to 20^{th} century logicians such as Gerhard Gentzen (who formulated natural deduction in 1934) and Dag Pravitz who promoted this style in the 1960s (and converted various other results of Gentzen into the natural deduction style).

Natural deduction provides a system of *inference rules* which explain how to construct and deconstruct formulas to build a logical argument. These rules represent the derivation of one formula (the *conclusion*) from several other formulas that are assumed or known to be true (the *premises*) via the notation:

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

Each logical operator (like conjunction $\wedge$ and disjunction $\vee$), will have one or more rules for *introducing* that operator (deriving a conclusion using that operator) and one or more rules for *eliminating* that operator (deriving a conclusion from a formula using that operator). These rules can be stacked together to form a logical argument: *a proof*, which looks like the following:

$$\dfrac{\dfrac{P_1 \quad P_2}{P_3} \quad P_4}{P_5}$$

This isn't an actual concrete proof yet, it's just an example of how a natural deduction proof looks when you stack together its inference rules. Informally, we might have something like the following:

$$\dfrac{\dfrac{\textit{I forgot my coat} \quad \textit{It's raining}}{\textit{I get wet}} \quad \textit{My hairdryer broke}}{\textit{My hair remains wet}}$$

But this isn't always the most helpful format to derive the proofs. Instead, we'll use a special "box"-like notation called Fitch-style which you can find in the recommend reading textbook *Logic in Computer Science* by Huth and Ryan. We will still apply the rules of natural deduction, but the Fitch-style gives us a nice way to layout the proof as we are deriving it.

We will step through the natural deduction rules for the core logical operators: $\wedge$ (conjunction/and), $\vee$ (disjunction/or), $\rightarrow$ (implication/if-then), $\neg$ (negation), as well as truth $\top$ and falsehood/falsity, which is often written in propositional logic as $\perp$ (pronounced "bottom").[1] Note that, in the literature, logical operators are sometimes alternatively called *logical connectives*.

1.1 Properties of formulas

We will consider three properties that a logical formula may have:

- *valid*: a formula which is always true (also called a *tautology*). In this part of the course, we will mostly prove the validity of formulas. For example, $P \wedge Q \rightarrow Q \wedge P$ is true no matter the truth/falsehood of P and Q. This will be the main property we consider in *Part A* (these notes).

- *satisfiable*: a formula which is true for *some* assignments of truth/falsehood to the atoms/variables it contains, *e.g.*, $a \wedge b$ is satisfiable, and the *satisfying assignment* is that $a \mapsto \mathsf{T}$ and $b \mapsto \mathsf{T}$. A valid formula is trivially satisfiable (true for all assignments).

- *unsatisfiable*: a formula which is always false (all rows in the truth table are false) regardless of assignments to the variables/atom, *e.g.* $P \wedge \neg(P \vee Q)$. We can prove a formula is unsatisfiable by proving its negation is valid.

[1] *Bottom* $\perp$ is often used in maths and computer science to represent undefined values or behaviour. In logic, if have arrived at falsity $\perp$ during a proof then we are in a situation where anything could be true as we've arrived at a logically inconsistent situation. This is sometimes quite useful for doing proofs-by-contradiction, as we will see in Section 1.5.

1.1.1 Entailment and sequents

Suppose we have a set of formulas $P_1, \ldots, P_n$ from which we want to prove Q by applying the rules of a particular logic (propositional logic here). The formulas $P_1, \ldots, P_n$ are the premises and Q is our goal conclusion. This is often written using the following notation called a *sequent*:

$$P_1, \ldots, P_n \vdash Q$$

The turnstile symbol $\vdash$ is read as *entails* and the premises to the left are sometimes call the *context* of assumed formulas. This is a compact representation of a formula Q along with any assumptions used to deduce it.

We say this sequent is *valid* if there is a proof that can be constructed by deriving the conclusion from the premises. For example, $P \wedge Q \vdash Q \wedge P$ is valid, meaning from $P \wedge Q$ there is a proof of $Q \wedge P$. If we wish to explain that a sequent is invalid we can write $P \nvdash Q$ meaning, it is not valid that P entails Q.

When there are no premises, we often drop the $\vdash$, *i.e.*, writing something like "$(P \wedge Q) \rightarrow P$ *is valid*" instead of "$\vdash (P \wedge Q) \rightarrow P$ *is valid*".

