Logic and Proof

Solutions

2021

Contents

1. Introduction

1. Determine whether the truth value of the following sentences is true, false, unknown, or if the sentence is not a proper logical statement. Justify all of your answers, and give satisfying and falsifying valuations of the variables where appropriate.

   a) Broccoli is delicious.

   b) x likes broccoli.

   c) The area of a circle of radius r is πr^2.

   d) Larry Paulson was born in x.

   e) Larry Paulson est né aux États-Unis.

   f) This sentence is false.

   g) Every even number greater than 2 can be written as the sum of two primes.

   h) There exists a unique prime n that satisfies $a^n + b^n = c^n$ for some positive integers a, b and c.

   i) There are two people in Peterborough with the exact same number of hairs on their head.

> This exercise aims to demonstrate some interesting considerations in the study of logic and proof theory, in particular the blurry and ambiguous nature of human language and the nuanced relationship between statements and their proofs. Even considering whether the sentences above are valid logical statements assumes that we are working within the syntax and semantics of the English language – they would be meaningless (or, more precisely, not syntactically correct) if our syntax was propositional logic, for example! Even then, almost none of these can be unequivocally labelled as true, false, unknown or invalid, though objections can get quite weird and pedantic. Any answer with a suitable justification can be accepted. Some ideas:
>
> a) *Broccoli is delicious.* Invalid, because a subjective opinion, though devout broccoli fans/haters may argue that this is an universal fact/lie.
>
> b) *x likes broccoli.* True or false, if x is interpreted as a person that likes/doesn't like broccoli. Then again, "liking" things is not always a clear, binary choice so perhaps the statement cannot be given a truth value.
>
> c) *The area of a circle of radius r is πr^2.* True, as it is the well-known formula for the area of the circle. Interpreting the formula requires switching to the syntax of mathematics, with its own rules and conventions; one may say that π is a free variable and the statement is unknown until we give the interpretation $\pi \mapsto 3.14159265\ldots$ (which would of course only satisfy the statement if the value assigned is *exactly* π, not some finite-decimal-expansion approximation of it). Finally – and this is not even an unnecessarily pedantic point – the statement can only hold on the Euclidean plane; the formula is not valid for the area of a circle on a sphere or hyperbolic plane.
>
> d) *Larry Paulson was born in x.* Satisfiable with interpretation $x \mapsto$ United States, but also with $x \mapsto 1955$; we do not have notions of "types" that constrain the nature of domains in which the variables can be instantiated. Of course, there is nothing

stopping us from studying *many-sorted logic* where the variable can be constrained to a particular sort/type, and quantifying over elements of a specific set. For example, "$\exists x \in \mathbb{N}$. Larry Paulson was born in x" will only admit 1955 as a witness. It would also be fair to say that Larry Paulson may not necessarily refer to the course lecturer and without a valuation telling us what to do with the variable "Larry Paulson" we cannot assign a truth value to the statement.

e) *Larry Paulson est n'e aux 'Etats-Unis.* The sentence means "Larry Paulson was born in the United States", which is true; but it uses the syntax of French, rather than English. Thus, if we have made the assumption that we take English as the base syntax, we have to conclude that the sentence is syntactically not well-formed, despite semantically being true. Again, this may seem pedantic, but appreciating the distinction between the two concepts is very important for getting a grasp of the formal study of logic.

f) *This sentence is false.* A well-known *liar paradox* which is true if and only if it is false. Philosophers have been debating the truth, falsity, or validity of this sentence for a long time and will probably continue doing so for the foreseeable future. On the surface it is clearly nonsensical, but it is difficult to properly explain why (or why not!): it's not an opinion, question or command but a "perfectly normal" declarative assertion. The only fishy thing about it is the self-reference, but disallowing that would also disallow perfectly fine self-referential statements as well, and one may always try to cheat the system by elaborate indirectly self-referential statements. Even Alfred Tarski's trick of introducing an infinite hierarchy of "meta-levels" of language that can only assert the truth or falsity of statements in lower "object-levels" fails, since one may assert valid and meaningful statements about the infinite levels themselves that cannot be placed in the hierarchy, making the system incomplete. We could throw our hands up and say that the liar paradox is neither true nor false, but then consider "This sentence is not true": saying that it is neither true nor false (viz. not true and not false) must in particular imply that it is not true, from which the paradox still follows. For very detailed discussions of the liar paradox see the Wikipedia and Stanford Encyclopaedia of Philosophy pages on the subject (as well as a worrying number of Reddit posts claiming to have solved it).

g) *Every even number greater than 2 can be written as the sum of two primes.* A well-known unsolved mathematical statement called *Goldbach's Conjecture*. It is not known to be true or false since no proof or counterexample is known; it has been verified for integers below 4×10^{18} but of course one cannot extrapolate from that to all natural numbers. The conjecture is certainly true or not true (in the "law of excluded middle" sense), but we don't have a constructive evidence of for the truth of either disjunct.

h) *There exists a unique prime n that satisfies $a^n + b^n = c^n$ for some positive integers a, b and c.* The statement itself is true for $n = 2$ by our beloved *Pythagorean Theorem*: $a^2 + b^2 = c^2$ has infinitely many positive integer solutions (and the theorem probably

has infinitely many proofs, but at least 112). The fact that $n = 2$ is the unique such prime (or indeed, natural number not less than 2) follows from Pierre de Fermat's infamous *Last Theorem*, which was an unsolved problem for hundreds of years until it was finally proved by Andrew Wiles in 1994. Fermat claimed to have discovered a "truly marvellous proof of this" which he could not fit in the margin of the book he was annotating; all other similar comments he made were subsequently verified, but his Last Theorem (which was technically a conjecture until 1994) resisted most direct and indirect approaches. Mathematicians found proofs of special cases (for all composite numbers and certain classes of primes), but most attempts at the general statement resulted in failure. Andrew Wiles built on rather advanced results of algebraic geometry and number theory to establish a stronger result called the *Taniyama–Shimura–Weil conjecture* (now known as the *modularity theorem*) that shows a connection between concepts called elliptic curves and modular forms, and – as shown by Gerhard Frey – implies FLT as a corollary. In addition to being an inspiring underdog story (not that Wiles wasn't already an established mathematician, but anyone crazy enough to tackle FLT after 358 years of failed attempts would face an almost insurmountable challenge), it also highlights an interesting phenomenon in logic and mathematics: the simplicity of the statement of a theorem often has no relation to the complexity of the required proof. Finding solutions even to propositional logic formulas is NP-complete (intractable) in the general case, though proof systems and other methods you learn about in this course can handle non-pathological formulas with ease. Fermat's Last Theorem is very easy to state as a first-order logic formula on natural numbers, but proving it seems to require deriving it from a far more involved and general result because elementary number theory (which is certainly what Fermat had access to) is simply not powerful enough. While we can't know if Fermat really had a valid, simple and marvellous proof of the proposition, he almost certainly didn't make use of 20th century algebraic geometry which seems to be the only good avenue towards the result. Having said that, there isn't that much ongoing research or interest trying to tackle FLT by alternative means; Wiles' primary contribution was proving a crucial step of the very influential modularity conjecture, and the fact that it implies Fermat's Last Theorem is a simple (and mathematically relatively uninteresting!) corollary that ticked a big box in the history of mathematics.

i) *There are two people in Peterborough with the exact same number of hairs on their head.* Bald people exist, so this is obviously true for $n = 0$. Surprisingly, even when restricted to non-bald people, this is almost certainly a true statement: there are "only" 100-150 thousand hairs on a person's head on average, but the population of Peterborough is above 150,000 so by the Pigeonhole Principle it is (almost) impossible for everyone to have different numbers of hairs. This is in the category of "unexpected consequences of seemingly obvious theorems": results that are very easy to state and intuitively accept, but may often require some work and ingenuity to prove and

> have surprising implications. Another well-known example is the fact that there must be two opposite (antipodal) points on the surface of Earth with *exactly* the same temperature and pressure – this follows from the intermediate value theorem, which essentially states that if a continuous curve starts below a line and ends above the line, it must cross the line at some point. A similarly atmospheric example is the fact that there must exist a point on Earth without any blowing wind, as a consequence of the hairy ball theorem which can be summarised as "you can't comb a coconut". Note that all of these are *pure existence* theorems that do not provide a concrete witness for the existence: they don't explicitly name the two Peterborough residents or geographic coordinates where the statements hold, only that they must exist. One may reasonably say that existence should really be established by a *constructive* proof that either states the witness or gives an algorithm for computing it; *constructive mathematics* takes such algorithmic processes as the basis of proof theory and has surprising connections to computing (see the Part II Types course).

2. a) Write two closed statements (without variables): one true and one false.

   b) Write three statements with at least two variables: one valid, one satisfiable and one unsatisfiable. Give satisfying and falsifying interpretations where appropriate.

   c) Write a set S of at least three statements containing at least one (common) variable such that every statement is satisfiable, but the set is inconsistent.

2. Propositional logic

1. Verify de Morgan's laws and Peirce's Law using truth tables.
2. Each of the following formulas is satisfiable but not valid. Exhibit an interpretation that makes the formula true and another that makes the formula false.

$$P \to Q \qquad \neg(P \lor Q \lor R) \qquad P \lor Q \to P \land Q \qquad \neg(P \land Q) \land \neg(Q \lor R) \land (P \lor R)$$

3. Associate each of the following terms with one of the propositional equivalences on Slide 205. Give a number-theoretic example for each property.

   unit, distributivity, idempotence, commutativity, inverse, associativity, annihilation

4. Convert each of the following propositional formulas into Conjunctive Normal Form and also into Disjunctive Normal Form. For each formula, state whether it is valid, satisfiable, or unsatisfiable; justify each answer.

$$(P \to Q) \land (Q \to P) \qquad ((P \land Q) \lor R) \land \neg(P \lor R) \qquad \neg(P \lor Q \lor R) \lor ((P \land Q) \lor R)$$

3. Proof systems for propositional logic

1. Briefly compare and contrast the following three formal proof systems:

   - Hilbert-style deductive system

A formal proof system for propositional logic centred around the modus ponens logical deduction rule. Hilbert systems are very minimal, with implication as the only connective and a small number of axiom schemas (such as K, S, and DN) that can be instantiated with propositions and then combined using MP. Despite its simplicity, the Hilbert-style deduction is a sound and complete proof system for propositional logic: all the theorems it derives are (semantically) true, and it can derive any true proposition. The flip side is that the system is really difficult to use by hand and proving even the simplest propositions (such as $A \to A$) takes several nontrivial steps. The main issue is that we have no notion of hypothetical assumptions that can be used in the required places of the proof, so every step of a Hilbert proof is a tautology (i.e. a proposition that is true without any assumptions). Nevertheless, one may prove of any classical proposition *in principle*, and often that's all we need!

- Gentzen-style natural deduction

  Natural deduction is an improvement over Hilbert systems in that it is *natural*: logical connectives are treated on their own merit, and the logical rules correspond to strategies we would naturally use to use or prove a given connective. The emphasis is therefore on deduction rules, divided into *introduction* and *elimination* rules which construct or consume a proposition involving a logical connective. The only axiom is **t**, the introduction rule for truth. Writing a proof in natural deduction involves constructing a derivation tree rooted at the intended proposition, with all branches corresponding to a particular inference rule. For example, if we have a derivation of A and a derivation of B, we can combine the two into a derivation of $A \wedge B$.

  An added flexibility and complication is the use of hypothetical assumptions, which allow us to use a proposition as an "axiom" (a leaf node) as long as the derived formula is hypothesised on this assumption: the introduction rule for implication derives $A \to B$ from a derivation of B given a hypothetical assumption $[A]$. Similarly, the elimination form for disjunction expresses case analysis: if we can show C both by assuming $[A]$ and independently by assuming $[B]$, we can deduce C from $A \vee B$.

  Dealing with lots of hypothetical assumptions can get quite unwieldy since we need to keep track of when they are "in scope" and when the assumptions have already been consumed. An alternative presentation of natural deduction uses *hypothetical judgments* of the form $\Gamma \vdash A$, where A is the proposition to be proved and Γ is the set of assumptions we have "in scope" at a particular point of the proof. As we progress through the derivation, this context of assumptions varies as we add and discharge hypotheses; for example, if we derive $\Gamma, A \vdash B$ (assuming Γ and A, we can derive B), we can further derive $\Gamma \vdash A \to B$ by the introduction implication rule. Propositions proved in the empty context of assumptions $\vdash A$ are called *theorems*; the soundness and completeness of ND states that theorems and tautologies (true propositions for every interpretation of propositional letters) "coincide".

While natural deduction is indeed far more natural than Hilbert systems, using it to construct derivation trees is still difficult: the elimination rules seen "bottom-up" become introduction rules, and quite often involve inventing one or more propositions that get consumed at the application of the rule and therefore do not appear in the hypothesis. For example, modus ponens (implication introduction) derives B from $A \to B$ and A, so if we encounter a B in a bottom-up derivation, we may not have any way of knowing what hypothesis A to introduce in order to continue the subderivations. Even worse, there is nothing telling us that the last rule that must have been applied is modus ponens – it could well have been disjunction or conjunction elimination, all of which end with the derivation of an arbitrary formula. For this reason, natural deduction in the general case is not suitable for hand-written proofs or automatic proof search – there is an unbounded number of cases that could be considered.

- Gentzen-style sequent calculus

Gentzen's sequent calculus avoids the problem above by making every rule into an introduction rule – thus, sequent proofs have a clear bottom-up reading, each step eliminating one connective (thereby also bounding the size of the proof by the syntactic depth of the formula). Introducing connectives via *right* rules resembles natural deduction, but using assumptions is done via *left* rules rather than elimination. This necessitates generalising the hypothetical judgments $\Gamma \vdash A$ to *sequents* $\Gamma \Rightarrow \Delta$, where both Γ and Δ are sets of propositions. The interpretation of $\Gamma \Rightarrow \Delta$ is that assuming *all* the formulas in Γ implies *at least one* formula in Δ: $\bigwedge \Gamma \vDash \bigvee \Delta$. This duality in the sequents themselves leads to a very elegant and compact proof system that embraces the duality of propositional connectives: proving conjunction is similar to assuming a disjunction (a branching rule), assuming a conjunction is similar to proving a disjunction (turning the connective into a comma), and negation is self-dual. Implication may seem less intuitive: the right rule adds the hypothesis to the set of assumptions as expected, but the left rule splits it up in a somewhat backwards-looking way:

$$\frac{\Gamma \Rightarrow \Delta, A \qquad B, \Gamma \Rightarrow \Delta}{A \to B, \Gamma \Rightarrow \Delta} \ (\to l)$$

To see why it nevertheless makes sense (a.k.a. why the rule is *sound*), we need need to think about how an assumption $A \to B$ could be used in a proof. As a hypothetical statement, it is not immediately useful: "If I had a million pounds, I would move to Hawaii" doesn't give me a million pounds, and neither does it let me move anywhere. However, if the hypothesis is discharged, we can make use of the consequent in the rest of the proof – this is precisely what modus ponens says. Thus, if in addition to $A \to B$ we also had an assumption A, we could apply MP "behind the scenes" and make use of B. The only place we could get this extra A assumption from is from Γ, hence the first hypothesis of the rule; then, the B conclusion can be further assumed in the proof of Δ after the implicit application of MP. Of course, we should still allow the

option of proving Δ from Γ directly, and use the existing Γ assumptions to establish Δ, which justifies the principle of carrying Γ and Δ around in all the proofs.

Even though sequent calculus doesn't explicitly make use of elimination rules, they still lurk in the background of the right-side inferences. A similar analysis can be performed with the $\neg l$ rule: the negated assumption $\neg A$ is not necessarily useful until it can be confronted with an additional assumption A derived from Γ and the resulting contradiction is enough to deduce Δ. Sequent calculus rules are satisfyingly symmetric/dual, but they are interpreted as very different reasoning principles.

Remark. Note that the lecture slides use the notation $S \vdash A$ as the metatheoretic assertion that A is deducible from elements of S, and the course does not dwell on natural deduction long enough to introduce the $\Gamma \vdash A$ notation as the alternative for the bracketing and crossing out of hypotheses in natural deduction (and misleadingly suggests that an assumption environment is characteristic of the multiple-conclusion sequent calculus only). While it could be seen as a clash of notation, the two concepts really mean the same thing: A being syntactically derivable/provable from a set of propositions Γ is the same thing as the natural deduction judgment $\Gamma \vdash A$ with a finite derivation, which, in turn, is the same as a singleton-conclusion sequent $\Gamma \Rightarrow \{A\}$ also with a finite derivation. Thus, $\Gamma \vdash A$ can be seen as the "syntactic truth" assertion in contrast to the "semantic truth" $\Gamma \vDash A$, with soundness and completeness of propositional logic expressible as $\vdash A \leftrightarrow \vDash A$ for every formula A.

2. a) Proof systems employ many "if X then Y"-style connectives and assertions on different reasoning levels. Explain, with examples, the differences and similarities between

$$A \wedge B \to C \qquad A, B \vdash C \qquad A, B \vDash C \qquad A, B \Rightarrow C \qquad \frac{A \quad B}{C} \qquad \begin{array}{c}[A \wedge B] \\ \vdots \\ C\end{array}$$

Formal logic is a zoo of implication-like symbols and constructions that are quite easy to get confused about – and that is more-or-less the intention of this question!

$$A \wedge B \to C$$

This is a formula of propositional logic, and $\to$ is one of the *propositional connectives* declared in the syntax. Treated purely symbolically it does not denote or correspond to implication in the semantic (if $A \wedge B$ then C) sense, simply because symbols of the logic do not have an intrinsic meaning until such a meaning is assigned to them via truth tables. Indeed, there are several other symbols used for implication, such as $A \wedge B \Longrightarrow C$ and (especially in older texts) $A \wedge B \supset C$ – they're all different symbols that traditionally get interpreted as implication. In implementations the connective would correspond to a constructor of the datatype of propositional formulas.

$$A, B \Rightarrow C$$

The notation for *sequents* used in the course: a conditional assertion with some number of hypotheses, and some number of consequents. For the general sequent calculus both sides of the sequent are sets/sequences of any number of formulas. The $\Rightarrow$ symbol is therefore a metasyntactic connective between sets of formulas, acting as a separator between the assumptions and the conclusions. Once again, the distinction between assumptions and conclusions is merely our intuitive understanding of sequents; syntactically, all we care about is that $\Gamma, A \Rightarrow \Delta, A$ is an axiom (schema), and the sequent rules decompose and move symbols around the $\Rightarrow$ separator – "assuming" and "proving" is our human narrative to explain how and why the rules are sound, but they are nothing more than syntax tree transformations. Another way to appreciate this is to imagine implementing sequent calculus in a functional language: the sequent $\Rightarrow$ can simply be implemented as a pair of lists of formulas, without having to make any implementation distinction between the "hypothesis" list and the "consequent" list.

$$A, B \vdash C$$

A so-called *simple conditional assertion*, a special case of a sequent where the set of conclusions must have exactly one element: $\Gamma \vdash A$. While this particular notation is not used in the course (and, conversely, most other literature uses $\vdash$ for the general sequent rather than $\Rightarrow$), you've almost certainly encountered it in the IB Semantics course as a separator between the type environment Γ and type judgment $e : T$ where it serves a similar purpose: with the assumptions (variable typing environment) in Γ we have a finite derivation of A (a term e of type T). Such single-conclusion sequents usually feature in natural deduction systems where the formula on the right is operated on by introduction and elimination rules, while the formulas in Γ are left untouched (other than the set Γ itself evolving throughout the proof). The $\Gamma \vdash A$ notation is also used as a general statement of deducibility/derivability/provability/syntactic truth of a proposition A from a set of assumptions Γ but this essentially coincides with the analogous assertions in natural deduction and the sequent calculus (since provability is a statement of derivability in a particular proof system).

$$\begin{array}{c} [A \wedge B] \\ \vdots \\ C \end{array}$$

The original notation for *hypothetical assumptions* in Gentzen's natural deduction, analogous to the $\vdash$ presentation above. It suggests temporarily introducing an "axiom" $A \wedge B$ (i.e. a formula that needs no proof), and if from this we can derive C, we have a derivation of the implication $A \wedge B \to C$. The "temporariness" of the axiom is exhibited by having to cross it out as soon as the rule requiring it (e.g. $\to I$ or $\vee E$) is applied. In the conditional assertion presentation $\Gamma \vdash A \to B$, the introduction of a new "axiom" corresponds to extending Γ with a new assumption $\Gamma, A \vdash B$. Thus, whenever a proof of A is required in a further derivation, we may use the hypothesis $\Gamma, A \vdash A$. Of course,

a formula can only be considered a theorem if it has no assumptions, i.e. no uncrossed assumptions $[P]$ in the proof tree or, analogously, an empty assumption context $\emptyset \vdash A$.

$$\frac{A \quad B}{C}$$

An *inference rule* that is not specific to propositional logic, but used for inductive definitions of sets via rules and axioms. As such, it is also not about logical implication per se, but simply acts as a branch node for constructing derivation trees: if we have all the subtrees rooted the hypotheses of the rule (A and B, for example), we can create a new tree rooted at the conclusion of the rule C. Rules with no assumptions are simply the leaf nodes of the derivation tree – the axioms. Most formal systems and relations you encounter can be represented as inductively defined sets: syntax definitions (of programming languages, logics, regular expressions), typing rules, logical deduction rules, reduction rules, etc.

$$A, B \vDash C$$

The *semantic entailment* assertion which states that any interpretation that satisfies A and B (or a set Γ in general) also satisfies C. This is the only judgment that concerns the semantics of propositional logic by talking about interpretations of propositional letters and thus corresponds to the "actual" meaning of truth as defined by truth tables. A semantic entailment with no assumptions is known as a tautology: all interpretations satisfy the formula. A sound and complete proof system guarantees syntactic theorems $\vdash A$ exactly correspond to semantic tautologies $\vDash A$.

b) Separate the statements above based on whether they belong to the *syntax* or the *semantics* of propositional logic. Which proof systems are the constructs associated with?

Semantic entailment $\Gamma \vDash A$ is the only semantic assertion. The syntactic entailment $\Gamma \vdash A$ states that A is derivable from Γ in the particular proof system in question and is commonly used in the Hilbert-style (where Γ is always empty) and the Gentzen styles too. This course specifically uses $\Rightarrow$ for the multiple-conclusion sequent calculus. The implication $\rightarrow$ is defined in the syntax of propositional logic and is not tied to a particular proof system. Finally, the inference rule $\frac{A \quad B}{C}$ is a general tool for inductively defining proof systems and other formal relations.

3. Prove the following sequents:

$$\neg\neg A \Rightarrow A \qquad A \wedge B \Rightarrow B \wedge A \qquad (A \vee B) \wedge (A \vee C) \Rightarrow A \vee (B \wedge C)$$
$$\neg(A \vee B) \Rightarrow \neg A \wedge \neg B \qquad \Rightarrow (A \wedge \neg A) \rightarrow B \qquad \Rightarrow ((A \rightarrow B) \rightarrow A) \rightarrow A$$

(You can write the proof trees upside-down if you prefer, so you don't have to reserve space.)

There is some flexibility in the order of rule applications, but the general shape of the

trees will be similar.

$$\dfrac{\dfrac{\overline{A \Rightarrow A}}{\Rightarrow A, \neg A}\,(\neg r)}{\neg\neg A \Rightarrow A}\,(\neg l) \qquad \dfrac{\overline{A, B \Rightarrow B} \quad \overline{A, B \Rightarrow A}}{\dfrac{A, B \Rightarrow B \wedge A}{A \wedge B \Rightarrow B \wedge A}\,(\wedge l)}\,(\wedge r) \qquad \dfrac{\dfrac{\dfrac{\overline{A \Rightarrow B, A}}{A, \neg A \Rightarrow B}\,(\neg l)}{A \wedge \neg A \Rightarrow B}\,(\wedge l)}{\Rightarrow (A \wedge \neg A) \to B}\,(\to r)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{A \Rightarrow A, B}}{\Rightarrow \neg A, A, B}\,(\neg r) \quad \dfrac{\overline{B \Rightarrow A, B}}{\Rightarrow \neg B, A, B}\,(\neg r)}{\Rightarrow \neg A \wedge \neg B, A, B}\,(\wedge r)}{\Rightarrow \neg A \wedge \neg B, A \vee B}\,(\vee r)}{\neg(A \vee B) \Rightarrow \neg A \wedge \neg B}\,(\neg l) \qquad \dfrac{\dfrac{\dfrac{\overline{A \Rightarrow A, B}}{\Rightarrow A, A \to B}\,(\to r) \quad \overline{A \Rightarrow A}}{(A \to B) \to A \Rightarrow A}\,(\to l)}{\Rightarrow ((A \to B) \to A) \to A}\,(\to r)$$

$$\dfrac{\dfrac{\dfrac{\overline{A, A \vee C \Rightarrow A, B} \quad \overline{B, A \vee C \Rightarrow A, B}}{A \vee B, A \vee C \Rightarrow A, B}\,(\vee l) \quad \dfrac{\overline{A \vee B, A \Rightarrow A, C} \quad \overline{A \vee B, C \Rightarrow A, C}}{A \vee B, A \vee C \Rightarrow A, C}\,(\vee l)}{A \vee B, A \vee C \Rightarrow A, B \wedge C}\,(\wedge r)}{A \vee B, A \vee C \Rightarrow A \vee (B \wedge C)}\,(\vee r)}{(A \vee B) \wedge (A \vee C) \Rightarrow A \vee (B \wedge C)}\,(\wedge l)$$

4. Derive the sequent calculus rules for the connectives $\leftrightarrow$ (iff) and $\oplus$ (exclusive or). Note that other connectives must not appear in these rules.

The sequent calculus for the standard propositional connectives is sound and complete, which means that if a formula is derivable, so is any formula logically (that is, semantically) equivalent to it. We of course has the logical equivalence $A \leftrightarrow B \simeq (A \to B) \wedge (B \to A)$, and a derivation tree of the latter (with arbitrary sets of formulas Γ and Δ) on each side of a sequent would have the following form:

$$\dfrac{\dfrac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \to B}\,(\to r) \quad \dfrac{B, \Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, B \to A}\,(\to r)}{\Gamma \Rightarrow \Delta, (A \to B) \wedge (B \to A)}\,(\wedge r)$$

$$\dfrac{\dfrac{\dfrac{\Gamma \Rightarrow \Delta, A, B \quad \overline{A, \Gamma \Rightarrow \Delta, A}}{B \to A, \Gamma \Rightarrow \Delta, A}\,(\to l) \quad \dfrac{\overline{B, \Gamma \Rightarrow \Delta, B} \quad B, A, \Gamma \Rightarrow \Delta}{B, B \to A, \Gamma \Rightarrow \Delta}\,(\to l)}{A \to B, B \to A, \Gamma \Rightarrow \Delta}\,(\to l)}{(A \to B) \wedge (B \to A), \Gamma \Rightarrow \Delta}\,(\wedge l)$$

Thus, if bi-implication were in the sequent calculus, any application of $(\leftrightarrow l)$ or $(\leftrightarrow r)$ should be replaceable with these derived rules, since the two formulas are equivalent. But then we can just take these as the derived inference rules for $\leftrightarrow$ to begin with!

$$\dfrac{A, \Gamma \Rightarrow \Delta, B \quad B, \Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, A \leftrightarrow B}\,(\leftrightarrow r) \qquad \dfrac{\Gamma \Rightarrow \Delta, A, B \quad B, A, \Gamma \Rightarrow \Delta}{A \leftrightarrow B, \Gamma \Rightarrow \Delta}\,(\leftrightarrow l)$$

> These are sound by construction, since any derivation involving these rules should have a derivation involving $\to$ and $\wedge$ whose associated rules are sound.
>
> We can do something similar to derive exclusive or: we have the logical duality $A \oplus B \simeq \neg(A \leftrightarrow B)$, so just like with $\wedge$ and $\vee$, we expect the right rule of $\oplus$ to resemble the left rule of $\leftrightarrow$ and vice versa. This can of course be verified by a derivation of $(A \wedge \neg B) \vee (\neg A \wedge B)$.
>
> $$\frac{\Gamma \Rightarrow \Delta, A, B \qquad B, A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, A \oplus B}\ (\oplus r) \qquad \frac{A, \Gamma \Rightarrow \Delta, B \qquad B, \Gamma \Rightarrow \Delta, A}{A \oplus B, \Gamma \Rightarrow \Delta}\ (\oplus l)$$

4. First-order logic

1. To test your understanding of quantifiers, consider the following formulas: *everybody loves somebody* vs. *there is somebody that everybody loves*:

   $$\forall x. \exists y. \text{loves}(x, y) \qquad \exists y. \forall x. \text{loves}(x, y)$$

   Does the first imply the second? Does the second imply the first? Consider both the informal meaning and the formal semantics defined in the course.

2. Let $\approx$ be a 2-place predicate symbol, which we write using infix notation as $x \approx y$ instead of $\approx(x, y)$.

   a) Give three formulas in FOL describing the axioms that make $\approx$ an *equivalence relation*.

   > Reflexivity, symmetry and transitivity:
   >
   > Ⓡ $\forall x. x \approx x$ Ⓢ $\forall x, y. x \approx y \to y \approx x$ Ⓣ $\forall x, y, z. x \approx y \wedge y \approx z \to x \approx z$

   b) Let the universe be the set $\mathbb{N}$ of natural numbers. Which axioms hold if $I[\approx]$ is:

   (i) the empty relation, $\emptyset$?

   > Ⓢ and Ⓣ. Since the relation never holds, the hypotheses of symmetry and transitivity are never satisfied so they vacuously hold. Reflexivity requires at least the identity loops for all elements of the domain, but since the domain is nonempty, the empty relation is not reflexive.

   (ii) the universal relation, $\{(x, y) \mid x, y \in \mathbb{N}\}$?

   > Ⓡ, Ⓢ and Ⓣ. This is not to say the universal relation trivially satisfies everything – it violates negative properties like irreflexivity and asymmetry.

   (iii) the equality relation, $\{(x, x) \mid x \in \mathbb{N}\}$?

   > Ⓡ, Ⓢ and Ⓣ. Equivalence wouldn't really be a good generalisation of equality if equality wasn't an equivalence!

   (iv) the relation $\{(x, y) \mid x, y \in \mathbb{N} \wedge x + y$ is even$\}$?

> Ⓡ, Ⓢ and Ⓣ. Two numbers are related if both are even or both are odd – in other words, $x \approx y$ iff $x \equiv y \pmod 2$ and congruence is an equivalence relation.

(v) the relation $\{(x, y) \mid x, y \in \mathbb{N} \wedge x + y = 100\}$?

> Ⓢ. Not reflexive, since the only number that is related to itself is 50. Not transitive either: $30 \approx 70$ and $70 \approx 30$, but $30 \not\approx 30$.

(vi) the relation $\{(x, y) \mid x, y \in \mathbb{N} \wedge x \leq y\}$?

> Ⓡ and Ⓣ. $\leq$ is a partial order so it is reflexive and transitive; instead of symmetry it satisfies *antisymmetry*.

3. Taking R as 2-place relation symbol and $=$ denoting equality, consider the following axioms:

$$\forall x.\ \neg R(x, x) \tag{1}$$
$$\forall x, y.\ \neg(R(x, y) \wedge R(y, x)) \tag{2}$$
$$\forall x, y, z.\ R(x, y) \wedge R(y, z) \to R(x, z) \tag{3}$$
$$\forall x, y.\ R(x, y) \vee (x = y) \vee R(y, x) \tag{4}$$
$$\forall x, y.\ R(x, z) \to \exists y.\ R(x, y) \wedge R(y, z) \tag{5}$$

> These properties correspond to: irreflexivity (different from R just not being reflexive – we explicitly forbid any loops); asymmetry (again, different from both the negation of symmetry and antisymmetry); transitivity; linearity (every pair of elements is related in some way); density (for any two related elements there exists one between them).

a) Exhibit two interpretations that satisfy axioms 1–5.

> Interpret R as the ordering $<$ on rational or real numbers (or an interval thereof). The less-than ordering on numbers is irreflexive, asymmetric, transitive and linear in general, and density is ensured by considering a domain in which every nonempty interval has e.g. a halfway point.

b) Exhibit two interpretations that satisfy axioms 1–4 and falsify axiom 5.

> Without the need for density we can consider $<$ on domains such as naturals or integers, or other examples of strict total orders such as the strict lexicographical order on a finite Cartesian product of totally ordered sets, e.g. $\mathbb{N} \times \mathbb{N}$.

c) Exhibit two interpretations that satisfy axioms 1–3 and falsify axioms 4 and 5.

> Now we need a strict relation that can be partial, i.e. we can have two distinct elements which are not related to itself in any direction. The usual example is the strict subset order $\subset$, since for example $\{1\} \not\subset \{2\}$ and $\{2\} \not\subset \{1\}$. Another is strict divisibility on integers: $d \mid n \leftrightarrow \exists k \in \mathbb{Z}.\ k \neq 1 \wedge n = k \cdot d$.

Hint: Consider simple examples such as $R(x, y) = x < y$ on some appropriate domain.

4. Some textbook and paper authors like to use nonstandard notation, or even define their own syntax for some particular formal system. While often difficult to read and generally not recommended, there is nothing inherently wrong with this practice – as long as the meaning (semantics) is defined appropriately, the syntax can be whatever the author chooses to use.

   Suppose that a particularly creative logician decides to reinvent the syntax of first-order logic. Below are three logically valid formulas written in the syntax which should hold for any formula A and predicate P and Q containing a free variable x.

   $$\overbrace{\left[P(x), \boxed{P(x)}\right]}^{x} \qquad \boxed{A} \triangleright A \qquad \boxed{\overbrace{\left[P(x), \boxed{Q(x)}\right]}^{x}} \bowtie \underbrace{\left\langle \boxed{P(x)}, Q(x) \right\rangle}_{x}$$

   > The aim of this question is to demonstrate the decoupling of syntax and semantics in formal logics by asking you to give meaning to a notation that you have never seen before and therefore should not have any preconceptions about. A problem with studying logic at such a foundational level is that we humans are already used to standard notational conventions and it becomes difficult to see $\wedge$, $\rightarrow$ and $=$ as mere syntax and $A \wedge B \rightarrow C$ as a sequence or tree of symbols. As such, things like the truth definition of first-order logic may seem needlessly verbose and complicated while also "stating the obvious": $A \wedge B$ is true if A is true and B is true. Well, let us throw off the shackles of common sense notation and consider a syntax where things are not as obvious!

   a) Describe the formal syntax of this logic as a grammar, giving the different syntactic forms of a formula A (see e.g. Slide 201). You may assume that the syntax already includes atomic formulas made up of a relation symbol $P, Q, R \ldots$ applied to some number of terms.

      > We fix a set $\mathcal{L} = \{x, y, \ldots\}$ of variables. The syntax of the language can be read off without having to understand what the formulas mean. If A, B are formulas, so are
      >
      > $$\boxed{A} \qquad \langle A, B \rangle \qquad [A, B] \qquad A \triangleright B \qquad A \bowtie B \qquad \overbrace{A[x]}^{x} \qquad \underbrace{A[x]}_{x}$$

   b) Consider a formula A in the syntax presented above. Let $\mathcal{I}$ be an interpretation of the symbols and V a valuation for the formula. Give a *truth definition* for the formula A by defining a predicate $\vDash_{\mathcal{I},V} A$ which holds when A is true in $\mathcal{I}$ under V. Make sure that the three formulas above are true with your definition.

      > Here we need to perform some detective work to figure out what the connectives are actually supposed to mean. We know that all formulas are valid, i.e. satisfied by any interpretation. The most telling may be the second one, which suggests that two boxes around a formula are related to no boxes around a formula – it resembles double negation elimination, if box means negation and $\triangleright$ means implication or bi-

implication. With this interpretation of box, we see the first formula suggesting that boxed and unboxed versions of the same formula are somehow related; again, this resembles the law of excluded middle $A \vee \neg A$, assuming the square brackets denote disjunction. The overbrace surrounds the disjunction with a variable x so it is likely a binding construct. We can't in general say that there exists an x for which either $P(x)$ or not $P(x)$ hold since that wouldn't hold in the empty domain (as it at least requires the existence of an element); but we can say that if x is universally quantified. Finally, the last formula relates $\neg(\forall x.\ P(x) \vee \neg Q(x))$, and some combination of $\neg P(x)$ and $Q(x)$. It is not difficult to see that an application of the generalised and propositional de Morgan laws to the LHS gives $\exists x.\ \neg P(x) \wedge Q(x)$, so we can guess that the angle brackets mean conjunction and the underbrace is existential quantification. The $\bowtie$ connective may again mean implication or bi-implication – the formulas are valid with both interpretations of $\triangleright$ and $\bowtie$. We actually get to make a choice on how to interpret them, which is not something we usually get to do when the interpretations are "obvious"! Purely based on the symmetry of the $\bowtie$ symbol we can use it for bi-implication and $\triangleright$ for implication – perhaps this was not the intended interpretation of the logician, but it is a legitimate option based purely on the three valid formulas.