1.2 Conjunction ("and")

Recall the truth table for conjunction:

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

From this, we see that in order to conclude the truth of $P \wedge Q$ we need the truth of P and the truth of Q. This justifies the following natural deduction rule for *introducing* conjunction:

$$\frac{P \qquad Q}{P \wedge Q} \wedge_i$$

The label is subscripted with 'i' for introduction. That is, given two premises; P is true and Q is true, then $P \wedge Q$ is true. There is just one introduction rule, corresponding to the fact that there is only one "true" row for $P \wedge Q$ in the truth table (highlighted in yellow above).

Note that the P and Q are place-holders here for *any propositional formula* so we could instantiate the rule, *e.g.*, to something like this if we needed it:

$$\frac{(P \vee R \rightarrow Q) \quad S}{(P \vee R \rightarrow Q) \wedge S} \wedge_i$$

What about elimination? Reading the highlighted line in the truth table from right-to-left shows how to *eliminate* a conjunction, *i.e.*, what smaller formulas can we conclude are true if we know that $P \land Q$ is true? We get two rules:

$$\frac{P \land Q}{P}\ \land_{e1} \qquad\qquad \frac{P \land Q}{Q}\ \land_{e2}$$

The labels have a subscript 'e' for elimination; this convention will continue. (*Aside*: If the syntax of the inference rules let us have multiple conclusions then we could collapse the two eliminations into one rule, but natural deduction instead has single conclusions. There are different proof systems which allow multiple conclusions (like the *sequent calculus*) but we won't cover that here).

Let's write a proof with just these rules by stacking them together.

Example 1. For any formula P, Q, R then $P \land (Q \land R) \vdash (P \land Q) \land R$ is valid, *i.e.* given $P \land (Q \land R)$ we can prove $(P \land Q) \land R$.

$$\dfrac{\dfrac{\dfrac{P \land (Q \land R)}{P}\land_{e1} \quad \dfrac{\dfrac{P \land (Q \land R)}{Q \land R}\land_{e2}}{Q}\land_{e1}}{P \land Q}\land_{i} \quad \dfrac{\dfrac{P \land (Q \land R)}{Q \land R}\land_{e2}}{R}\land_{e2}}{(P \land Q) \land R}\land_{i}$$

The root of the tree is our goal $(P \land Q) \land R$, which is built from the premises on the line above. These chains of reasoning go up to the "leaves" of the tree, which is the assumed formula $P \land (Q \land R)$. At each step (each line) we've applied either conjunction introduction or one of the conjunction elimination rules (as can be seen from the labels on the right).

The following exercise is to prove the converse of the above property.

Exercise 1.1. Prove that $(P \land Q) \land R \vdash P \land (Q \land R)$ is valid by instantiating and stacking together inference rules.

This proof, and the one above, together imply that conjunction $\land$ is *associative*, *i.e.*, $P \land (Q \land R) = (P \land Q) \land R$.

Fitch-style proof So far we have constructed proofs by stacking natural deduction inference rules on top of each other. This leads us towards a *bottom-up* proof strategy starting with the goal and working up towards the premises. In this course we are going to mostly use a *top-down* approach called "Fitch-style". This style begins with assumptions, numbers each line of a proof, and uses indentation and boxes to represent sub-proofs and the scope of their assumptions.

The above proof is rewritten in the following way in Fitch notation:

1.	$P \land (Q \land R)$	premise	
2.	P	$\land_{e1}$ 1	
3.	$Q \land R$	$\land_{e2}$ 1	
4.	Q	$\land_{e1}$ 3	
5.	R	$\land_{e2}$ 3	
6.	$P \land Q$	$\land_{i}$ 2, 4	
7.	$(P \land Q) \land R$	$\land_{i}$ 6, 5	$\square$

The proof follows in a number of linear steps. On the left we number each line of the argument. On the right, we explain which rule was applied to which formula, *e.g.*, on the second line we have applied conjunction elimination $\wedge_{e1}$ to line 1 to get the formula P. Or for example, on line 6, conjunction introduction is applied to lines 2 and 4 to get $P \wedge Q$. We finish on line 7 with our goal, which is marked with $\Box$ which a way of saying the proof is finished and we've reached our goal (the symbol means $Q.E.D$ which is an abbreviation of *quod erat demonstrandum*, Latin for "what was to be demonstrated").

We haven't used any sub-proofs yet (which have a box drawn around them); these appear in the next subsection on implication.

Order of numbers in labels Note that the order of the line numbers in labels tells us the order of the premises to a natural deduction rule and so the order is important. For example, line 6 above applies ($\wedge_i$ 2, 4) to introduce $P \wedge Q$, but if it was actually ($\wedge_i$ 4, 2) we would be introducing $Q \wedge P$ which is not our intended goal.