Now we actually get to give the truth definition for the language. It is entirely analogous to Tarski's definition, but with the modified syntax. The interpretation $\mathcal{I}$ and valuation V are required to interpret atomic formulae $P(a, f(x)), Q(g(y, b)), \ldots$ where we need interpretations for constants a, b, function symbols f, g, relation symbols P, Q, and variables x, y. The rest proceeds by recursing on the formula syntax, with the variable cases extending the valuation V with the newly bound variable.

For an interpretation $\mathcal{I} = (D, I)$ on domain D and valuation V, we define a formula A to be *true* in $\mathcal{I}$ under V, denoted $\vDash_{\mathcal{I},V}$, by case analysis on the syntax of formulas from part (a):

- $\vDash_{\mathcal{I},V} P(t_1, \ldots, t_n)$ if $I[P](\mathcal{I}_V[t_1], \mathcal{I}_V[t_n]) = 1$.
- $\vDash_{\mathcal{I},V} \boxed{A}$ if $\vDash_{\mathcal{I},V} A$ does not hold
- $\vDash_{\mathcal{I},V} \langle A, B \rangle$ if $\vDash_{\mathcal{I},V} A$ holds and $\vDash_{\mathcal{I},V} B$ holds
- $\vDash_{\mathcal{I},V} [A, B]$ if $\vDash_{\mathcal{I},V} A$ holds or $\vDash_{\mathcal{I},V} B$ holds
- $\vDash_{\mathcal{I},V} A \triangleright B$ if $\vDash_{\mathcal{I},V} A$ does not hold or $\vDash_{\mathcal{I},V} B$ holds
- $\vDash_{\mathcal{I},V} A \bowtie B$ if $\vDash_{\mathcal{I},V} A$ and $\vDash_{\mathcal{I},V} B$ both hold or neither hold
- $\vDash_{\mathcal{I},V} \overbrace{A[x]}^{x}$ if $\vDash_{\mathcal{I},V\{m/x\}} A$ holds for all $m \in D$
- $\vDash_{\mathcal{I},V} \underbrace{A[x]}_{x}$ if there exists some $m \in D$ such that $\vDash_{\mathcal{I},V\{m/x\}} A$ holds

c) The formula below is a statement about arithmetic on natural numbers. Give an interpretation $\mathcal{I} = (D, I)$ of the constant, function, and relation symbols, and a valuation V of the

free variables, which satisfy this formula.

$$\left\langle \overbrace{\left[n \sim \bullet, \underbrace{n \sim k^{+}}_{k} \right]}^{n}, \left\langle v \sim \bullet^{+++}, \boxed{\underbrace{\bullet \sim \ell^{+}}_{\ell}} \right\rangle \right\rangle$$

Above we gave a syntax and semantics for the formula language of this logic. The other variable aspect of first-order syntax is the term language, which makes up the syntactic entities that we can actually reason about. This follows naturally from the generalisation of propositional letters $P, Q, \ldots$ to predicates $P(a, f(x)), Q(g(y, b)), \ldots$ that now relate terms built from variables, constants and function symbols: if our logic involves variables, we need something that the variables can be instantiated with. Again, a difficulty with getting an intuition for the term syntax is that it is either entirely abstract, or too concrete and familiar: it's hard to see the point of saying "let 0 denote zero and $+$ denote addition and $=$ denote equality". Hence this question asks you to give an interpretation to symbols that you are not familiar with based entirely on a formula that the interpretation must satisfy.

The first step is rewriting the formula in the familiar syntax of FOL:

$$\left(\forall n.\, (n \sim \bullet) \vee \exists k.\, n \sim k^{+}\right) \;\wedge\; (v \sim \bullet^{+++}) \;\wedge\; \left(\neg \exists \ell.\, \bullet \sim \ell^{+}\right)$$

Next, we analyse the arities of the various symbols featuring in the formula, based on how they are notated: $\bullet$ is a constant symbol, $(-)^{+}$ is a unary function symbol written in superscript, $\sim$ is a binary relation symbol written infix (it can't be a function symbol since $(v \sim \bullet^{+++})$ is used as a formula). Now, we know that the formula expresses a property of arithmetic on natural numbers, and with a bit of trial and error it seems likely that $\bullet$ denotes 0, $(-)^{+}$ denotes successor and $\sim$ is equality. The first conjunct states that any natural number n is either zero or is the successor of another natural number, and the third conjunct states that there does not exist a number ℓ whose successor is zero. These are axioms of Peano arithmetic. The middle conjunct involves a free variable v and simply states that it is equal to 3 (the third successor of 0) – the only valuation that will satisfy the formula is one that maps v to 3. Thus, we define the interpretation and valuation as follows:

$$D = \mathbb{N} \qquad V = v \mapsto 3$$

$$I[\bullet] = 0 \in \mathbb{N} \qquad I[(-)^{+}] = n \mapsto n + 1 \in \mathbb{N} \to \mathbb{N} \qquad I[\sim] = \{(n, m) \mid n = m\} \subseteq \mathbb{N}^2$$

This exercise hopefully gave you a better grasp of symbols, first-order formulas, interpretations and truth definitions, and how syntax of formulas and terms really isn't constrained by anything other than notational conventions. This of course doesn't mean that you should question the semantics of symbols every time you encounter

them ("Excuse me, what is the intended interpretation of the symbols '£3.50' on that price list?") but when studying formal systems one must remember not to make assumptions about the syntax based on perceived semantics and accidentally muddle the distinction between the two. For example, it may be tempting to apply a logical equivalence in the middle of a sequent proof e.g. to reparenthesise a nested conjunction from $\Gamma \Rightarrow \Delta, (A \wedge B) \wedge C$ to $\Gamma \Rightarrow \Delta, A \wedge (B \wedge C)$ in order to get access to A right away. The step may seem innocuous (after all, conjunction *is* associative) but crucially it is an application of a semantic property in a purely syntactic proof and assumes that the proof system is sound and complete, and $\wedge$ actually stands for an associative operator. From the point of view of syntax, $\wedge$ is merely a branch node in the abstract syntax tree of propositional logic and the tree of $(A \wedge B) \wedge C$ is not interchangeable with the tree of $A \wedge (B \wedge C)$. If we did something similar in the syntax above to rewrite $(A \triangleright B) \triangleright C$ as $A \triangleright (B \triangleright C)$, we would clearly be making a mistake! You may of course say "well I obviously won't do that because I know that implication is not associative" but again, that comes from a semantic understanding of the symbol $\triangleright$; an automated theorem prover searching for a sequent calculus proof of the formula will have no idea what it's doing other than manipulating syntax trees according to strictly predetermined rules.

To take this to its logical conclusion, there is nothing stopping us from completely butchering the syntax of first-order logic by mixing up all the symbols:

$$(\rightarrow n.\,(n\,(+1)\,=)\,\exists \wedge k.\,n\,(+1)\,0k)\;\neg\;(v\,(+1)\,000 =)\;\neg\;(\forall \wedge \ell.\,=\,(+1)\,0\ell)$$

Silly as this may look, it has a perfectly consistent formula syntax and interpretation, e.g. with $=$ interpreted as zero and $\neg$ as disjunction, but of course it completely violates all notational conventions and could very much be described as *cursed*. Deciphering this was actually the original iteration of the question, but it seemed less cryptic to use more cryptic symbols instead!

Optional exercise

1. Using OCaml, define datatypes for representing propositions and interpretations. Write a function to test whether or not a proposition holds under an interpretation (both supplied as arguments). Write a function to convert a proposition to Negation Normal Form.

5. Formal reasoning in first-order logic

1. a) For each of the following FOL formulas, circle the free variables, and underline the bound variables. Connect each bound variable to its binding occurrence (either graphically, by numbering, or whatever works for you).

   i. $\forall x.\, x = x$

   ii. $\exists x.\, P(x, y) \wedge \forall y.\, \neg P(y, x)$

   iii. $(\forall x.\, P(x, y)) \rightarrow (\exists y.\, P(x, y))$

   iv. $\exists z.\, P(x, y) \wedge Q(x, z)$

   v. $\forall x.\, (P(x) \rightarrow Q(x)) \wedge S(x, y)$

   vi. $P(x, \forall z.\, (\exists x.\, Q(y, x)) \rightarrow Q(x, z))$

> The scope of the binding extends as far after the dot as possible, within the limits of any outside parentheses.
>
> i. $\forall x.\ \underline{x} = \underline{x}$
>
> ii. $\exists x.\ P(\underline{x}, \textcircled{y}) \land \forall y.\ \neg P(\underline{y}, \underline{x})$
>
> iii. $\Big(\forall x.\ P(\underline{x}, \textcircled{y})\Big) \to \Big(\exists y.\ P(\textcircled{x}, \underline{y})\Big)$
>
> iv. $\exists z.\ P(\textcircled{x}, \textcircled{y}) \land Q(\textcircled{x}, \underline{z})$
>
> v. $\forall x.\ (P(\underline{x}) \to Q(\underline{x})) \land S(\underline{x}, \textcircled{y})$
>
> vi. $P\Big(\textcircled{x}, \forall z.\ (\exists x.\ Q(\textcircled{y}, \underline{x})) \to Q(\textcircled{x}, \underline{z})\Big)$

b) Apply the substitution $[x \mapsto f(x, y), y \mapsto g(z)]$ to the formulas above.

> Substitution occurs only for *free* variables. Occasionally we need to α-rename the bound variables to avoid variable capture: free variables in terms to be substituted must remain free.
>
> - $\forall x.\ \underline{x} = \underline{x}$ (all variables are bound, so no substitution happens)
> - $\exists x.\ P(\underline{x}, g(\textcircled{z})) \land \forall y.\ \neg P(\underline{y}, \underline{x})$
> - $\Big(\forall x.\ P(\underline{x}, g(\textcircled{z}))\Big) \to \Big(\exists w.\ P(f(\textcircled{x}, \textcircled{y}), \underline{w})\Big)$ (y has to be renamed)
> - $\exists w.\ P(f(\textcircled{x}, \textcircled{y}), g(\textcircled{z}) \land Q(f(\textcircled{x}, \textcircled{y}), \underline{w})$ (z has to be renamed)
> - $\forall x.\ (P(\underline{x}) \to Q(\underline{x})) \land S(\underline{x}, g(\textcircled{z}))$
> - $P\Big(f(\textcircled{x}, \textcircled{y}), \forall w.\ (\exists x.\ Q(g(\textcircled{z}), \underline{x})) \to Q(f(\textcircled{x}, \textcircled{y}), \underline{w})\Big)$ (z renamed)

2. Consider the following proof attempt of the set-theoretic conjecture:

$$\forall A, B.\ \ \mathcal{P}(A \cup B) = \mathcal{P}(A) \cup \mathcal{P}(B)$$

Proof. Let A, B be sets. We prove the proposition using equational reasoning:

$$\begin{aligned}
\mathcal{P}(A \cup B) &= \{X \mid X \subseteq A \cup B\} && \text{(def. of powerset)} \\
&= \{X \mid \forall x.\ x \in X \to x \in A \cup B\} && \text{(def. of subsets)} \\
&= \{X \mid \forall x.\ x \in X \to (x \in A \lor x \in B)\} && \text{(def. of union)} \\
&= \{X \mid \forall x.\ (x \in X \to x \in A) \lor (x \in X \to x \in B)\} && \text{(lemma of prop. logic)} \\
&= \{X \mid (X \subseteq A) \lor (X \subseteq B)\} && \text{(def. of subsets)} \\
&= \{X \mid X \subseteq A\} \cup \{X \mid X \subseteq B\} && \text{(def. of union)} \\
&= \mathcal{P}(A) \cup \mathcal{P}(B) && \text{(def. of powerset)}
\end{aligned}$$

□

Is this a valid proof? Why or why not?

> This is not a valid proof, and no "proof" of it may be correct since the theorem itself is false. Most of the steps are actually correct, and the mistake is somewhat hidden. The problem is going from
>
> $$\forall x.\ (x \in X \to x \in A) \lor (x \in X \to x \in B) \tag{1}$$
>
> to
>
> $$(X \subseteq A) \lor (X \subseteq B) \tag{2}$$

Even though the definition of the subset relation $S \subseteq A$ is $\forall x.\ x \in S \to x \in A$ and we have two similar implications in step (1), we would actually need an intermediate isomorphism

$$(\forall x.\ x \in X \to x \in A) \lor (\forall x.\ x \in X \to x \in B)$$

to be able to get to (2), from which the rest of the proof would follow. However, universal quantification *does not* distribute over disjunction, only conjunction:

$$\forall x.\ P(x) \lor Q(x) \not\simeq (\forall x.\ P(x)) \lor (\forall y.\ Q(y))$$

Dually, existential quantification distributes over disjunction, but does not distribute over conjunction. These properties follow from the interpretation of universal quantification as a generalised conjunction, and existential quantification as a generalised disjunction.

3. Verify the following equivalences by appealing to the truth definition of FOL.

$$\neg(\exists x.\ P(x)) \simeq \forall x.\ \neg P(x) \qquad (\forall x.\ P(x)) \land R \simeq \forall x.\ (P(x) \land R)$$
$$(\exists x.\ P(x)) \lor (\exists x.\ Q(x)) \simeq \exists x.\ (P(x) \lor Q(x))$$

While semantic equivalences in propositional logic can be established by comparing truth tables (see Ex. 2.1), this is not possible in FOL due to the presence of quantifiers and variables. The associated notion of semantic truth is therefore more involved, making use of domains, interpretations, valuations, and a structural inductive definition of the truth of FOL formulae. But, in essence, we interpret a FOL formula as the intuitive English translation for what it says, and establish equivalences via this interpretation.

Assume a domain D and interpretation I. Two formulae A and B are equivalent, if $\vDash_{\mathcal{I},V} A$ holds if and only if $\vDash_{\mathcal{I},V} B$ holds for some valuation V.

- $\neg(\exists x.\ P(x)) \simeq \forall x.\ \neg P(x)$: The LHS holds if there does not exist an element $d \in D$ such that $I[P](d) = 1$. The RHS holds if for all $d \in D$, $I[P](d) = 1$ does not hold – that is, a d for which $I[P](d) = 1$ holds cannot exist, which is precisely the LHS.
- $(\forall x.\ P(x)) \land R \simeq \forall x.\ (P(x) \land R)$: Both sides are true if and only if $I[R] = 1$ and $I[P](d) = 1$ for all $d \in D$.
- $(\exists x.\ P(x)) \lor (\exists x.\ Q(x)) \simeq \exists x.\ (P(x) \lor Q(x))$: Both sides are true if and only if there exists an element $d \in D$ such that either $I[P](d) = 1$ or $I[Q](d) = 1$.

4. Prove the equivalence $(\forall x.\ P(x) \lor P(a)) \simeq P(a)$.

There are several approaches. We can use equivalence reasoning to show that

$$(\forall x.\ P(x) \lor P(a)) \longleftrightarrow P(a)$$

is a theorem of FOL. Alternatively, we can make use of the fact that the sequent calculus is a sound and complete proof system for FOL, so we can also establish the sequents $(\forall x.\ P(x) \lor P(a)) \Rightarrow P(a)$ and $P(a) \Rightarrow (\forall x.\ P(x) \lor P(a))$. We can even mix the two techiques: rewrite the LHS using the FOL equivalence $\forall x.\ P(x) \lor P(a) \simeq (\forall x.\ P(x)) \lor P(a)$,

then do two sequent proofs:

$$\dfrac{\dfrac{\overline{P(a) \Rightarrow P(a)}}{\forall x.\, P(x) \Rightarrow P(a)}\,(\forall l, a/x) \qquad \dfrac{\overline{P(a) \Rightarrow P(a)}}{(\forall x.\, P(x)) \vee P(a) \Rightarrow P(a)}\,(\vee l)}{} \qquad \dfrac{\overline{P(a) \Rightarrow \forall x.\, P(x), P(a)}}{P(a) \Rightarrow (\forall x.\, P(x)) \vee P(a)}\,(\vee r)$$

5. Prove the following sequents. *Hint*: the last one requires two uses of $(\forall l)$.

$$(\forall x.\, P(x)) \wedge (\forall x.\, Q(x)) \Rightarrow \forall y.\, (P(y) \wedge Q(y))$$
$$\forall x.\, P(x) \wedge Q(x) \Rightarrow (\forall y.\, P(y)) \wedge (\forall y.\, Q(y))$$
$$\forall x.\, P(x) \to P(f(x)), P(a) \Rightarrow P(f(f(a)))$$

These proofs are largely straightforward. Some re-ordering of the steps is allowed, though not of the quantifier inferences.

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{P(y), \forall x.\, Q(x) \Rightarrow P(y)}}{\forall x.\, P(x), \forall x.\, Q(x) \Rightarrow P(y)}\,(\forall l, y/x) \qquad \dfrac{\overline{\forall x.\, P(x), Q(y) \Rightarrow Q(y)}}{\forall x.\, P(x), \forall x.\, Q(x) \Rightarrow Q(y)}\,(\forall l, y/x)}{\forall x.\, P(x), \forall x.\, Q(x) \Rightarrow P(y) \wedge Q(y)}\,(\wedge r)}{\forall x.\, P(x), \forall x.\, Q(x) \Rightarrow \forall y.\, P(y) \wedge Q(y)}\,(\forall r)}{\forall x.\, P(x) \wedge \forall x.\, Q(x) \Rightarrow \forall y.\, P(y) \wedge Q(y)}\,(\wedge l)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{P(y), Q(y) \Rightarrow P(y)}}{P(y) \wedge Q(y) \Rightarrow P(y)}\,(\wedge l)}{\forall x.\, P(x) \wedge Q(x) \Rightarrow P(y)}\,(\forall l, y/x)}{\forall x.\, P(x) \wedge Q(x) \Rightarrow \forall y.\, P(y)}\,(\forall r) \qquad \dfrac{\dfrac{\dfrac{\overline{P(y), Q(y) \Rightarrow Q(y)}}{P(y) \wedge Q(y) \Rightarrow Q(y)}\,(\wedge l)}{\forall x.\, P(x) \wedge Q(x) \Rightarrow Q(y)}\,(\forall l, y/x)}{\forall x.\, P(x) \wedge Q(x) \Rightarrow \forall y.\, Q(y)}\,(\forall r)}{\forall x.\, P(x) \wedge Q(x) \Rightarrow (\forall y.\, P(y)) \wedge (\forall y.\, Q(y))}\,(\wedge r)$$

We will write $f^2(a)$ for $f(f(a))$ and Γ for $\forall x.\, P(x) \to P(f(x))$ to save space, and also omit some parentheses.

$$\dfrac{\dfrac{\overline{P(a) \Rightarrow P(a), P(f^2a)} \qquad \dfrac{\dfrac{\overline{P(fa), P(a) \Rightarrow P(fa), P(f^2a)} \qquad \overline{P(f^2a), P(fa), P(a) \Rightarrow P(f^2a)}}{P(fa) \to P(f^2a), P(fa), P(a) \Rightarrow P(f^2a)}\,(\to l)}{\forall x.\, P(x) \to P(fx), P(fa), P(a) \Rightarrow P(f^2a)}\,(\forall l, fa/x)}{\forall x.\, P(x) \to P(fx), P(a) \to P(fa), P(a) \Rightarrow P(f^2a)}\,(\to l)}{\forall x.\, P(x) \to P(fx), P(a) \Rightarrow P(f^2a)}\,(\forall l, a/x)$$

6. Prove the following sequents. *Hint*: the last one requires two uses of $(\exists r)$.

$$P(a) \vee \exists x.\, P(f(x)) \Rightarrow \exists y.\, P(y)$$
$$\exists x.\, P(x) \vee Q(x) \Rightarrow (\exists y.\, P(y)) \vee (\exists y.\, Q(y))$$
$$\Rightarrow \exists z.\, P(z) \to P(a) \wedge P(b)$$

Once again, some reordering of steps is possible, but the quantifier steps must be done in the other shown since otherwise the constraint on $(\exists l)$ is likely to be violated.

$$\dfrac{\dfrac{\overline{P(a) \Rightarrow P(a)}}{P(a) \Rightarrow \exists y.\, P(y)}\,(\exists r, a/y) \qquad \dfrac{\dfrac{\overline{P(f(x)) \Rightarrow P(f(x))}}{P(f(x)) \Rightarrow \exists y.\, P(y)}\,(\exists r, f(x)/y)}{\exists x.\, P(f(x)) \Rightarrow \exists y.\, P(y)}\,(\exists l)}{P(a) \vee \exists P.\, (f(x)) \Rightarrow \exists y.\, P(y)}\,(\vee l)$$

$$\dfrac{\dfrac{\dfrac{\overline{P(x) \Rightarrow P(x), \exists y.\, Q(y)}}{P(x) \Rightarrow \exists y.\, P(y), \exists y.\, Q(y)}\,(\exists r, x/y) \qquad \dfrac{\overline{Q(x) \Rightarrow \exists y.\, P(y), Q(x)}}{Q(x) \Rightarrow \exists y.\, P(y), \exists y.\, Q(y)}\,(\exists r, x/y)}{P(x) \vee Q(x) \Rightarrow \exists y.\, P(y), \exists y.\, Q(y)}\,(\vee l)}{P(x) \vee Q(x) \Rightarrow \exists y.\, P(y) \vee \exists y.\, Q(y)}\,(\vee r)}{\exists x.\, P(x) \vee Q(x) \Rightarrow (\exists y.\, P(y)) \vee (\exists y.\, Q(y))}\,(\exists l)$$

In the proof below, the right-hand side $P(a) \wedge P(b), P(a) \wedge P(b)$ can be collapsed to the single formula $P(a) \wedge P(b)$. This makes sense because we regard a sequent as a set of formulas, and the transformation is sound due to the idempotence of disjunction.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{P(a), P(b) \Rightarrow P(a)} \qquad \overline{P(a), P(b) \Rightarrow P(b)}}{P(a), P(b) \Rightarrow P(a) \wedge P(b), P(a) \wedge P(b)}\,(\wedge r)}{P(a) \Rightarrow P(a) \wedge P(b), P(b) \to P(a) \wedge P(b)}\,(\to r)}{P(a) \Rightarrow P(a) \wedge P(b), \exists z.\, P(z) \to P(a) \wedge P(b)}\,(\exists r, b/z)}{\Rightarrow P(a) \to P(a) \wedge P(b), \exists z.\, P(z) \to P(a) \wedge P(b)}\,(\to r)}{\Rightarrow \exists z.\, P(z) \to P(a) \wedge P(b)}\,(\exists r, a/z)$$

7. Prove the formula $\neg \forall y.\, (Q(a) \vee Q(b)) \wedge \neg Q(y)$ using equivalences, and then formally using the sequent calculus.

The semantic proof relies on standard FOL equivalences, including the *expansion* rules which let us extract instances of quantified formulae with the appropriate connective (conjunction for $\forall$, disjunction for $\exists$).

$$\begin{aligned}
& \neg \forall y.\, (Q(a) \vee Q(b)) \wedge \neg Q(y) \\
\simeq\ & \exists y.\, \neg((Q(a) \vee Q(b)) \wedge \neg Q(y)) && \text{(generalised de Morgan)} \\
\simeq\ & \exists y.\, \neg(Q(a) \vee Q(b)) \vee \neg\neg Q(y) && \text{(propositional de Morgan)} \\
\simeq\ & \exists y.\, \neg(Q(a) \vee Q(b)) \vee Q(y) && \text{(double negation elimination)} \\
\simeq\ & \neg(Q(a) \vee Q(b)) \vee \exists y.\, Q(y) && \text{(reduce scope of existential)} \\
\simeq\ & \neg(Q(a) \vee Q(b)) \vee Q(a) \vee Q(b) \vee \exists y.\, Q(y) && \text{(expansion twice)} \\
\simeq\ & \top \vee \exists y.\, Q(y) && \text{(law of excluded middle)} \\
\simeq\ & \top && \text{(truth is annihilator for disjunction)}
\end{aligned}$$

The sequent calculus proof is as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Q(a), \neg Q(b) \Rightarrow Q(a)}}{Q(a), \neg Q(b), \neg Q(a) \Rightarrow}(\neg l) \quad \dfrac{\dfrac{}{Q(b), \neg Q(a) \Rightarrow Q(b)}}{Q(b), \neg Q(a), \neg Q(b) \Rightarrow}(\neg l)}{Q(a) \vee Q(b), \neg Q(a), \neg Q(b) \Rightarrow}(\vee l)}{Q(a) \vee Q(b), \neg Q(a), (Q(a) \vee Q(b)) \wedge \neg Q(b) \Rightarrow}(\wedge l)}{(Q(a) \vee Q(b)) \wedge \neg Q(a), (Q(a) \vee Q(b)) \wedge \neg Q(b) \Rightarrow}(\wedge l)}{(Q(a) \vee Q(b)) \wedge \neg Q(a), \forall y.\, (Q(a) \vee Q(b)) \wedge \neg Q(y) \Rightarrow}(\forall l, b/y)}{\forall y.\, (Q(a) \vee Q(b)) \wedge \neg Q(y) \Rightarrow}(\forall l, a/y)}{\Rightarrow \neg \forall y.\, (Q(a) \vee Q(b)) \wedge \neg Q(y)}(\neg r)$$

6. Clause methods for propositional logic

1. Outline the steps of the Davis–Putnam–Logeman–Loveland method. Explain the goal of the method, and why the steps of the algorithm are sound. Why does the empty clause represent a contradiction?

DPLL is an algorithm for deciding the satisfiability of propositional formulae in clausal (conjunctive normal) form. Given a formula presented as a set of clauses, DPLL can either determine that the formula is not satisfiable, or return a satisfying model if it is. DPLL can also be used to determine if a formula is valid (i.e. satisfied by every interpretation) by negating it first and trying to derive a contradiction: if DPLL reaches the contradiction, there does not exist a falsifying interpretation of the original formula, so it must be valid. The contradiction is represented by the *empty clause* $\{\ \}$: since a clause $\{P_1, \ldots, P_n\}$ stands for a disjunction $P_1 \vee \cdots \vee P_n$, and the disjunction of no formulae is false, deriving the empty clause is equivalent to deriving falsity from the clauses.

The steps of the DPLL algorithm to establish the validity of a formula φ are as follows.

a) Negate φ and convert it to conjunctive normal form, then clauses.

b) Delete *tautological clauses* $\{P, \neg P, \ldots\}$. These are true for any interpretation of P, since the conjunct will contain $P \vee \neg P \simeq \top$.

c) *Unit propagation*: for every *unit clause* containing a single literal $\{L\}$ (where $L = P$ or $L = \neg P$ for some P), delete every clause containing L, and delete $\neg L$ from every remaining clause. This is sound since any model of the clauses must assign true to L (otherwise the corresponding conjunct would be false); any clause containing L will therefore be true, and the negation of L (which will be assigned $\bot$) will be a neutral element in the remaining clauses.

d) *Pure literal elimination*: remove every clause containing a *pure literal*, i.e. a literal L for which there is no clause containing $\neg L$. This is sound since we can always assign pure literals to be true (it cannot lead to a contradiction, since there is no clause containing their negation), so all clauses containing them will be automatically satisfied.

e) If an empty clause is reached, we reached a contradiction. Conversely, if all clauses are deleted, the original clause set is satisfiable, and the model can be determined from the assignments performed in the unit propagation and pure literal elim. steps.

f) Otherwise (no unit clauses, no pure literals, nonempty clauses left), choose a literal L to case-split on, and recursively apply the algorithm to the L and $\neg L$ subcases. The clause set is satisfiable if and only if one of the subcases is satisfiable. Since we exhaustively check both cases, this step is also sound.

2. Apply the DPLL procedure to the clause set:

$$\{P, Q\} \qquad \{\neg P, Q\} \qquad \{P, \neg Q\} \qquad \{\neg P, \neg Q\}$$

We cannot perform unit propagation or pure literal elimination, so we need to start with a case-split. Choosing P, we get:

- If P is true, we can unit-propagate it to get the clauses $\{Q\}, \{\neg Q\}$ which give the empty clause by unit-propagating Q.
- If P is false, we again get the clauses $\{Q\}, \{\neg Q\}$ and derive a contradiction.

3. Explain the resolution algorithm and how it differs from DPLL.

The *resolution rule* is a rule inference that gives rise to a refutation theorem-proving algorithm for propositional (and first-order) logic. The clausal resolution step combines two clauses containing *complementary literals*: the clauses $\{\Gamma, P\}$ and $\{\neg P, \Delta\}$ can be resolved into the single clause $\{\Gamma, \Delta\}$, where Γ and Δ stand for an arbitrary sequence of literals. The intuition behind the resolution step is nothing more than gluing together implications: the clause $\{\Gamma, P\}$ is equivalent to $\neg\Gamma \to P$, $\{\neg P, \Delta\}$ is equivalent to $P \to \Delta$, and the resolution step corresponds to the conclusion $\neg\Gamma \to \Delta$ which follows from the transitivity of $\to$.

$$\frac{\{A_1, \ldots, A_m, B\} \qquad \{\neg B, C_1, \ldots, C_n\}}{\{A_1, \ldots, A_m, C_1, \ldots, C_n\}} \text{ (res B)}$$

The resolution algorithm is based on proof by contradiction, so the original formula has to be negated before converting into clausal form. The rule is repeatedly applied to the clauses until the empty clause is reached; if we haven't reached the empty clause and cannot apply any more resolution steps, the original set of formulae is satisfiable. Resolution is *complete*, so if a set of clauses is unsatisfiable, resolution will be able to derive a contradiction (i.e. it won't get "stuck").

DPLL and resolution are not interchangeable and should not be mix-and-matched. DPLL *modifies* and *consumes* clauses: unit propagation and pure literal elimination removes literals from clauses and deletes the clauses themselves, until the empty clause is created or all clauses are deleted. Resolution simply *combines* clauses by making transitive inferences and adding the resolvent to the clause set: the existing clauses are not modified or removed, and one clause can be used several times. DPLL can be used both to find models of a

clause set (see SMT solvers), or perform refutation proofs of validity. Resolution is tailored to refutation proofs: getting "stuck" in a resolution proof does not give us a model, and we can never derive the empty clause set, since resolution does not decrease the number of clauses. The main advantage of resolution is that it is very simple to automate and can be made efficient with heuristics for choosing the pair of clauses to resolve.

4. Use resolution (showing the steps of converting the formula into clauses) to prove at least three of the following formulas.

$$(P \to Q \lor R) \to ((P \to Q) \lor (P \to R))$$
$$((P \to Q) \to P) \to P$$
$$(Q \to R) \land (R \to P \land Q) \land (P \to Q \lor R) \to (P \leftrightarrow Q)$$
$$(P \land Q \to R) \land (P \lor Q \lor R) \to ((P \leftrightarrow Q) \to R)$$
$$(P \to R) \land (R \land P \to S) \to (P \land Q \to R \land S)$$

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

When negating a formula, it's useful to remember some equivalences to avoid having to do a lot of unnecessary work. The most useful one is $\neg(P \to Q) \simeq P \land \neg Q$, so when negating any implication, the hypothesis can be immediately read off as a clause (without negation, but perhaps with some rearranging). Thus, in the first formula, $(P \to Q \lor R)$ gives one of our clauses, then $\neg((P \to Q) \lor (P \to R)) \simeq (P \land \neg Q) \land (P \land \neg R)$ gives the rest:

$$① \{\neg P, Q, R\} \quad ② \{P\} \quad ③ \{\neg Q\} \quad ④ \{\neg R\}$$

$$① + ② \overset{P}{\Rightarrow} ⑤ \{Q, R\} \qquad ⑤ + ③ \overset{Q}{\Rightarrow} ⑥ \{R\} \qquad ⑥ + ④ \overset{R}{\Rightarrow} \square$$

b) $((P \to Q) \to P) \to P$

Negate and convert to clauses:

$$\begin{aligned}\neg((P \to Q) \to P) \to P &\simeq ((P \to Q) \to P) \land \neg P \simeq (\neg(P \to Q) \lor P) \land \neg P \\ &\simeq ((P \land \neg Q) \lor P) \land \neg P \simeq (P \lor P) \land (\neg Q \lor P) \land \neg P\end{aligned}$$

$$① \{P\} \quad ② \{\neg Q, P\} \quad ③ \{\neg P\}$$

A single resolution step on ① and ③ derives $\square$, so the original formula (Peirce's law) is valid.

c) $(Q \to R) \land (R \to P \land Q) \land (P \to Q \lor R) \to (P \leftrightarrow Q)$

Negate and convert to clauses. Note how negation only affects the consequent $P \leftrightarrow Q$; the hypotheses can readily be read off as clauses:

- $Q \to R$ gives $\{\neg Q, R\}$

- $R \to (P \wedge Q) \simeq (R \to P) \wedge (R \to Q)$ gives $\{\neg R, P\}$ and $\{\neg R, Q\}$
- $P \to Q \vee R$ gives $\{\neg P, Q, R\}$

For the negated conclusion, convert $\neg(P \leftrightarrow Q)$ into clauses:

$$\neg(P \leftrightarrow Q) \simeq P \leftrightarrow \neg Q \simeq (P \to \neg Q) \wedge (\neg Q \to P) \simeq (\neg P \vee \neg Q) \wedge (Q \vee P)$$

$$① \{\neg Q, R\} \quad ② \{\neg R, P\} \quad ③ \{\neg R, Q\} \quad ④ \{\neg P, Q, R\} \quad ⑤ \{\neg P, \neg Q\} \quad ⑥ \{P, Q\}$$

One possible sequence of resolution steps is below. This is an example of *linear resolution*, where the output of one resolution step is the input to the next one – it can also be represented as a left-branching tree.

$$① + ② \overset{R}{\Rightarrow} \{\neg Q, P\} + ⑥ \overset{Q}{\Rightarrow} \{P\} + ④ \overset{P}{\Rightarrow} \{Q, R\}$$
$$+ ③ \overset{R}{\Rightarrow} \{Q\} + ⑤ \overset{Q}{\Rightarrow} \{\neg P\} + \{P\} \overset{P}{\Rightarrow} \square$$

d) $(P \wedge Q \to R) \wedge (P \vee Q \vee R) \to ((P \leftrightarrow Q) \to R)$

Negate and convert to clauses. Again, a lot of the formula goes untouched; we read off the following hypotheses:

- $P \wedge Q \to R$ gives $\{\neg P, \neg Q, R\}$
- $P \vee Q \vee R$ gives $\{P, Q, R\}$
- $P \leftrightarrow Q \simeq (P \to Q) \wedge (Q \to P)$ gives $\{\neg P, Q\}, \{P, \neg Q\}$

Finally, with the negated conclusion $\neg R$, the clause set is:

$$① \{\neg P, \neg Q, R\} \quad ② \{P, Q, R\} \quad ③ \{\neg P, Q\} \quad ④ \{P, \neg Q\} \quad ⑤ \{\neg R\}$$

Below is one possible resolution sequence.

$$① + ⑤ \overset{R}{\Rightarrow} ⑥ \{\neg P, \neg Q\} \qquad ② + ⑤ \overset{R}{\Rightarrow} ⑦ \{P, Q\}$$
$$⑥ + ④ \overset{P}{\Rightarrow} ⑧ \{\neg Q\} \qquad ⑦ + ③ \overset{P}{\Rightarrow} ⑨ \{Q\} \qquad ⑧ + ⑨ \overset{Q}{\Rightarrow} \square$$

The last few steps were analogous to the second example in Section 6.4 of the notes. Collapsing duplicate clauses like $\{Q, Q\}$ is allowed and often necessary.

e) $(P \to R) \wedge (R \wedge P \to S) \to (P \wedge Q \to R \wedge S)$

Negate and convert to clauses. Only $R \wedge S$ gets negated, the rest of the clauses can be read off from the hypotheses:

- $P \to R$ gives $\{\neg P, R\}$
- $R \wedge P \to S$ gives $\{\neg R, \neg P, S\}$
- $P \wedge Q$ gives $\{P\}, \{Q\}$

With the negated conclusion $\neg R \vee \neg S$, the clause set is:

$$① \{ \neg P, R \} \quad ② \{ \neg P, \neg R, S \} \quad ③ \{ P \} \quad ④ \{ Q \} \quad ⑤ \{ \neg R, \neg S \}$$

Below is a possible linear resolution sequence.

$$① + ② \overset{R}{\Rightarrow} \{ \neg P, S \} + ⑤ \overset{S}{\Rightarrow} \{ \neg P, \neg R \} + ① \overset{R}{\Rightarrow} \{ \neg P \} + ③ \overset{P}{\Rightarrow} \square$$

5. Convert these axioms to clauses, showing all steps. Then prove $\text{Winterstorm} \rightarrow \text{Miserable}$ by resolution.

$$\text{Wet} \wedge \text{Cold} \rightarrow \text{Miserable}$$
$$\text{Winterstorm} \rightarrow \text{Storm} \wedge \text{Cold}$$
$$\text{Storm} \rightarrow \text{Rain} \wedge \text{Windy}$$
$$\text{Rain} \wedge (\text{Windy} \vee \neg\text{Umbrella}) \rightarrow \text{Wet}$$

The propositions listed are our *knowledge base*: the set of axioms that we assume to be true. Our task is to prove $\text{Winterstorm} \rightarrow \text{Miserable}$ using resolution, i.e. that this implication follows from the axioms in our knowledge base. Resolution involves negating the proof goal, since it proceeds by refuting the negation of the proposition; importantly, our initial axioms are *not* negated, as they are assumptions, not goals. Another way to see this is to call the conjunction of the axioms Γ, and the proof goal $\text{Winterstorm} \rightarrow \text{Miserable}$ P. We are required to prove the proposition $\Gamma \rightarrow P$, i.e. P follows from the axioms Γ. To do a resolution proof, we negate this whole implication, and as seen before, this leaves the hypotheses Γ untouched, only negating P: $\neg(\Gamma \rightarrow P) \simeq \Gamma \wedge \neg P$. Hence, the axioms are converted to clauses without negation, then we add the clauses we get from negating $\text{Winterstorm} \rightarrow \text{Miserable}$ and do resolution on the whole clause set.