Exercise 1.2. Rewrite your proof to Exercise 1.1 using Fitch style.

Remark. (**important**) Depending on what is being proved, a top-down approach (starting from the premises) or bottom-up approach (starting from the goal/conclusion) can be easier. In practice, if you are stuck it can help to start *at both ends* and work towards the middle. You can do this by putting the goal near a bottom of a piece of paper, giving enough space to meet in the middle.

It doesn't matter if things get messy– the primary goal is to reach a proof. You can rewrite it afterwards to be more clear; you should do so in your class work and assessments.

1.3 Implication

Recall the truth table for implication:

P	Q	$P \to Q$
F	F	T
F	T	T
T	F	F
T	T	T

Implication $P \to Q$ is interesting because if $\neg P$ (if P is false) then Q can be true or false, *i.e.*, Q can be anything if P is false (the top two lines).

Exercise 1.3. Recall that $P \to Q = \neg P \vee Q$. Show this is true by comparing the truth tables for each side of this equation.

As with conjunction, we'll consider the two style of rule: elimination and introduction. The elimination rule for implication in natural deduction is:

$$\frac{P \to Q \qquad P}{Q} \to_e$$

This rule is also known as *modus ponens*.[2] It says that if we know $P \to Q$ and we know P then we know Q. You can verify the soundness of this rule by looking at the truth table: indeed Q is true when both $P \to Q$ and P are true.

There are various other natural deduction rules one might construct by looking at the truth table– but this one can be used to derive the others. The particular set of natural deduction rules we look at was carefully honed by logicians to provide a kind of "minimal" calculus for proofs.

Introduction of an implication $P \to Q$ follows from a *subproof* (which is drawn in a box) which starts with an assumption of P and ends with Q as a conclusion after any number of steps. The rule is written as follows:

$$\frac{\boxed{\begin{array}{c} P \\ \vdots \\ Q \end{array}}}{P \to Q} \to_i$$

Thus subproofs in both tree- and Fitch-style proofs are of the form:

$$\boxed{\begin{array}{c} \textit{assumption} \\ \vdots \\ \textit{conclusion} \end{array}}$$

Remark. (**important**) When we start a subproof box, the first formula is always an assumption. When the box is closed, the assumption does not go away but becomes the premise of the implication when applying the $\to_i$ rule.

This is an important point: when proving a theorem we have to be careful not to introduce additional assumptions which are not part of the theorem. For example, let's say we are proving a theorem expressed by a formula Q but in doing so we assume P but P is not one of Q's assumptions. Then instead we will have proved $P \to Q$ rather than Q. This is something to keep in mind when writing complex proofs. The proof system of natural deduction allows us to keep track of our assumptions and their eventual inclusion in the final result.

Aside: mechanised proof assistants (software systems in which we can write machine-checked proofs, such as *Coq*, *Isabelle*, *Agda*) have a similar basis to natural deduction and give us confidence and precision in writing proofs.

There is an alternate presentation of natural deduction called *sequent-style natural deduction*, which is described in Appendix A, where the inference rules are expressed in terms of sequents $P_1, \ldots, P_n \vdash Q$. This won't be assessed on the course, but is worth looking at if you want to read more widely on logic. Another proof calculus (also due to Gentzen) is the *sequent calculus* which won't be described here, but there is plenty of information online.

[2] *modus ponens* is short for the Latin phrase *modus ponendo ponens* which means "the way that affirms by affirming".

Example 2. The following simple formula about conjunction and implication is valid: $\vdash (P \wedge Q) \to (Q \wedge P)$. Here is its proof in Fitch-style:

1.	$P \wedge Q$	assumption
2.	P	$\wedge_{e1}$ 1
3.	Q	$\wedge_{e2}$ 1
4.	$Q \wedge P$	$\wedge_i$ 3, 2
5.	$P \wedge Q \to Q \wedge P$	$\to_i$ 1-4 $\square$

In the last line, we apply implication introduction and we label it with the range of the lines of the subproof used (in this case 1-4).

Remark. In Example 1 we proved that given $P \wedge (Q \wedge R)$ then $(P \wedge Q) \wedge R$. We can turn this into an implication $P \wedge (Q \wedge R) \to (P \wedge Q) \wedge R$ simply by using implication introduction on the original proof.