With some practice, converting formulae into clauses "on the fly" becomes easy; it's important to remember the standard propositional equivalences, and whether the formulae are negated or not! Below are some of the initial steps:

- $\text{Wet} \wedge \text{Cold} \rightarrow \text{Mis} \simeq \neg\text{Wet} \vee \neg\text{Cold} \vee \text{Mis}$
- $\text{Winter} \rightarrow \text{Storm} \wedge \text{Cold} \simeq (\text{Winter} \rightarrow \text{Storm}) \wedge (\text{Winter} \rightarrow \text{Cold})$
- $\text{Storm} \rightarrow \text{Rain} \wedge \text{Wind} \simeq (\text{Storm} \rightarrow \text{Rain}) \wedge (\text{Storm} \rightarrow \text{Wind})$
- $\text{Rain} \wedge (\text{Wind} \vee \neg\text{Umb}) \rightarrow \text{Wet} \simeq (\text{Rain} \wedge \text{Wind}) \vee (\text{Rain} \wedge \neg\text{Umb}) \rightarrow \text{Wet} \simeq (\text{Rain} \wedge \text{Wind} \rightarrow \text{Wet}) \wedge (\text{Rain} \wedge \neg\text{Umb} \rightarrow \text{Wet})$

We also get two clauses from $\neg(\text{Winter} \rightarrow \text{Mis})$, giving us the final clause set of:

$$① \{ \neg\text{Wet}, \neg\text{Cold}, \text{Mis} \} \quad ② \{ \neg\text{Winter}, \text{Storm} \} \quad ③ \{ \neg\text{Winter}, \text{Cold} \}$$
$$④ \{ \neg\text{Storm}, \text{Rain} \} \quad ⑤ \{ \neg\text{Storm}, \text{Wind} \} \quad ⑥ \{ \neg\text{Rain}, \neg\text{Wind}, \text{Wet} \}$$
$$⑦ \{ \neg\text{Rain}, \text{Umb}, \text{Wet} \} \quad ⑧ \{ \text{Winter} \} \quad ⑨ \{ \neg\text{Mis} \}$$

> One sequence of resolution steps is below. Note how we make use of unit clauses as much as possible – resolution with a unit clause is the only way to decrease the size of the clause, as the resolvent of clauses C and D will have size $|C| + |D| - 2$. Unit clauses can also be resolved "in bulk" with another clause containing the negation of the unit literals, as shown in the last step (and the elimination of clause ⑥ beforehand).
>
> $$② + ⑧ \Longrightarrow Ⓐ\ \{\text{Storm}\} \qquad ③ + ⑧ \Longrightarrow Ⓑ\ \{\text{Cold}\} \qquad Ⓐ + ④ \Longrightarrow Ⓒ\ \{\text{Rain}\}$$
> $$Ⓐ + ⑤ \Longrightarrow Ⓓ\ \{\text{Wind}\} \qquad Ⓒ + ⑥ \Longrightarrow Ⓔ\ \{\neg\text{Wind}, \text{Wet}\} \qquad Ⓓ + Ⓔ \Longrightarrow Ⓕ\ \{\text{Wet}\}$$
> $$Ⓕ\ \{\text{Wet}\} + Ⓑ\ \{\text{Cold}\} + ⑨\ \{\neg\text{Mis}\} + ①\ \{\neg\text{Wet}, \neg\text{Cold}, \text{Mis}\} \Longrightarrow \square$$

7. Skolem functions, Herbrand's Theorem and unification

1. a) Explain the process of Skolemisation on a formula of your choice.

> Skolemisation is the transformation of first-order formulae that removes existential quantifiers while maintaining the consistency of the formula. Every existentially quantified variable is replaced by a new function symbol applied to all universally quantified variables bound outside of the scope of the existential. For example, consider the FOL formula:
>
> $$\Big(\forall x.\ \exists y.\ \forall z.\ (P(\underline{x}) \land Q(\underline{z}, \underline{y})) \lor (\exists w.\ R(\underline{w}))\Big) \leftrightarrow \exists y.\ \forall v.\ S(\underline{y}, \underline{v})$$
>
> The variables y, w and y are existentially quantified and will be Skolemised to constant symbols of different arities:
>
> - y variable gets bound in the scope of one universal quantification, so it gets Skolemised to $f(x)$
> - w is also bound in the scope of z, so it gets replaced with $g(x, z)$
> - y is different from the y on the LHS of $\leftrightarrow$ so it gets its own Skolem symbol. In this case, the binding is top-level, so the function has arity zero – that is, it is a constant value c.
>
> Thus, the Skolemised formula is as follows:
>
> $$\Big(\forall x.\ \forall z.\ (P(\underline{x}) \land Q(\underline{z}, f(x))) \lor R(g(x, z))\Big) \leftrightarrow \forall v.\ S(c, \underline{v})$$

b) The notes state that "[Skolemisation] does not preserve the meaning of a formula. However, it does preserve *inconsistency*, which is the critical property". Justify the two claims in this statement, demonstrating them on your example above.

> The justification for Skolemisation can be seen by considering the Tarskian truth definition of a formula $\exists x.\ P(x)$. We say that $\exists x.\ P(x)$ is satisfiable (or consistent) if there exists an interpretation $\mathcal{I} = (I, D)$ such that for all valuations V, we have $\vDash_{\mathcal{I},V} \exists x.\ P(x)$. By the truth definition, this holds if there exists an $m \in D$ such that

$\vDash_{\mathcal{I},V\{m/x\}} P(x)$ holds, which, in turn, is the case if $I[P]((V\{m/x\})(x)) = I[P](m) = 1$.

$$\exists x.\ P(x) \text{ satisfiable } \leftrightarrow\ \exists I, D.\ \exists m \in D.\ I[P](m) = 1$$

The Skolemised version of the formula is simply $P(c)$ for some new constant symbol c, and $P(c)$ is consistent if there exists an interpretation $\mathcal{I}' = (I', D')$ such that for all valuations V' we have $\vDash_{\mathcal{I}',V'} P(c)$. By the truth definition, this holds if $I'[P](I'[c]) = 1$. The crucial point is that the only way $I'[c]$ can be defined is if there is an interpretation of the constant c in the domain D', i.e. an element $m \in D'$ such that $I'[c] = m$. Then:

$$P(c) \text{ satisfiable } \leftrightarrow\ \exists I', D'.\ \exists m \in D'.\ I'[c] = m\ \wedge\ I[P](m) = 1$$

which is nothing more than the satisfiability condition of $\exists x.\ P(x)$. In short, if there is an interpretation of $\exists x.\ P(x)$, there must exist an interpretation of $P(c)$, and, contrapositively, if $P(c)$ is inconsistent then so is $\exists x.\ P(x)$. This reasoning can be extended to Skolem functions incorporating a sequence $\vec{x}$ of quantified variables in $\forall \vec{x}.\ \exists y.\ P(\vec{x}, y)$: if for all sequences of elements $\vec{n} \in D$ corresponding to $\vec{x}$ there must exist an $m \in D$ such that $I[P](\vec{n}, m)$ holds, we can construct a model for $\forall \vec{x}.\ P(f(\vec{x}))$ in which the interpretation of the function symbol f maps $\vec{n}$ to m as prescribed by the model for the original formula. If we can find no model satisfying $\forall \vec{x}.\ P(f(\vec{x}))$, we won't be able to find a model satisfying $\forall \vec{x}.\ \exists y.\ P(\vec{x}, y)$ either.

For a concrete example consider the formula $\forall x.\ \exists y.\ P(x, y)$ (which states that P is a total relation) and take $P(x, y)$ to mean "y is a prime number greater than x", where x and y range over the naturals. By Euclid's proof of the infinitude of primes, we know that this is a satisfying interpretation: for any $x \in \mathbb{N}$, there exists a prime $y \in \mathbb{N}$ such that $x < y$. If there is an assignment of a prime y to every x in the domain $\mathbb{N}$, there must exist (at least one) function $f : \mathbb{N} \to \mathbb{N}$ that assigns x to a larger prime y, for example, the next largest one. This interpretation then satisfies the Skolemised formula $\forall x.\ P(x, f(x))$.

Now, take the formula $\psi = \exists y.\ P(y)\ \wedge\ \neg P(y)$. Skolemising this gives $P(c)\ \wedge\ \neg P(c)$. It's easy to see that no interpretation for P or c exists that would satisfy this formula, which means ψ is unsatisfiable as well – there can't exist a y such that $P(y) \wedge \neg P(y)$.

2. Skolemize the following formulas, dropping all quantifiers.

$$\forall u.\ \exists x, y.\ P(x, y) \qquad \exists x, y.\ \forall z.\ \exists w.\ P(x, y, z, w) \qquad \forall u.\ (\exists x.\ P(x, x))\ \wedge\ \forall v.\ \exists y.\ Q(u, y)$$

Nothing too surprising here.

$$P(f(u), g(u)) \qquad P(a, b, z, f(z)) \qquad P(f(u), f(u))\ \wedge\ Q(u, g(u, v))$$

3. Consider a first-order language A with z and o as constant symbols, with n as a 1-place function symbol and a as a 2-place function symbol, and with C as a 2-place predicate symbol.

   a) Describe the Herbrand universe for this language.

The Herbrand universe of a first-order language is the set of all possible *closed* terms that can be constructed from the constants and function symbols of the language. It is an infinite set constructed recursively: taking any function symbol such as a, the Herbrand universe H must contain the terms $a(t_1, t_2)$ for all terms t_1 and t_2 taken from H itself. Some elements of H are:

$$\{z, o, n(z), n(o), a(z,z), a(z,o), a(o,z), n(n(z)), n(n(o)), n(a(z,z)), a(n(o),z), \dots\}$$

b) This language has an interpretation $\mathcal{I}$ in the domain $D = \mathbb{Z}$ of integers, with z and o interpreted as $0 \in \mathbb{Z}$ and $1 \in \mathbb{Z}$ respectively, n being the negation function $x \mapsto -x$, a being the addition function $x, y \mapsto x + y$, and C being the less-than comparison relation $x, y \mapsto x < y$. What is the Herbrand model of the symbols of the language with respect to this interpretation $\mathcal{I}$?

The Herbrand interpretation of a first-order language L with respect to a given model $\mathcal{I} = (D, I)$ is the "syntactic interpretation" of L which can be directly translated into $\mathcal{I}$. It is an example of a so-called *free construction* in mathematics, because the interpretation can be freely generated from the syntax; any real interpretation would involve some "creativity" (such as interpreting $a(o,z)$ and $a(z,o)$ as the same number 1), while the Herbrand model simply interprets a term as itself, not bothering about actually giving it a meaning. The only place where we need to refer to the real interpretation is giving a meaning to relation symbols, which need to return a truth value even in the free interpretation.

In our example, the constant and function symbols are z, o, n and a; these must be interpreted as constants and functions on the desired domain, which, in this case, is the Herbrand universe H.

- Constant terms are interpreted as themselves: $I_H[z] = z \in H$, $I_H[o] = o \in H$
- $I_H[n] \colon H \to H$ is the function that takes a term $t \in H$, and prepends it with the symbol n (notice how we're not talking about "negating" the element – that happens in the standard interpretation in $\mathbb{Z}$). That is, $I_H[n](t) = n(t)$.
- $I_H[a] \colon H \times H \to H$ is the function that takes terms t_1 and t_2 in H, and combines them with the symbol a. You can think of this as creating a new syntax tree with root a and subtrees t_1 and t_2. That is, $I_H[a](t_1, t_2) = a(t_1, t_2)$.

The relation symbol < cannot be given such a trivial interpretation, which has to be a relation $I_H[<] \colon H \times H \to \mathbb{B}$ – the output must be an actual truth value, not just a syntax tree. Without having a real interpretation in $\mathbb{Z}$ to refer to, we cannot make any obvious choices on how to make sense of a formula like $\varphi = C(a(o,z), n(n(o)))$. Thus, we "ask" what an intended interpretation would say to this, and make that the Herbrand interpretation as well. The choice of interpretation matters: φ would be true if $I_{\mathbb{Z}}[C](x, y) = x \leq y$, but false if $I_{\mathbb{Z}}[C](x, y) = x < y$.

> Formally, the relation $I_H[C]\colon H \times H \to \mathbb{B}$ takes two terms $t_1, t_2 \in H$, interprets them using our base model $I_\mathbb{Z}$, and compares the answers using the base interpretation of C: $I_H[C](t_1, t_2)$ iff $I_\mathbb{Z}(t_1) < I_\mathbb{Z}(t_2)$.
>
> Herbrand interpretations are useful because every real interpretation of a set of first-order clauses *factorises* through a Herbrand interpretation in a unique way. That is, given an interpretation $\mathcal{I} = (D, I)$ of a set of clauses S (or a quantifier-free formula that may contain variables), there exists a unique function $\hat{I}\colon H \to D$ such that the interpretation $I[t]$ of a term coincides with $\hat{I}(I_H[t])$ for all terms t. Instead of interpreting the set of clauses directly, we can take a "detour" through the systematically generated Herbrand interpretation in a consistency-preserving way.
>
> For example, take the clauses
>
> $$S = \{\, C(x, a(x, o)) \,\} \quad \{\, C(x, x), C(z, o) \,\}$$
>
> These stand for the FOL formula $(\forall x.\ C(x, a(x, o))) \wedge (\forall y.\ C(y, y) \vee C(z, o))$. We have the above interpretation $(\mathbb{Z}, I_\mathbb{Z})$ which satisfies S: it is indeed the case that $x < x + 1$ for all $x \in \mathbb{Z}$, and that either $y < y$ or $0 < 1$ holds for all y. The claim of Lemma 12 in the notes is that there must exist a Herbrand interpretation satisfying S. The universe H is as described above, consisting of elements such as $a(n(o), z)$ and $n(n(n(z)))$. Similarly, interpretations of the constant and functions symbols are the identity: $I_H[a(n(o), z)] = a(n(o), z)$. The interpretation of C is where we need to make use of $I_\mathbb{Z}$, as explained above. To show that I_H satisfies the clauses, we need to prove that $I_H[C](t, a(t, o))$ holds for all terms $t \in H$ (and similarly for the second clause). How do we know that this is true? Well, $I_H[C](t, a(t, o)) \longleftrightarrow I_\mathbb{Z}[t] < I_\mathbb{Z}[a(t, o)] \longleftrightarrow I_\mathbb{Z}[t] < I_\mathbb{Z}[t] + 1$ for all $t \in H$, but this certainly holds, since we know that $x < x + 1$ for all $x \in \mathbb{Z}$ – in particular, $I_\mathbb{Z}[t]$. The real interpretation $I_\mathbb{Z}$ can be split into the "freely generated" Herbrand interpretation I_H, followed by a unique function $\hat{I}_\mathbb{Z}\colon H \to \mathbb{Z}$, mapping the ground terms of the Herbrand universe to integers.

4. For at least three of the following pairs of terms, give a most general unifier or explain why none exists. Do not rename variables prior to performing the unification.

$f(g(x), z)$	$f(y, h(y))$
$j(x, y, z)$	$j(f(y, y), f(z, z), f(a, a))$
$j(x, z, x)$	$j(y, f(y), z)$
$j(f(x), y, a)$	$j(y, z, z)$
$j(g(x), a, y)$	$j(z, x, f(z, z))$

> Things to pay attention to: occurs-check violations (e.g. trying to unify $f(x)$ with x), and not unifying symbols with different symbols. Substitutions also need to be accumulated when unifying arguments of an n-ary function symbol in sequence, and the final substitution is

the composition of the component substitutions (see Section 7.6 of the notes).

- The MGU of $f(g(x), z)$ and $f(y, h(y))$ is $[g(x)/y, h(g(x))/z]$, and the common instance is $f(g(x), h(g(x)))$. As noted above, substitutions accumulate: in the second step we are unifying $z[g(x)/y] = z$ and $h(y)[g(x)/y] = h(g(x))$, not just z and $h(y)$. Without accumulation, the substitution would be $\sigma = [g(x)/y, h(y)/z]$ – but this is not even a unifier, because $f(g(x), z)[\sigma] = f(g(x), h(y))$ but $f(y, h(y))[\sigma] = f(g(x), h(g(x))$.
- The MGU of $j(x, y, z)$ and $j(f(y, y), f(z, z), f(a, a))$ is the composition of $[f(y, y)/x]$, $[f(z, z)/y]$ and $[f(a, a)/z]$, namely:

$$[f(f(f(a,a),f(a,a)),f(f(a,a),f(a,a)))/x,\\ f(f(a,a),f(a,a))/y,\\ f(a,a)/z]$$

  This is a slightly contrived example to demonstrate how the naive unification algorithm can take exponential time.
- The terms $j(x, z, x)$ and $j(y, f(y), z)$ are not unifiable. A unifier must identify the variables x, y and z, and thus also unify y with $f(y)$, which violates the occurs check.
- The terms $j(f(x), y, a)$ and $j(y, z, z)$ are also not unifiable. We have to unify y both with a and $f(x)$, but a and $f(x)$ are not unifiable as they are different function symbols.
- The MGU of $j(g(x), a, y)$ and $j(z, x, f(z, z))$ is the composition of $[g(x)/z]$, $[a/x]$ and $[f(z, z)/y]$, namely $[a/x, f(g(a), g(a))/y, g(z)/z]$. The common instance is $j(g(a), a, f(g(a), g(a)))$.

5. Which of the following substitutions are most general unifiers for the terms $f(x, y, z)$ and $f(w, w, v)$?

$$[x/y, x/w, v/z] \qquad [y/x, y/w, v/z] \qquad [y/x, v/z]$$
$$[x/y, x/z, x/w, x/v] \qquad [u/x, u/y, u/w, y/z, y/v]$$

- $[x/y, x/w, v/z]$: this is a unifier with the common instance $f(x, x, v)$.
- $[y/x, y/w, v/z]$: this is a unifier with the common instance $f(y, y, v)$.
- $[y/x, v/z]$: this is not a unifier, as the first term becomes $f(y, y, v)$, while the other is $f(w, w, v)$. We're missing $[y/w]$, which the previous substitution had.
- $[x/y, x/z, x/w, x/v]$: this is a unifier with the common instance $f(x, x, x)$. But it unifies "more things" than required, that is, it's not a most general unifier: we can get to the same term by applying the first substitution $[x/y, x/w, v/z]$ above to get $f(x, x, v)$, then also substituting x for v. Thus, $[x/y, x/z, x/w, x/v] = [x/y, x/w, v/z] \circ [x/v]$, so this is not an MGU.
- $[u/x, u/y, u/w, y/z, y/v]$: this is a unifier with the common instance $f(u, u, y)$. Again, it performs more substitutions than required and can be decomposed (for example)

> as $[u/x, u/y, u/w, y/z, y/v] = [x/y, x/w, v/z] \circ [u/x, y/v]$.

8. First-order resolution

1. What techniques allow us to convert first-order formulas into "propositional" clauses, and prove them using resolution? How are quantifiers and variables handled?

   > See the *First-order resolution* supplement.

2. Is the clause $\{ P(x, b), P(a, y) \}$ logically equivalent to the unit clause $\{ P(a, b) \}$? Is the clause $\{ P(y, y), P(y, a) \}$ logically equivalent to $\{ P(y, a) \}$? Explain both answers.

   > Logical equivalence would imply that $(\forall xy.\ P(x, b) \lor P(a, y)) \leftrightarrow P(a, b)$ is valid. Of course, this is not the case: $P(a, b) \to (\forall xy.\ P(x, b) \lor P(a, y))$ is clearly wrong. We can find a falsifying model over the domain $\{ 0, 1 \}$: $0 < 1 \not\to (1 < 1 \lor 0 < 0)$.
   >
   > The second pair of clauses is also not equivalent: $\forall y.\ P(y, y) \lor P(y, a)$ does not imply $\forall x.\ P(x, a)$ because $\forall y.\ P(y, y) \to P(y, a)$ is not valid. Again, a falsifying model over $\{ 0, 1 \}$ could be: $0 = 0 \not\to 0 = 1$.
   >
   > This shows that factoring usually results in logically weaker clauses so it's worth retaining the original clause if we want our proof procedure to be complete.

3. Show that every set S of definite clauses is consistent. *Hint*: first consider propositional logic, then extend your argument to first order logic.

   > Definite clauses contain exactly one positive literal: they are of the form $\{ \neg A_1, \ldots, \neg A_n, B \}$ and can be interpreted as implications $A_1 \land \cdots \land A_n \to B$. To satisfy a set of these implications, it is sufficient to ignore the hypotheses and satisfy the consequent (since $A \to \top$ is a tautology). Thus, all we need to do is set the positive literals occurring in each clause to true, and by the definite clause guarantee every clause will be satisfied. The same idea works for first-order clauses, except the positive literal may contain variables: these will need to be universally quantified. For instance, $A_1(x) \land \cdots \land A_n(x) \to B(x)$ is satisfied by the interpretation $B(x) = \top$ for all $x \in D$. Note that we can't use the Herbrand logic here: satisfying a single ground instance of a clause does not satisfy the full clause.

4. Convert the following formulas into clauses, showing each step: negating the formula, eliminating $\to$ and $\leftrightarrow$, pushing in negations, Skolemising, dropping the universal quantifiers, and converting the resulting formula into CNF. Apply resolution (and possibly factoring) to prove or disprove the formulas in each case.

$$(\exists x.\ \forall y.\ R(x, y)) \to (\forall y.\ \exists x.\ R(x, y))$$
$$(\forall y.\ \exists x.\ R(x, y)) \to (\exists x.\ \forall y.\ R(x, y))$$
$$\exists x.\ \forall y, z.\ (P(y) \to Q(z)) \to (P(x) \to Q(x))$$
$$\neg(\exists y.\ \forall x.\ R(x, y) \leftrightarrow \neg(\exists z.\ R(x, z) \land R(z, x)))$$

a) Negate and convert to clauses

$$\begin{aligned}
&\neg((\exists x.\ \forall y.\ R(x,y)) \to (\forall y.\ \exists x.\ R(x,y))) \\
&\simeq (\exists x.\ \forall y.\ R(x,y)) \land (\exists y.\ \forall x.\ \neg R(x,y)) && \text{(negate)} \\
&\Longrightarrow (\forall y.\ R(a,y)) \land (\forall x.\ \neg R(x,b)) && \text{(Skolemise)} \\
&\Longrightarrow R(a,y) \land \neg R(x,b) && \text{(drop } \forall\text{s)} \\
&\Longrightarrow \{R(a,y)\} \quad \{\neg R(x,b)\}
\end{aligned}$$

The two clauses can be unified with $[a/x, b/y]$, and resolved to the empty clause, proving the original formula.

b) Negate and convert to clauses

$$\begin{aligned}
&\neg((\forall y.\ \exists x.\ R(x,y)) \to (\exists x.\ \forall y.\ R(x,y))) \\
&\simeq (\forall y.\ \exists x.\ R(x,y)) \land (\forall x.\ \exists y.\ \neg R(x,y)) && \text{(negate)} \\
&\Longrightarrow (\forall y.\ R(f(y),y)) \land (\forall x.\ \neg R(x,g(x))) && \text{(Skolemise)} \\
&\Longrightarrow R(f(y),y) \land \neg R(x,g(x)) && \text{(drop } \forall\text{s)} \\
&\Longrightarrow \{R(f(y),y)\} \quad \{\neg R(x,g(x))\} && \text{(convert to clauses)}
\end{aligned}$$

Unifying the first argument gives $[f(y)/x]$, but attempting to unify $g(x)[f(y)/x] = g(f(y))$ and y is not possible due to the occurs check. Since no resolution steps are possible, the original formula must be invalid.

c) Negate and convert to clauses

$$\begin{aligned}
&\neg(\exists x.\ \forall y,z.\ (P(y) \to Q(z)) \to (P(x) \to Q(x))) \\
&\simeq \forall x.\ \exists y,z.\ (P(y) \to Q(z)) \land P(x) \land \neg Q(x) && \text{(negate)} \\
&\simeq (\exists y,z.\ \neg P(y) \lor Q(z)) \land (\forall x.\ P(x) \land \neg Q(x)) && \text{(convert to miniscope)} \\
&\Longrightarrow (\neg P(a) \lor Q(b)) \land (\forall x.\ P(x) \land \neg Q(x)) && \text{(Skolemise)} \\
&\Longrightarrow (\neg P(a) \lor Q(b)) \land P(x) \land \neg Q(x) && \text{(drop } \forall\text{s)} \\
&\Longrightarrow ① \{\neg P(a), Q(b)\} \quad ② \{P(x)\} \quad \neg ③ \{Q(x)\} && \text{(convert to clauses)}
\end{aligned}$$

Resolve ① and ② on $P(a)$ with the unifier $[a/x]$ to get ④ $\{Q(b)\}$. Resolve ④ and ③ on $Q(b)$ with the unifier $[b/x]$ to get $\square$.

d) Negate and convert to clauses

$$\begin{aligned}
&\neg\neg(\exists y.\ \forall x.\ R(x,y) \leftrightarrow \neg(\exists z.\ R(x,z) \land R(z,x))) \\
&\simeq \exists y.\ \forall x.\ R(x,y) \leftrightarrow \neg(\exists z.\ R(x,z) \land R(z,x)) && \text{(negate)} \\
&\simeq \exists y.\ \forall x.\ (R(x,y) \to \neg(\exists z.\ R(x,z) \land R(z,x))) \land ((\exists z.\ R(x,z) \land R(z,x)) \lor R(x,y)) \\
& && \text{(expand } \leftrightarrow\text{)}
\end{aligned}$$

$$\simeq \exists y.\, \forall x.\, (\neg R(x,y) \vee (\forall z.\, \neg R(x,z) \vee \neg R(z,x))) \wedge ((\exists z.\, R(x,z) \wedge R(z,x)) \vee R(x,y))$$
(de Morgan)

$$\Longrightarrow \forall x.\, (\neg R(x,a) \vee (\forall z.\, \neg R(x,z) \vee \neg R(z,x))) \wedge ((R(x,f(x)) \wedge R(f(x),x)) \vee R(x,a))$$
(Skolemise)

$$\Longrightarrow (\neg R(x,a) \vee \neg R(x,z) \vee \neg R(z,x)) \wedge ((R(x,f(x)) \wedge R(f(x),x)) \vee R(x,a))$$
(drop $\forall$s)

$$\simeq (\neg R(x,a) \vee \neg R(x,z) \vee \neg R(z,x)) \wedge (R(x,f(x)) \vee R(x,a)) \wedge (R(f(x),x) \vee R(x,a))$$
(convert to CNF)

$$\Longrightarrow ① \{\neg R(x,a), \neg R(x,z), \neg R(z,x)\}\ ② \{R(x,f(x)), R(x,a)\}\ ③ \{R(f(x),x), R(x,a)\}$$
(convert to clauses)

- There are no unit clauses, so we should see if factoring is possible. Indeed it is: the first clause has ④ $\{\neg R(a,a)\}$ as a factored instance.
- Resolve ④ and ② on $R(a,a)$ with the unifier $[a/x]$ to get ⑤ $\{R(a,f(a))\}$
- Resolve ④ and ③ on $R(a,a)$ with the unifier $[a/x]$ to get ⑥ $\{R(f(a),a)\}$
- Resolve ⑥ and ① on $R(f(a),a)$ and $R(x,a)$ with the unifier $[f(a)/x]$ to get ⑦ $\{\neg R(f(a),y), \neg R(y,f(a))\}$
- Resolve ⑥ and ⑦ on $R(f(a),a)$ and $R(f(a),y)$ with the unifier $[a/y]$ to get ⑧ $\{\neg R(a,f(a))\}$
- Resolve ⑧ and ⑤ on $R(a,f(a))$ to get $\square$.

5. Refute the following set of clauses using resolution and factoring.

$$① \{P(x,b), P(a,y)\} \qquad ② \{\neg P(x,b), \neg P(c,y)\} \qquad ③ \{\neg P(x,d), \neg P(a,y)\}$$

This is an example of when "greedy" factoring is problematic. We can factor all three clauses to get $\{P(a,b)\}$, $\{\neg P(c,b)\}$ and $\{\neg P(a,d)\}$, but there is no way to move forward with only these terms, since they have no variables left to unify. We need to be more strategic and only factor when we need to. Factoring ② yields $\{\neg P(c,b)\}$ which can be resolved with ① to get ④ $\{P(a,y)\}$. This, together with the factored clause ③, $\{\neg P(a,d)\}$, yields the empty clause, as required.

6. Prove the following formulas by resolution, showing all steps of the conversion into clauses. Note that P is just a predicate symbol, so in particular, x is not free in P.

$$(\forall x.\, P \vee Q(x)) \to (P \vee \forall x.\, Q(x)) \qquad \exists x, y.\, (R(x,y) \to \forall z, w.\, R(z,w))$$

a) Negating $(\forall x.\, P \vee Q(x)) \to (P \vee \forall x.\, Q(x))$ leaves the hypothesis untouched, so we can immediately read off the clause $\{P, Q(x)\}$. Negating the conclusion gives $\neg P \wedge \exists x.\, \neg Q(x)$, and Skolemising results in $\neg P \wedge \neg Q(a)$. The final clause set is:

$$① \{P, Q(x)\} \quad ② \{\neg P\} \quad ③ \{\neg Q(a)\}$$

Resolve ① and ③ on $Q(a)$ with the unifier $[a/x]$ to get ④ $\{P\}$, which, with ②, gives

the empty clause.

b) Negating $\exists x, y.\ (R(x,y) \to \forall z, w.\ R(z,w))$ gives $\forall x, y.\ R(x,y) \wedge \exists z, w.\ \neg R(z,w)$. Skolemisation introduces two 2-place Skolem functions: $\forall x, y.\ R(x,y) \wedge \neg R(f(x,y), g(x,y))$. The two clauses are:

$$① \{R(x,y)\} \qquad ② \{\neg R(f(x,y), g(x,y))\}$$

While it seems like we cannot unify x with $f(x,y)$ due to the occurs check, we must always remember that variables in a clause can be renamed arbitrarily:

$$① \{R(u,v)\} \qquad ② \{\neg R(f(x,y), g(x,y))\}$$

Now, the two clauses resolve with the unifier $[f(x,y)/u, g(x,y)/v]$ and yield $\square$.

9. Optional exercises

1. In your own words, explain the motivation behind Herbrand interpretations.

   - How is a Herbrand interpretation constructed from a set of clauses S?
   - Why do we need Herbrand interpretations?
   - What is the significance of the Skolem–Gödel–Herbrand Theorem?

   If you wish, consult a pre-2013 version of the course lecture notes, which discuss Herbrand models in more detail.

   > This was explained in detail in Ex. 7.3 and the supplementary document.

2. Consider the Prolog program consisting of the definite clauses

$$P(f(x,y)) \leftarrow Q(x), R(y)$$
$$Q(g(z)) \leftarrow R(z)$$
$$R(a) \leftarrow$$

   Describe the Prolog computation starting from the goal clause $\leftarrow P(v)$. Keep track of the substitutions affecting v to determine what answer the Prolog system would return.

   > A Prolog program consists of a database of definite clauses (containing exactly one positive literal, representing the conclusion of an implication) and a goal clause (containing only negative literals, representing the set of unsolved goals). Computation happens by linear resolution: the goal clause is repeatedly resolved with one of the definite clauses until all goals are discharged.
   >
   > In this example, the Prolog program represents the following definite clauses:
   >
   > $$① \{\neg Q(x), \neg R(y), P(f(x,y))\} \quad ② \{\neg R(z), Q(g(z))\} \quad ③ \{R(a)\}$$
   >
   > The goal clause is Ⓖ $\{\neg P(v)\}$; the aim of the program is to find out what value for the variable v would give a contradiction. The advantages of the Prolog constraints are that one of the resolvents is always the result of the previous resolution (or the goal clause at

the start), and the unifiers compose after each resolution step.

- Resolve ① $\{\neg Q(x), \neg R(y), P(f(x,y))\}$ and Ⓖ with the unifier $[f(x,y)/v]$ to get Ⓖ $\{\neg Q(x), \neg R(y)\}$
- Resolve ② $\{\neg R(z), Q(g(z))\}$ and Ⓖ with the unifier $[g(x)/x]$ to get Ⓖ $\{\neg R(z), \neg R(y)\}$
- Resolve ③ $\{R(a)\}$ and Ⓖ with the unifier $[a/z]$ to get Ⓖ $\{\neg R(y)\}$
- Resolve ③ $\{R(a)\}$ and Ⓖ with the unifier $[a/y]$ to get $\square$.

The substitutions are:

$$[f(x,y)/v] \circ [g(z)/x] \circ [a/z] \circ [a/y] = [f(g(a),a)/v, g(a)/x, a/y, a/z]$$

That is, the final answer is $v = f(g(a),a)$.

10. Decision procedures and SMT solvers

1. In Fourier–Motzkin variable elimination, any variable not bounded both above and below is deleted from the problem. For example, given the set of constraints

$$3x \geq y \qquad x \geq 0 \qquad y \geq z \qquad z \leq 1 \qquad z \geq 0$$

the variables x and then y can be removed (with their constraints), reducing the problem to $z \leq 1 \wedge z \geq 0$. Explain how this happens and why it is correct.

> If a variable w is constrained in only one direction, then a suitable value for it can be calculated after the constraints on the other variables have been solved. If the set of constraints is ultimately unsatisfiable, this will never be due to the variable not constrained on both sides since unsatisfiable constraints must be reducible to $w \leq l$ and $u \leq w$ for some $l < u$. In the example above, x starts with having no upper bounds, so its value can be selected to be an arbitrarily large value once the constraints for other variables have been satisfied. Removing the first two terms also leaves y without any upper bounds, so it too can be eliminated. The remaining constraints involving z are easily satisfied as $z = 0$ or $z = 1$, and values for y, then x can be chosen accordingly afterwards; for instance, $y = 1$ and $x = 3$.