Example 3. The following is valid:

$$(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$$

Here is its proof:

1.	$P \to (Q \to R)$	ass.
2.	$P \to Q$	ass.
3.	P	ass.
4.	$Q \to R$	$\to_e$ 1, 3
5.	Q	$\to_e$ 2, 3
6.	R	$\to_e$ 4, 5
7.	$P \to R$	$\to_i$ 3-6
8.	$(P \to Q) \to (P \to R)$	$\to_i$ 2-7
9.	$(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$	$\to_i$ 1-8 $\square$

Here we have an example of multiple nesting of subproofs. (Tip: I proved this by working top-down and bottom-up at the same time, which was made easier by typing the proof).

Occasionally it is useful to "copy" a formula from earlier in a proof. For example, the following proof of $\vdash P \to P$ copies a formula from one line of the proof to the other in order to introduce a trivial implication:

1.	P	assumption
2.	P	copy 1
3.	$P \to P$	$\to_i$ 1-2 $\square$

Exercise 1.4. Prove $P \to (Q \to P)$ is valid.

1.3.1 Bi-implication ("if and only if")

Propositional logic often includes the bi-implication operator $\leftrightarrow$ also read as "*if and only if*" and sometimes written as *iff* (double f). A bi-implication $P \leftrightarrow Q$ is equivalent to the conjunction of two implications, pointing in opposite directions:

$$P \leftrightarrow Q \stackrel{\text{def}}{=} (P \to Q) \wedge (Q \to P)$$

This means that P and Q have exactly the same truth table, or we say they are *equivalent* (see Section 2 later).

Therefore to construct or deconstruct a logical bi-implication one can consider it as "implemented" by conjunction and implication, reducing the number of introduction/elimination rules that need to be remembered. Nonetheless, thinking about what these would be is a nice exercise.

Exercise 1.5. (optional) Try to derive your own elimination and introduction rules for bi-implication. There is usually one introduction and two eliminations.

1.4 Disjunction ("or")

Recall the truth table for disjunction (which has the three rows in which $P \vee Q$ is true):

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

The fact that we can conclude $P \vee Q$ from either P or from Q separately justifies the following two introduction rules for disjunction in natural deduction:

$$\frac{P}{P \vee Q}\vee_{i1} \qquad \frac{Q}{P \vee Q}\vee_{i2}$$

Example 4. Prove $(P \wedge Q) \to (P \vee Q)$ is valid.

1.	$P \wedge Q$	assumption
2.	P	$\wedge_{e1}$ 1
3.	$P \vee Q$	$\vee_{i1}$ 2
4.	$P \wedge Q \to P \vee Q$	$\to_i$ 1-3 $\square$

This could have been written equivalently as a natural deduction tree:

$$\dfrac{\boxed{\dfrac{\dfrac{P \wedge Q}{P}\wedge_{e1}}{P \vee Q}\vee_{i1}}}{(P \wedge Q) \to (P \vee Q)}\to_i$$

This will be the last tree-based proof we see; from now on we'll just keep using the Fitch style.

What about disjunction elimination? Given the knowledge that $P \vee Q$ is true then what can be conclude? Either P is true, or Q is true, or both are true. Therefore, we don't know exactly what true formulas we can derive from the truth of $P \vee Q$, we just know a selection of possibilities.

The natural deduction way of eliminating disjunction is to have two subproofs as premises which are contingent on the assumption of either P or Q:

$$\dfrac{P \vee Q \quad \boxed{\begin{array}{c} P \\ \vdots \\ R \end{array}} \quad \boxed{\begin{array}{c} Q \\ \vdots \\ R \end{array}}}{R} \vee_e$$

Example 5. For any propositions P, Q, R then $(P \wedge Q) \vee (P \wedge R) \to P$ is valid.

1.	$(P \wedge Q) \vee (P \wedge R)$	assumption
2.	$P \wedge Q$	assumption
3.	P	$\wedge_{e1}$ 2
4.	$P \wedge R$	assumption
5.	P	$\wedge_{e1}$ 4
6.	P	$\vee_e$ 1, 2-3, 4-5
7.	$(P \wedge Q) \vee (P \wedge R) \to P$	$\to_i$, 1-6 $\square$

You can see that the application of disjunction elimination $\vee_e$ involves three things: a disjunctive formula (line 1) and two subproofs (lines 2-3 and lines 4-5) which respectively assume the two subformulas of disjunction and conclude with the same formula (P), which forms the conclusion of the subproof on line 6.

Exercise 1.6. Prove $P \vee Q \to Q \vee P$ is valid.