2. Apply Fourier–Motzkin variable elimination to the following sets of constraints.

   a) ① $x \geq z$ ② $y \geq 2z$ ③ $z \geq 0$ ④ $x + y \leq z$

   > Eliminate x by combining ① $z \leq x$ and ④ $x + y \leq z \leftrightarrow x \leq z - y$ to get $z \leq z - y$, which is equivalent to ⑤ $y \leq 0$.
   >
   > ② $y \geq 2z$ ③ $z \geq 0$ ⑤ $y \leq 0$
   >
   > Eliminate y by combining ② $2z \leq y$ and ⑤ $y \leq 0$ to get $2z \leq 0$, i.e. ⑥ $z \leq 0$.
   >
   > ③ $z \geq 0$ ⑥ $z \leq 0$
   >
   > This is satisfiable with $z = 0$; the previous constraint $2z \leq y \leq 0$ implies $y = 0$ and

> $z \leq x \leq z - y$ implies $x = 0$.

b) ① $x \leq 2y$ ② $x \leq y + 3$ ③ $z \leq x$ ④ $0 \leq z$ ⑤ $y \leq 4x$

> Eliminate z by combining ④ $0 \leq z$ and ③ $z \leq x$ to get ⑥ $0 \leq x$.
>
> $$\text{① } x \leq 2y \qquad \text{② } x \leq y + 3 \qquad \text{⑤ } \frac{y}{4} \leq x \qquad \text{⑥ } 0 \leq x$$
>
> Eliminate x by combining: ⑥ and ① to get $0 \leq 2y \leftrightarrow 0 \leq y$; ⑥ and ② to get $0 \leq y + 3 \leftrightarrow -3 \leq y$; ⑤ and ① to get $\frac{y}{4} \leq 2y \leftrightarrow 0 \leq y$ and ⑤ and ② to get $\frac{y}{4} \leq y + 3 \leftrightarrow -4 \leq y$. In the resulting constraints y is only bounded from below so they are satisfiable; the most limiting constraint is $0 \leq y$. Again, a possible model is $x = y = z = 0$.

3. Summarise the main ideas behind SMT solvers: how do they combine decision procedures with clause-based methods and what kinds of problems do they allow us to solve?

> SMT solvers allow us to solve complex decision problems by separating purely logical reasoning from theory-specific constraints. An SMT instance is a FOL formula where the constant, function, and relation symbols have extrinsic interpretations, e.g. in the theory of real numbers and inequalities, lists or arrays, bit vectors, etc. While it may be possible to solve formulae purely symbolically (e.g. using first-order resolution), particular theories often have specialised decision procedures that are more efficient. If a formula is unsatisfiable, it may be due to some theory-specific conflict like $x \leq 0 \wedge 3 \leq x$, but it could be a purely logical contradiction like $x = 1 \wedge \neg(x = 1)$. SMT solvers turn the formula into clausal form, treating atomic formulae like $\boxed{x = 0}$ and $\boxed{y \leq 5}$ as opaque propositional letters, with the exception of some negated or dual relationships, such as interpreting $y \neq 1$ as $\neg \boxed{y = 1}$. In the first pass, a propositional model-finding algorithm like DPLL is used to "estimate" a model and weed out formulae which are logically inconsistent. The estimate – which may be vastly simpler than the original formula – is then passed on to the theory-specific decision procedure to confirm if it is indeed a model. If it is not, the decision procedure returns a refutation or counterexample to DPLL which can now refine its guess or conclude that no models it can propose are actually valid. This separation of responsibilities makes SMT solvers well suited for even very large formulas that occur in practical settings like system verification and automated theorem proving.

4. Apply the SMT algorithm sketched in the notes to the following set of clauses. Recall that the constraints $\boxed{c > 0}$ and $\boxed{c < 0}$ are unrelated.

$$\{c = 0, c > 0\} \qquad \{a \neq b\} \qquad \{c < 0, a = b\}$$

> The DPLL algorithm receives the following set of clauses of opaque literals:
>
> $$\text{① } \left\{\boxed{c = 0}, \boxed{c > 0}\right\} \qquad \text{② } \left\{\neg \boxed{a = b}\right\} \qquad \text{③ } \left\{\boxed{c < 0}, \boxed{a = b}\right\}$$

Unit propagation removes clause ② and $\boxed{a = b}$ from clause ③.

$$① \left\{ \boxed{c = 0}, \boxed{c > 0} \right\} \qquad ③ \left\{ \boxed{c < 0} \right\}$$

Again, we unit propagate ③ which doesn't affect ①.

$$① \left\{ \boxed{c = 0}, \boxed{c > 0} \right\}$$

Now, we choose a literal to do a case split on; if $\boxed{c = 0}$ is true, ① is satisfied and we have the proposed DPLL model

$$\boxed{a = b} \qquad \boxed{c < 0} \qquad \boxed{c = 0}$$

which gets passed on to a decision procedure for inequalities. Analysing the contents of the literals, it is easy to see that the constraints are unsatisfiable, so the decision procedure returns the negation of the model which becomes an additional clause:

$$④ \left\{ \neg \boxed{a = b}, \neg \boxed{c < 0}, \neg \boxed{c = 0} \right\}$$

With the first case rejected, DPLL analyses the case $\neg \boxed{c = 0}$. Clause ④ gets deleted, and clause ① becomes $① \left\{ \boxed{c > 0} \right\}$, which, upon unit propagation, gives another model

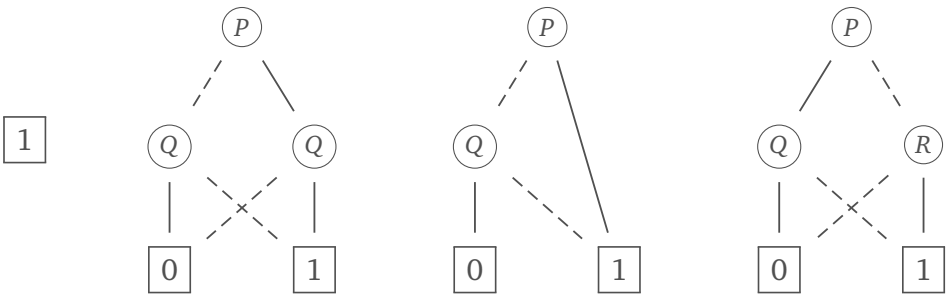
$$\boxed{a = b} \qquad \boxed{c < 0} \qquad \neg \boxed{c = 0} \qquad \boxed{c > 0}$$

However, this too gets rejected by the decision procedure. Since no more backtracking is possible, the SMT solver concludes that the set of clauses is unsatisfiable.

11. Binary decision diagrams

1. Compute the BDD for each of the following formulas, taking the variables as alphabetically ordered.

$$P \wedge Q \to Q \wedge P \qquad P \vee Q \to P \wedge Q \qquad \neg(P \vee Q) \vee P \qquad \neg(P \wedge Q) \leftrightarrow (P \vee R)$$

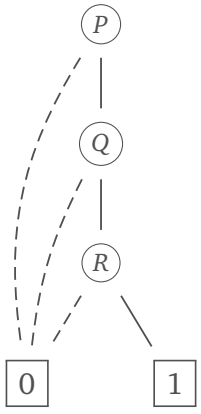


The first formula is a tautology, so its canonical BDD is simply $\boxed{1}$ – upon construction we see that the case analyses on the individual propositions do not actually branch into distinct subdiagrams and can therefore be collapsed.

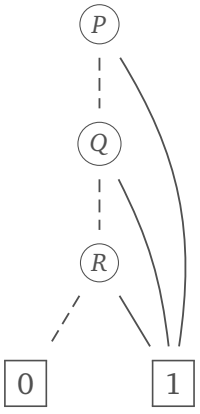
2. Verify these equivalences using BDDs.

$$(P \wedge Q) \wedge R \simeq P \wedge (Q \wedge R) \qquad (P \vee Q) \vee R \simeq P \vee (Q \vee R)$$
$$P \vee (Q \wedge R) \simeq (P \vee Q) \wedge (P \vee R) \qquad P \wedge (Q \vee R) \simeq (P \wedge Q) \vee (P \wedge R)$$
$$\neg(P \wedge Q) \simeq \neg P \vee \neg Q \qquad (P \leftrightarrow Q) \leftrightarrow R \simeq P \leftrightarrow (Q \leftrightarrow R)$$
$$(P \vee Q) \rightarrow R \simeq (P \rightarrow R) \wedge (Q \rightarrow R) \qquad (P \wedge Q) \rightarrow R \simeq P \rightarrow (Q \rightarrow R)$$

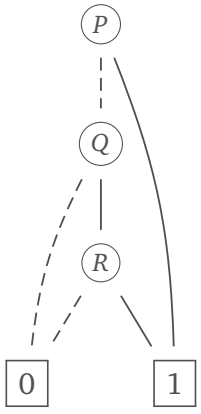
> Since the purpose of BDDs is that they are canonical representations of propositional formulae, the BDDs generated from both sides of the equivalences will be identical (and will only be shown once below). The method of constructing the BDDs will usually differ, however, so it is a good practice opportunity to go through the process of building both diagrams independently.



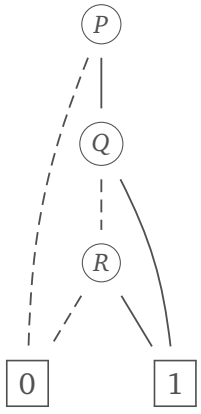
$(P \wedge Q) \wedge R$



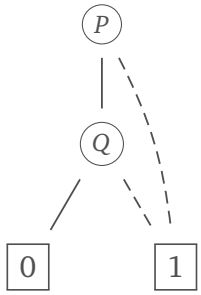
$(P \vee Q) \vee R$



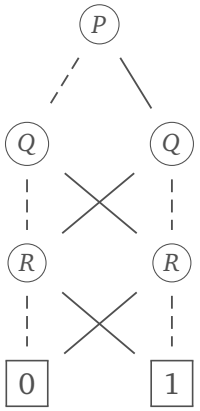
$P \vee (Q \wedge R)$



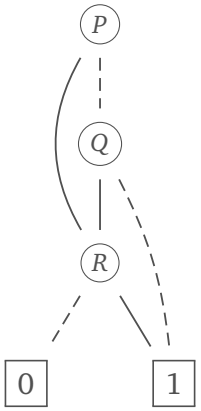
$P \wedge (Q \vee R)$



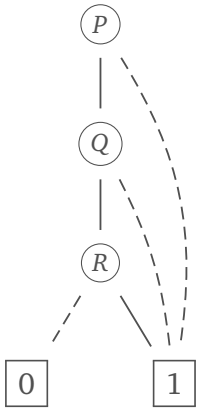
$\neg(P \wedge Q)$



$(P \leftrightarrow Q) \leftrightarrow R$



$(P \vee Q) \rightarrow R$



$(P \wedge Q) \rightarrow R$

12. Modal logics

1. Explain why adding the T, 4 and B axioms make the transition relation reflexive, transitive and symmetric, respectively? Consider both the informal meaning and the formal semantics.

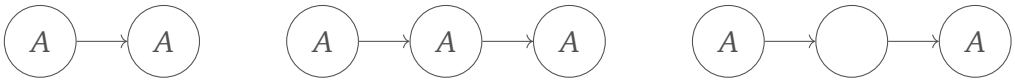
We first consider a reflexive/transitive/symmetric frame and show that it must satisfy the respective axioms.

Reflexivity $\to T$. Let (W, R) be a reflexive modal frame: for all $w \in W$, $(w, w) \in R$. We show that $\vDash_{W,R} \Box A \to A$. Take a world $w \in W$ and assume $w \Vdash \Box A$; that is, $v \Vdash A$ for all $w \in W$ such that $R(w, v)$. Since R is reflexive, one of the successor worlds must be w itself, so we also have $w \Vdash A$. Put together, this implies $w \Vdash \Box A \to A$ for an arbitrary w, so in fact $\vDash_{W,R} \Box A \to A$ as required.

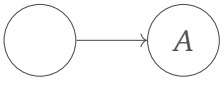
Transitivity $\to 4$. Let (W, R) be a transitive modal frame and assume $w \Vdash \Box A$ for a world $w \in W$. We need to show that $w \Vdash \Box\Box A$; that is, for all $v \in W$ for which $R(w, v)$, and for all $u \in W$ for which $R(v, u)$, we have $u \Vdash A$. Take such $v, u \in W$ satisfying $R(w, v)$ and $R(v, u)$. Since R is transitive, we also have $R(w, u)$, and combining this with the assumption $w \Vdash \Box A$, we have that $u \Vdash A$, as required.

Symmetry $\to B$. Let (W, R) be a symmetric modal frame and assume $w \Vdash A$ for a world $w \in W$. We need to show that $w \Vdash \Box\Diamond A$; that is, for every $u \in W$ with $R(w, u)$ there exists a $v \in W$ with $R(u, v)$ satisfying $w \Vdash A$. Take such an arbitrary $u \in W$ satisfying $R(w, u)$. Since R is symmetric, we also have $R(u, w)$, and by assumption $w \Vdash A$, so it serves as the required witness of existence.

The converse direction is a bit more subtle, especially since it is quite easy to come up with examples that seemingly violate the statements. For example, all states in the following three frames satisfy T, 4 and B respectively, but the corresponding frames are not reflexive, transitive, or symmetric:



The key point to recognise, however, is the following: when we say that T, 4 or B are *axioms* in a modal frame, we mean that they are satisfied in any world and *under any interpretation*. That is, given a frame (W, R), it satisfies axiom T if $\vDash_{W,R} \Box A \to A$, which, by definition, means that $\vDash_{W,R,I} \Box A \to A$ under any interpretation I. The above "counterexample" only satisfies T under a particular assignment, but one can find an assignment where T does not hold in every world:



Consequently, we cannot say that our frame (W, R) satisfies T so whether R is reflexive or not is irrelevant. The only way to ensure that a frame satisfies T no matter what assignment we choose is to make it reflexive. This flexibility over the interpretation is a crucial requirement for the converse proofs.

$T \to$ **Reflexivity**. Let (W, R) be a frame and assume $\vDash_{W,R} \Box A \to A$; that is, under all interpretations I and in all worlds $w \in W$, $w \Vdash \Box A \to A$. We need to show that R is

reflexive, that is, for all $w \in W$, $R(w,w)$ holds. Take an arbitrary world $w \in W$, and for contradiction, assume that there is no transition from w to itself. Consider the interpretation $I_R(A) = W \setminus \{w\}$: A holds in all worlds except w. By the initial assumption, we have that if $w \Vdash \Box A$ then $w \Vdash A$ under the interpretation I_R, and we indeed have $w \Vdash \Box A$ since every world that w can transition to (which can only be states other than w since $\neg R(w,w)$) satisfies A. But then $w \Vdash A$ due to the axiom T, which is a contradiction since our assignment of A specifically excluded w. Thus, there must be a loop $R(w,w)$ for any w, proving that the frame is reflexive.

4 $\rightarrow$ **Transitivity**. Let (W,R) be a frame and assume $\vDash_{W,R} \Box A \rightarrow \Box\Box A$. We need to prove that for all $w,v,u \in W$, if $R(w,v)$ and $R(v,u)$ then $R(w,u)$. Take such worlds $w,v,u \in W$ with $R(w,v)$ and $R(v,u)$ and for contradiction assume that $\neg R(w,u)$. Consider the interpretation $I_T(A) = W \setminus \{u\}$. By the initial assumption, we have that if $w \Vdash \Box A$ then $w \Vdash \Box\Box A$ under I_T, and we indeed have $w \Vdash \Box A$ since all the worlds that w can transition into (which does not include u by the assumption $\neg R(w,u)$) satisfy A. Then, by 4, we have $w \Vdash \Box\Box A$, which implies that u must satisfy A, contradicting our initial assumption. Since we reach a contradiction from $\neg R(w,u)$, we can conclude that R must be transitive.

$B \rightarrow$ **Symmetry**. Let (W,R) be a frame and assume $\vDash_{W,R} A \rightarrow \Box\Diamond A$. We need to prove that for all $w,v \in W$, if $R(w,v)$ then $R(v,w)$. Take such worlds $w,v \in W$ with $R(w,v)$ and for contradiction assume that $\neg R(v,w)$. Consider the interpretation $I_S(A) = W \setminus \{u \in W \mid R(v,u)\}$: A holds in every state other than the ones v can transition to. Since by assumption $\neg R(v,w)$, we have that $w \Vdash A$, and by the axiom B, this implies $w \Vdash \Box\Diamond A$. However, this in particular means that there must be a state that u transitions to satisfying A, which contradicts our original assumption on I_S; thus, it can't be the case that $\neg R(v,w)$ so R must be symmetric.

2. Why does the dual of an operator string equivalence also hold? For example, how can we deduce $\Diamond\Diamond A \simeq \Diamond A$ from $\Box\Box A \simeq \Box A$?

We have the de Morgan duality $\neg\Box A \simeq \Diamond\neg A$ for modalities, which means every $\Box$ can be expressed as $\neg\Diamond\neg$ and vice versa. Rewriting a string of modalities in such a way results in a dual string of modalities separated by pairs of negations and "bookended" by a single negation on each side. The double negations cancel out, and so do the negations on the ends. For example, the equivalence $\Box\Box A \simeq \Box A$ translates to $\neg\Diamond\neg\neg\Diamond\neg A \simeq \neg\Diamond\neg A$ for all A, which simplifies to $\neg\Diamond\Diamond\neg A \simeq \neg\Diamond\neg A$. Negation preserves equivalences, so we have $\Diamond\Diamond\neg A \simeq \Diamond\neg A$, and this is merely the equivalence $\Diamond\Diamond B \simeq \Diamond B$ for $B = \neg A$.

3. a) Prove the sequents $\Diamond(A \vee B) \Rightarrow \Diamond A, \Diamond B$ and $\Diamond A \vee \Diamond B \Rightarrow \Diamond(A \vee B)$, thus proving the equivalence $\Diamond(A \vee B) \simeq \Diamond A \vee \Diamond B$.

Removing $\Diamond$ on the right is safe, so the applications of $(\Diamond r)$ and $(\vee l)$ can be permuted.

$$\dfrac{\dfrac{\dfrac{}{A \Rightarrow A, \Diamond B}}{A \Rightarrow \Diamond A, \Diamond B}\,(\Diamond r) \quad \dfrac{\dfrac{}{B \Rightarrow \Diamond A, B}}{B \Rightarrow \Diamond A, \Diamond B}\,(\Diamond r)}{\dfrac{A \vee B \Rightarrow \Diamond A, \Diamond B}{\Diamond(A \vee B) \Rightarrow \Diamond A, \Diamond B}\,(\Diamond l)}\,(\vee l)$$

There is no flexibility in the converse case, since removing the $\Diamond$ on the right too early would prevent us from being able to safely remove $\Diamond$ on the left.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \Rightarrow A, B}}{A \Rightarrow A \vee B}\,(\vee r)}{A \Rightarrow \Diamond(A \vee B)}\,(\Diamond r)}{\Diamond A \Rightarrow \Diamond(A \vee B)}\,(\Diamond l) \quad \dfrac{\dfrac{\dfrac{\dfrac{}{B \Rightarrow A, B}}{B \Rightarrow A \vee B}\,(\vee r)}{B \Rightarrow \Diamond(A \vee B)}\,(\Diamond r)}{\Diamond B \Rightarrow \Diamond(A \vee B)}\,(\Diamond l)}{\Diamond A \vee \Diamond B \Rightarrow \Diamond(A \vee B)}\,(\vee l)$$

b) Similarly, prove the equivalence $\Box(A \wedge B) \simeq \Box A \wedge \Box B$.