Remark. From looking at the truth table for disjunction, one might wonder why disjunction elimination does not look like:

$$\dfrac{P \vee Q \quad \boxed{\begin{array}{c} P \\ \vdots \\ R \end{array}} \quad \boxed{\begin{array}{c} Q \\ \vdots \\ R \end{array}} \quad \boxed{\begin{array}{c} P \wedge Q \\ \vdots \\ R \end{array}}}{R} \vee_e$$

This would match more closely the idea of reading the truth-table "backwards" from right-to-left on true values of $P \vee Q$. The reason we don't have this is that natural deduction strives for minimality and the third subproof with assumption $P \wedge Q$ is redundant since if we have $P \wedge Q$ true we can apply either the subproof $\boxed{P \ldots R}$ or the subproof $\boxed{Q \ldots R}$ by first applying $\wedge_{e1}$ or $\wedge_{e2}$ to the assumption $P \wedge Q$ to get P or Q respectively.

1.5 Negation

Negation introduction and elimination are given by:

$$\frac{\boxed{\begin{array}{c}P\\ \vdots\\ \bot\end{array}}}{\neg P}\,\neg_i \qquad \frac{P \quad \neg P}{\bot}\,\neg_e$$

Introduction says that given a proof that assumes P but ends in falsehood $\bot$ then we know $\neg P$, this is similar to the notion of *proof by contradiction*, which is derived from this (see below).

Elimination states that given a proof of P and a simultaneous proof of $\neg P$ then we conclude falsehood $\bot$, *i.e.*, we have a logical inconsistency on our hands and so end up proving false: P and $\neg P$ cannot both be true at the same time.

Example 6. For all P, Q then $P \to Q \vdash \neg Q \to \neg P$.

1.	$P \to Q$	premise
2.	$\neg Q$	assumption
3.	P	assumption
4.	Q	$\to_e$ 1, 3
5.	$\bot$	$\neg_e$ 2, 4
6.	$\neg P$	$\neg_i$ 3-5
7.	$\neg Q \to \neg P$	$\to_i$ 2-6 $\square$

Remark. This example is often given as a derived inference rule called *modus tollens*[3] that is similar to modus ponens (implication elimination):

$$\frac{P \to Q \quad \neg Q}{\neg P}\,mt$$

If an inference rule can be derived from others we say it is *admissible*. The system of rules we take as the basis for natural deduction reasoning contains no admissible rules.

Remark. If we want to prove a formula P is unsatisfiable then we can instead prove that $\neg P$ is valid (always true), hence proving that P is unsatisfiable (always false).

Exercise 1.7. Prove that $P \wedge \neg(P \vee Q)$ is unsatisfiable.

Remark. Some formulae are not valid, *e.g.*, $P \to \neg P$, which can be seen from drawing its true table. However, this formula is *satisfiable*, if P is false then $P \to \neg P$ is true. Natural deduction does not help us to prove satisfiability. Part B will look at algorithmic approaches to deciding satisfiability.

[3] *modus tollens* is short for the Latin phrase *modus tollendo tollens* which means "the way that denies by denying".

1.5.1 Double negation

A special rule holds called *double-negation elimination* which allows us to remove double negations on a proposition:

$$\frac{\neg\neg P}{P}\ \neg\neg_e$$

Example 7. The principle of *proof by contradiction* is represented by following the derived inference rule:

$$\frac{\boxed{\begin{array}{c}\neg P\\ \vdots\\ \bot\end{array}}}{P}\ \text{PBC}$$

That is, if we assume $\neg P$ and conclude $\bot$, then we have P. To show how to derive this, let the subproof in the above rule be called Δ, then we construct the following proof:

$$\begin{array}{ll}
1. & \boxed{\begin{array}{ll}\neg P & \text{ass.}\\ \Delta & \vdots\\ \bot & \end{array}}\\
2. & \\
3. & \\
4. & \neg\neg P \quad \neg_i\ 1\text{-}3\\
5. & P \quad \neg\neg_e\ 4
\end{array}$$

Of course, Δ might be much longer than 3 lines, but we use the numbering in the above proof for clarity.

1.6 Truth and falsity

If we have $\bot$ (false), then we can derive any formula:

$$\frac{\bot}{P}\ \bot_e$$

There is no $\bot$ introduction as such, though $\neg_e$ provides a kind of $\bot$ introduction (from conflicting formula). Dually, we can always introduce truth from no premises, but there is no elimination:

$$\frac{}{\top}\ \top_i$$

1.7 A further derived rule: Law of Excluded Middle

An interesting rule that we can derive in the propositional logic is called the *Law of Excluded Middle* or LEM for short. It says that for any formula P we have the following valid rule:

$$\frac{}{P \vee \neg P}\ \text{LEM}$$

i.e., whatever P is, then either P is true or $\neg P$ is true. Here is its derivation:

1.	$\neg(P \vee \neg P)$	ass.
2.	P	ass.
3.	$P \vee \neg P$	$\vee_{i1}$ 2
4.	$\perp$	$\neg_e$ 3, 1
5.	$\neg P$	$\neg_i$ 2-4
6.	$P \vee \neg P$	$\vee_{i2}$ 5
7.	$\perp$	$\neg_e$ 6, 1
8.	$\neg\neg(P \vee \neg P)$	$\neg_i$ 1-7
9.	$P \vee \neg P$	$\neg\neg_e$ 8 $\square$

This rule can be useful in particular proofs.

Exercise 1.8. Using LEM, prove that $P \to Q \vdash \neg P \vee Q$ is valid.

Aside: constructive vs non-constructive logic In this course, we study a particular kind of propositional logic called *classical* or *non-constructive* logic. Another variant is known as *intuitionistic* or *constructive* logic which has a slightly different set of inference rules: $\neg\neg_e$ is not included. By removing double-negation elimination we can no longer derive proof-by-contradiction or LEM.

The central principle of constructive logic is to reason about *proof* rather than *truth* (as in classical logic). In constructive logic, a formula P represents the proof of formula P: a mathematical object witnessing the truth of P which we can separately analyse. The inference rules of natural deduction are now about preserving proof rather than truth, *e.g.*, conjunction elimination says given a proof of $P \wedge Q$ then we can prove P.

In constructive logic, $\neg\neg_e$ is rejected since it would mean we can get a proof of P from a proof of the negation of the negation of P, but this proof is not a proof of P. This is particularly troublesome when using $\neg\neg_e$ to prove LEM. If LEM was allowed in constructive logic, then for any formula P we can construct either a proof of P or a proof of $\neg P$. But what is that proof and where has it come from? Out of thin air! (LEM has no premises). The essence of constructive logic is to disallow such things so that we always know we have a concrete proof for our formulas, constructed from proofs of its subformulas or premises. Section 1.2.5 of the Huth and Ryan course textbook gives some more detail and shows an example mathematical proof about rational numbers in classical logic which cannot be proved constructively.

Constructive logics are useful because they correspond to type systems in functional programming: a result known as the *Curry-Howard correspondence*. Unfortunately, we will not have time to study that here.

2 Algebraic properties of logic

In Section 1.3.1 we saw *bi-implication* $\leftrightarrow$, a derived logical operator where:

$$P \leftrightarrow Q \stackrel{\text{def}}{=} (P \rightarrow Q) \wedge (Q \rightarrow P)$$

If there is a bi-implication between two formula then it means their truth tables are exactly the same and we can see the two formulas as equivalent. From the natural deduction rules (or from the truth tables) a number of general equivalences can be derived which give us algebraic laws about propositional formula (this was first described by Boole in 1847). We'll use the operator $=$ to denote equivalent formula, where $P = Q$ can be read as "P is equivalent to Q" (or that there is a bi-implication). We can thus prove equivalences by proving the bi-implication of the two formula.

The following lists a number of equivalences between general propositional formula which amounts to algebraic properties of conjunction and disjunction:

property	conjunction	disjunction
idempotence	$P \wedge P = P$	$P \vee P = P$
commutativity	$P \wedge Q = Q \wedge P$	$P \vee Q = Q \vee P$
associativity	$(P \wedge Q) \wedge R = P \wedge (Q \wedge R)$	$(P \vee Q) \vee R = P \vee (Q \vee R)$
unitality	$P \wedge \top = P$	$P \vee \bot = P$
annihilation	$P \wedge \bot = \bot$	$P \vee \top = \top$

We will often refer to these as "axioms" though really we can derive them via natural deduction, or as "algebraic laws".

(Aside: you might like to think about other notions in mathematics that have similar axioms or a subset of these axioms, e.g. addition is commutative, associative, has 0 as its unit, but is not idempotent and does not have an annihilator, unless we extend integers with ∞).

Exercise 2.1. Prove the algebraic property of unitality for conjunction, i.e., that $P \wedge \top = P$.