First we prove $\Box(A \wedge B) \Rightarrow \Box A \wedge \Box B$, which is a dual of $\Diamond A \vee \Diamond B \Rightarrow \Diamond(A \vee B)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A, B \Rightarrow A}}{A \wedge B \Rightarrow A}\,(\wedge l)}{\Box(A \wedge B) \Rightarrow A}\,(\Box l)}{\Box(A \wedge B) \Rightarrow \Box A}\,(\Box r) \quad \dfrac{\dfrac{\dfrac{\dfrac{}{A, B \Rightarrow B}}{A \wedge B \Rightarrow B}\,(\wedge l)}{\Box(A \wedge B) \Rightarrow B}\,(\Box l)}{\Box(A \wedge B) \Rightarrow \Box B}\,(\Box r)}{\Box(A \wedge B) \Rightarrow \Box A \wedge \Box B}\,(\wedge r)$$

Next, we prove $\Box A, \Box B \Rightarrow \Box(A \wedge B)$, which is a dual of $\Diamond(A \vee B) \Rightarrow \Diamond A, \Diamond B$.

$$\dfrac{\dfrac{\dfrac{}{A, \Box B \Rightarrow A}\,(\Box l)}{\Box A, \Box B \Rightarrow A} \quad \dfrac{\dfrac{}{\Box A, B \Rightarrow B}\,(\Box l)}{\Box A, \Box B \Rightarrow B}}{\dfrac{\Box A, \Box B \Rightarrow A \wedge B}{\Box A, \Box B \Rightarrow \Box(A \wedge B)}\,(\Box r)}\,(\wedge r)$$

4. Prove the following sequents.

$$\Diamond(A \to B), \Box A \Rightarrow \Diamond B \qquad \Box\Diamond\Box A, \Box\Diamond\Box B \Rightarrow \Box\Diamond\Box(A \wedge B)$$

The first step performed must be $(\Diamond l)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \Rightarrow \Diamond B, A}\,(\Box l)}{\Box A \Rightarrow \Diamond B, A} \quad \dfrac{\dfrac{}{B, \Box A \Rightarrow B}\,(\Diamond r)}{B, \Box A \Rightarrow \Diamond B}}{A \to B, \Box A \Rightarrow \Diamond B}\,(\to l)}{\Diamond(A \to B), \Box A \Rightarrow \Diamond B}\,(\Diamond l)$$

The second problem is tricky and requires careful attention to the order in which the modal operators are tackled. To avoid losing information, the critical rules are only applied when every formula on the left begins with a box, and every formula on the right begins with a diamond (in both cases excluding the formula directly affected by the rule. It is especially important to avoid batch-applying rules involving modalities, since applying a rule with a side-condition may remove propositions that we expect to be able to operate on later.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A, \Box B \Rightarrow A}}{\Box A, \Box B \Rightarrow A}(\Box l) \quad \dfrac{}{\Box A, B \Rightarrow B}}{\Box A, \Box B \Rightarrow A \wedge B}(\Box l)(\wedge r)}{\Box A, \Box B \Rightarrow \Box(A \wedge B)}(\Box r)}{\Box A, \Box B \Rightarrow \Diamond\Box(A \wedge B)}(\Diamond r)}{\Box A, \Diamond\Box B \Rightarrow \Diamond\Box(A \wedge B)}(\Diamond l)}{\Box A, \Box\Diamond\Box B \Rightarrow \Diamond\Box(A \wedge B)}(\Box l)}{\Diamond\Box A, \Box\Diamond\Box B \Rightarrow \Diamond\Box(A \wedge B)}(\Diamond l)}{\Box\Diamond\Box A, \Box\Diamond\Box B \Rightarrow \Box\Diamond\Box(A \wedge B)}(\Box l)}{\Box\Diamond\Box A, \Box\Diamond\Box B \Rightarrow \Box\Diamond\Box(A \wedge B)}(\Box r)$$

13. Tableaux-based methods

1. Use the free-variable tableau calculus to prove the following formulas.

$$(\exists y.\ \forall x.\ R(x, y)) \rightarrow (\forall x.\ \exists y.\ R(x, y))$$
$$(P(a, b) \vee \exists z.\ P(z, z)) \rightarrow \exists x, y.\ P(x, y)$$
$$((\exists x.\ P(x)) \rightarrow Q) \rightarrow (\forall x.\ P(x) \rightarrow Q)$$

$$(\exists y.\ \forall x.\ R(x, y)) \rightarrow (\forall x.\ \exists y.\ R(x, y))$$

We negate and convert to NNF:

$$\neg((\exists y.\ \forall x.\ R(x, y)) \rightarrow (\forall x.\ \exists y.\ R(x, y))) \simeq (\exists y.\ \forall x.\ R(x, y)) \wedge (\exists x.\ \forall y.\ \neg R(x, y))$$

We are using the free-variable tableau calculus, so we must Skolemise by replacing the two existentials with Skolem constants:

$$(\forall x.\ R(x, a)) \wedge (\forall y.\ \neg R(b, y))$$

Finally, we put this on the LHS of a sequent and derive a contradiction:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{w \mapsto b, v \mapsto a}}{R(w,a),\ \neg R(b,v) \Rightarrow}}{R(w,a),\ \forall y.\ \neg R(b,y) \Rightarrow}\ (\forall l, u/y)}{\forall x.\ R(x,a),\ \forall y.\ \neg R(b,y) \Rightarrow}\ (\forall l, w/x)}{(\forall x.\ R(x,a)) \wedge (\forall y.\ \neg R(b,y)) \Rightarrow}\ (\wedge l)$$

For

$$(P(a,b) \vee \exists z.\ P(z,z)) \rightarrow \exists x, y.\ P(x,y)$$

we negate, convert to NNF and Skolemise:

$$\begin{aligned} & \neg((P(a,b) \vee \exists z.\ P(z,z)) \rightarrow \exists x, y.\ P(x,y)) \\ \simeq\ & (P(a,b) \vee \exists z.\ P(z,z)) \wedge (\forall x.\ \forall y.\ \neg P(x,y)) && \text{(negate)} \\ \Longrightarrow\ & (P(a,b) \vee P(c,c)) \wedge (\forall x.\ \forall y.\ \neg P(x,y)) && \text{(Skolemise)} \end{aligned}$$

where c is a new Skolem constant.

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{u_1 \mapsto a, w_1 \mapsto b}}{P(a,b),\ \neg P(u_1,w_1) \Rightarrow}}{P(a,b),\ \forall y.\ \neg P(u_1,y) \Rightarrow}\ (\forall l, w_1/y)}{P(a,b),\ \forall x.\ \forall y.\ \neg P(x,y) \Rightarrow}\ (\forall l, u_1/x) \qquad \dfrac{\dfrac{\dfrac{\overline{u_2 \mapsto c, w_2 \mapsto c}}{P(c,c),\ \neg P(u_2,w_2) \Rightarrow}}{P(c,c),\ \forall y.\ \neg P(u_2,y) \Rightarrow}\ (\forall l, w_2/y)}{P(c,c),\ \forall x.\ \forall y.\ \neg P(x,y) \Rightarrow}\ (\forall l, u_2/x)}{P(a,b) \vee P(c,c),\ \forall x.\ \forall y.\ \neg P(x,y) \Rightarrow}\ (\vee l)$$
$$\dfrac{}{(P(a,b) \vee P(c,c)) \wedge (\forall x.\ \forall y.\ \neg P(x,y)) \Rightarrow}\ (\wedge l)$$

For

$$((\exists x.\ P(x)) \rightarrow Q) \rightarrow (\forall x.\ P(x) \rightarrow Q)$$

we negate, convert to NNF and Skolemise:

$$\begin{aligned} & \neg(((\exists x.\ P(x)) \rightarrow Q) \rightarrow (\forall x.\ P(x) \rightarrow Q)) \\ \simeq\ & ((\forall x.\ \neg P(x)) \vee Q) \wedge (\exists x.\ P(x) \wedge \neg Q) && \text{(negate)} \\ \Longrightarrow\ & ((\forall x.\ \neg P(x)) \vee Q) \wedge P(a) \wedge \neg Q && \text{(Skolemise)} \end{aligned}$$

Separating the conjunctions and placing on the LHS of a sequent, we get

$$\dfrac{\dfrac{\dfrac{\overline{u \mapsto a}}{\neg P(u),\ P(a),\ \neg Q \Rightarrow}}{\forall x.\ \neg P(x),\ P(a),\ \neg Q \Rightarrow}\ (\forall l, u/x) \qquad \dfrac{}{Q,\ P(a),\ \neg Q \Rightarrow}}{(\forall x.\ \neg P(x)) \vee Q,\ P(a),\ \neg Q \Rightarrow}\ (\vee l)$$

2. Compare the sequent calculus, resolution and the free-variable tableau calculus by using each

of them to prove the following formula.

$$(P(a,b) \lor \exists z.\, P(z,z)) \to \exists x, y.\, P(x,y)$$

Sequent calculus. A proof system for direct proof: the formula is placed on the RHS of the sequent without any alterations, and one follows the syntax-directed connective- and quantifier-introduction rules to reach the basic sequent $\Gamma, A \Rightarrow A, \Delta$, representing the conclusion of A from an assumption of A. While sequent calculus requires no preprocessing of the formula (negation or conversion to a normal form), the proof algorithm is more complicated due to it needing to handle all possible logical connectives both in the assumption and the goal context. The $(\exists l)$ and $(\forall r)$ quantifier rules impose side conditions on the freshness of variable names, while $(\forall l)$ and $(\exists r)$ require one to find appropriate witness terms and instantiations in the middle of a proof. Since we may not be able to know what particular instance or witness will be needed, these steps may require educated guessing and backtracking.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{P(a,b) \Rightarrow P(a,b)}}{P(a,b) \Rightarrow \exists y.\, P(a,y)}\,(\exists r, b/y)}{P(a,b) \Rightarrow \exists x, y.\, P(x,y)}\,(\exists r, a/x) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{P(z,z) \Rightarrow P(z,z)}}{P(z,z) \Rightarrow \exists y.\, P(z,y)}\,(\exists r, z/y)}{P(z,z) \Rightarrow \exists x, y.\, P(x,y)}\,(\exists r, z/x)}{\exists z.\, P(z,z) \Rightarrow \exists x, y.\, P(x,y)}\,(\exists l)}{(P(a,b) \lor \exists z.\, P(z,z)) \Rightarrow \exists x, y.\, P(x,y)}\,(\lor l)$$
$$\Rightarrow (P(a,b) \lor \exists z.\, P(z,z)) \to \exists x, y.\, P(x,y)\,(\to r)$$

Resolution. A refutation proof technique for establishing validity: we show that the negation of a formula is unsatisfiable. Resolution does away with the complicated and rigid sequent calculus framework in favour of a single inference rule applied to a normalised representation of the problem. The formula is first negated, then converted to conjunctive normal form, also known as clausal form. Existentially quantified variables are eliminated using Skolemisation, and universal quantifiers are left implicit. The resolution rule makes hypothetical inferences amongst the clauses, aiming to reach the empty clause that denotes a contradiction. Instantiation of variables is done via unification, ensuring that instance search is never done "blindly". The main drawback is that CNF conversion is error-prone when done by hand, and may well result in exponential blow-up in the number of clauses – often most of the work of a resolution proof happens in this normalisation step.

Conversion of the formula to clausal form has already been demonstrated above:

$$(P(a,b) \lor P(c,c)) \land (\forall x.\, \forall y.\, \neg P(x,y))$$

In clausal form, this is:

$$① \{P(a,b), P(c,c)\} \qquad ② \{\neg P(x,y)\}$$

We resolve the two clauses on $P(a,b)$, unifying x with a and y with b to get $\{P(c,c)\}$, then resolve this new clause with ② again (unifying both x and y with c) to reach the empty clause.

Free-variable tableau calculus. We combine two favourable aspects of the sequent and resolution techniques: limited preprocessing and natural reasoning steps on the one hand, and a small set of inference rules and streamlined handling of variables on the other. Free-variable tableau calculus is also a contradiction-based proof technique, so the formula is first negated and converted to negation normal form. This eliminates implication and pushes negation to atomic formulae, but does not distribute conjunctions over disjunctions. The reduced number of connectives means that we do not require a large number of sequent rules: the left introductions for disjunction, conjunction, and universal quantification suffice. The basic sequent becomes $\Gamma, A, \neg A \Rightarrow$, expressing a contradiction. Free-variable tableaux avoid unguided instance search for the $(\forall l)$ rule by replacing universally quantified variables with fresh ones, and instantiating them at the very end with unification. To do this, the NNF formula must also be Skolemised, otherwise we could not differentiate between the $(\exists l)$ and $(\forall l)$ rules.

The tableaux proof of the formula has already been demonstrated above; the important differences with the normal sequent proof is that instantiation of the variables is deferred until the very end.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{u_1 \mapsto a, w_1 \mapsto b}}{P(a,b),\ \neg P(u_1,w_1) \Rightarrow}}{P(a,b),\ \forall y.\ \neg P(u_1,y) \Rightarrow}\ (\forall l, w_1/y)}{P(a,b),\ \forall x.\ \forall y.\ \neg P(x,y) \Rightarrow}\ (\forall l, u_1/x) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{u_2 \mapsto c, w_2 \mapsto c}}{P(c,c),\ \neg P(u_2,w_2) \Rightarrow}}{P(c,c),\ \forall y.\ \neg P(u_2,y) \Rightarrow}\ (\forall l, w_2/y)}{P(c,c),\ \forall x.\ \forall y.\ \neg P(x,y) \Rightarrow}\ (\forall l, u_2/x)}{P(a,b) \vee P(c,c),\ \forall x.\ \forall y.\ \neg P(x,y) \Rightarrow}\ (\vee l)$$

$$\dfrac{P(a,b) \vee P(c,c),\ \forall x.\ \forall y.\ \neg P(x,y) \Rightarrow}{(P(a,b) \vee P(c,c)) \wedge (\forall x.\ \forall y.\ \neg P(x,y)) \Rightarrow}\ (\wedge l)$$

Optional exercise

Temporal logic is not the only type of modal logic: depending on how we interpret $\Box A$, we can admit different axioms and relational properties for our logic. Some are of philosophical interest, while others have found use in computer science and mathematics. Below are a few examples:

Name	**Domain**	**Interpretation of $\Box A$**	**Interpretation of $\Diamond A$**
Temporal	time	A always holds	
Alethic	necessity		A possibly holds
Doxastic	belief	I believe that A holds	
Epistemic	knowledge		For all I know, A holds
Deontic	duty	It is obligatory that A holds	

a) Complete the table either by intuition or through research. Recall that $\Diamond A$ is defined as $\neg\Box\neg A$.

In some cases the duality is self-evident and can be expressed using appropriate pairs of English words. In other cases (namely belief and knowledge) a bit more thinking is required; we are essentially trying to capture being "indifferent" to the proposition, i.e. that we are not going to argue against it if it is proposed to be true. This is suitably between affirming and denying the proposition, which is what $\Diamond$ intends to express.

Name	**Domain**	**Interpretation of $\Box A$**	**Interpretation of $\Diamond A$**
Temporal	time	A always holds	A eventually holds
Alethic	necessity	A necessarily holds	A possibly holds
Doxastic	belief	I believe that A holds	A is consistent with my beliefs
Epistemic	knowledge	I know that A holds	For all I know, A holds
Deontic	duty	It is obligatory that A holds	It is permitted that A holds

b) Assign each of the formulae below to the modal logics in which they could be reasonably assumed as axioms. For example, does belief of A imply the truth of A?

a) $\Box(A \to B) \land \Box A \to \Box B$

b) $\Box A \to A$

c) $\Box A \to \Box\Box A$

d) $\Diamond A \to \Box\Diamond A$

e) $\Diamond\top$

f) $\Box A \to \Diamond A$

Distribution: $\Box(A \to B) \land \Box A \to \Box B$. This is nothing but the ("uncurried" form of) axiom K, assumed to hold in every modal logic.

Reflexivity: $\Box A \to A$.

- *Time*: we assume that the future includes the present, so if $\Box A$ holds at a current "time step" (world) we must also have A.
- *Necessity*: necessary truth is stronger than simple truth.
- ~~*Belief*~~: most definitely not reflexive: believing something doesn't make it true.
- *Knowledge*: the difference between knowledge and belief is that (ideally) knowing something to be true means it must be true; presumably the only way to know a proposition to be true is to have proof or irrefutable evidence for it.
- ~~*Duty*~~: in an ideal world, maybe – but there will always be rule-breakers, so obligatory things are not necessarily true.

Transitivity: $\Box A \to \Box\Box A$.

- *Time*: if something holds always in the future, this will also be case at any point in the future.
- *Necessity*: necessary things are necessarily necessary – logical laws are assertions whose truth cannot be denied. However, if we're talking about physical, rather than logical necessity, transitivity would mean that the physical laws themselves entail that they should be laws of the universe, which is more questionable.
- *Belief and knowledge*: transitivity represents positive introspection: if I know something

> to be true, I know that I know it to be true[a]. Negative introspection $\neg\Box A \rightarrow \Box\neg\Box A$ would then be "if I don't know A, I know that I don't know A" or "I am aware of the limits of my knowledge".

[a] Epistemic analysis of *Friends* S05E14 anyone?

c) *Provability logic* is an interesting variant of a modal logic which interprets $\Box A$ as "A *is provable in the theory* T" where T is some axiomatic system that we are working in (such as Peano arithmetic). What (if anything) can we say about a particular system T if we know that:

(i) the formula $\Box A \rightarrow A$ is an axiom?

(ii) the formula $\neg\Box\bot$ is an axiom?

(iii) the formula $\Box A \rightarrow \Diamond A$ is *not* an axiom?