"Distributivity" and "absorption" laws give us a relationship between $\wedge$ and $\vee$:

$$\begin{array}{rrcl}
\textit{distributivity} & (P \vee Q) \wedge R & = & (P \wedge R) \vee (Q \wedge R) \\
& (P \wedge Q) \vee R & = & (P \vee R) \wedge (Q \vee R) \\
\textit{absorption} & (P \wedge Q) \vee P & = & P \\
& (P \vee Q) \wedge P & = & P
\end{array}$$

De Morgan's two laws give us a useful interaction between negation and conjunction and disjunction respectively:

$$\begin{array}{rrcl}
\textit{De Morgan's} & \neg(P \wedge Q) & = & \neg P \vee \neg Q \\
& \neg(P \vee Q) & = & \neg P \wedge \neg Q
\end{array}$$

"Complementation" laws give us interaction between a formula and its negation:

$$\begin{aligned} \textit{complementation} \qquad P \wedge \neg P \;&=\; \bot \\ P \vee \neg P \;&=\; \top \end{aligned}$$

Note that the second complementation law here only hold in classical (non-constructive) logic (which we study here primarily) where we allow the law of excluded middle, derived from double-negation elimination. Relatedly, the following law, known as "involution" only holds when we have double negation elimination:

$$\textit{involution} \qquad \neg\neg P \;=\; P$$

This notion of equivalence, given by the operator =, is a *congruence* with respect to all the operators of logic. This means that if we have an equivalence between two formula we can get an equivalence between larger formulas by "plugging" the first equivalence into a template for a formula. For example, we get the following congruence property for conjunction:

$$P = Q \quad \Rightarrow \quad P \wedge R = Q \wedge R$$

We can think of this as plugging the equivalence $P = Q$ into a formula template $- \wedge R$. A similar congruence property holds for the template $R \wedge -$ and for all the other operators, e.g., disjunction (with templates $R \vee -$ and $- \vee R$), and implication, and negation.

The congruence property of = is useful because it means we can use equivalences to "rewrite" parts of a propositional formula. We can then given *equational proofs* that some formula P is equivalent to another formula Q by applying algebraic laws one at a time, possibly to subparts of a formula.

Example 8. Prove that $P \vee (Q \wedge \neg P) = Q \vee P$.

We can proceed in the following steps where the subformula to which I have applied an equality is underlined and the rule which I am applying is written on the right:

$$\begin{aligned} &\underline{P \vee (Q \wedge \neg P)} && \{\textit{distributity}\} \\ =\; &(P \vee Q) \wedge \underline{(P \vee \neg P)} && \{\textit{complementation}\} \\ =\; &\underline{(P \vee Q) \wedge \top} && \{\textit{unitality of conjunction}\} \\ =\; &P \vee Q && \{\textit{commutativity of disjunction}\} \\ =\; &Q \vee P && \square \end{aligned}$$

Exercise 2.2. Prove that $P \wedge \neg(P \wedge Q) = P \wedge \neg Q$.

3 Collected rules of natural deduction

	Introduction	*Elimination*
$\wedge$	$\dfrac{P \qquad Q}{P \wedge Q}\ \wedge_i$	$\dfrac{P \wedge Q}{P}\ \wedge_{e1} \qquad \dfrac{P \wedge Q}{Q}\ \wedge_{e2}$
$\vee$	$\dfrac{P}{P \vee Q}\ \vee_{i1} \qquad \dfrac{Q}{P \vee Q}\ \vee_{i2}$	$\dfrac{P \vee Q \quad \boxed{\begin{array}{c} P \\ \vdots \\ R \end{array}} \quad \boxed{\begin{array}{c} Q \\ \vdots \\ R \end{array}}}{R}\ \vee_e$
$\rightarrow$	$\dfrac{\boxed{\begin{array}{c} P \\ \vdots \\ Q \end{array}}}{P \rightarrow Q} \rightarrow_i$	$\dfrac{P \rightarrow Q \qquad P}{Q} \rightarrow_e$
$\neg$	$\dfrac{\boxed{\begin{array}{c} P \\ \vdots \\ \bot \end{array}}}{\neg P}\ \neg_i$	$\dfrac{P \qquad \neg P}{\bot}\ \neg_e$
$\top$	$\dfrac{}{\top}\ \top_i$	
$\bot$		$\dfrac{\bot}{P}\ \bot_e$
$\neg\neg$	(derivable: $\dfrac{P}{\neg\neg P}\ \neg\neg_i$)	$\dfrac{\neg\neg P}{P}\ \neg\neg_e$

These are all the rules we have and need for propositional proofs. You should aim to know all of the above rules by the end of the course/exam.

We derived other useful inference rules from these rules, like modus tollens, proof-by-contradiction, and law-of-excluded-middle. They are useful to know but the table above gives the essential rules for propositional proofs.

4 Exercises

This section collects together the exercises given so far. They may not all be covered in lectures, so they provide useful additional examples to practise on.

Exercise 1.1. Prove that $(P \wedge Q) \wedge R \vdash P \wedge (Q \wedge R)$ is valid by instantiating and stacking together inference rules.

Exercise 1.2. Rewrite your proof to Exercise 1.1 using Fitch style.

Exercise 1.3. Recall that $P \to Q = \neg P \vee Q$. Show this is true by comparing the truth tables for each side of this equation.

Exercise 1.4. Prove $P \to (Q \to P)$ is valid.

Exercise 1.5. (optional) Try to derive your own elimination and introduction rules for bi-implication. There is usually one introduction and two eliminations.

Exercise 1.6. Prove $P \vee Q \to Q \vee P$ is valid.

Exercise 1.7. Prove that $P \wedge \neg(P \vee Q)$ is unsatisfiable.

Exercise 1.8. Using LEM, prove that $P \to Q \vdash \neg P \vee Q$ is valid.

Exercise 2.1. Prove the algebraic property of unitality for conjunction, i.e., that $P \wedge \top = P$.

Exercise 2.2. Prove that $P \wedge \neg(P \wedge Q) = P \wedge \neg Q$.

A Sequent-style natural deduction (not examinable)

Recall from Section 1.1.1 that a sequent is a compact representation of a formula P along with any assumptions used to deduce it, written in the form:

$$P_1, \ldots, P_n \vdash P$$

The turnstile symbol $\vdash$ is read as *entails* and the premises to the left are called the *context* of assumed formulas. The right-hand side is the *conclusion.* For example, the following judgment captures the idea of conjunction introduction:

$$P, Q \vdash P \wedge Q$$

An alternate formulation of natural deduction gives the usual introduction and elimination rules in sequent form, making explicit the assumption context of the formula. This sequent-style of natural deduction is not assessed in CO519, but is included here for completeness and to help with any wider reading.

A key rule that was implicit in the previous formulation of natural deduction is the use of an assumption as a formula. This is usually called the *axiom* rule:

$$\frac{}{\Gamma, P \vdash P}\ (axiom)$$

This says that given some context with an assumption P and any other assumptions, represented by the Greek symbol Γ (uppercase gamma)[4] then we can conclude P. This is similar to the idea of copying in Fitch-style proofs.

The order of assumptions on the left of $\vdash$ is not important (we can freely move assumptions around). The sequent style captures that there may be other assumptions Γ in scope. A meta rule says that we can add arbitrary redundant assumptions into our context (called *weakening*):

$$\frac{\Gamma \vdash A}{\Gamma, \Gamma' \vdash A}\ (\textit{weaken})$$

This is useful when we have two subproofs that we want to make have the same set of assumptions (see below). The rest of the rules are the introduction and elimination rules for operators.

Conjunction

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q}\wedge_i \quad \frac{\Gamma \vdash P \wedge Q}{\Gamma \vdash P}\wedge_{e1} \quad \frac{\Gamma \vdash P \wedge Q}{\Gamma \vdash Q}\wedge_{e2}$$

These rules are very similar to the previously shown natural deduction rules, but they now carry a context of assumptions Γ. If the context doesn't match between two premises, weakening (above) can be applied so that they match.

Disjunction

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q}\vee_{i1} \quad \frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q}\vee_{i2} \quad \frac{\Gamma \vdash P \vee Q \quad \Gamma, P \vdash R \quad \Gamma, Q \vdash R}{\Gamma \vdash R}\vee_e$$

The disjunction elimination rule is much less unruly than the previous formulation, but has the same meaning. Note, we are now extending the context of assumptions with P and Q in the last two premises.

Implication

$$\frac{\Gamma \vdash A \rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}\rightarrow_e \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}\rightarrow_i$$

As an example, here is the proof of $P \wedge Q \rightarrow P \vee Q$ in this style:

$$\cfrac{\cfrac{\cfrac{\cfrac{}{P \wedge Q \vdash P \wedge Q}\,\textit{axiom}}{P \wedge Q \vdash P}\wedge_{e1}}{P \wedge Q \vdash P \vee Q}\vee_{i1}}{\vdash (P \wedge Q) \rightarrow (P \vee Q)}\rightarrow_i$$

Negation, falsity, and truth

$$\frac{\Gamma, P \vdash \bot}{\Gamma \vdash \neg P}\neg_i \qquad \frac{\Gamma \vdash P \; \Gamma \vdash \neg P}{\Gamma \vdash \bot}\neg_e \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash P}\bot_e \qquad \frac{}{\Gamma \vdash \top}\top_i$$

[4]Gamma Γ is the third letter of the Greek alphabet, corresponding to Latin C, hence Γ for "Context". Logicians like Greek as it gives them lots more symbols to use to represent things tersely. These are conventions which take some getting used to.